# An Area-decomposition Based Approach for Cooperative Tasking and Coordination of UAVs in a Search and Coverage Mission

**Vinh K, Solomon Gebreyohannes, Ali Karimoddini**

*Abstract*— **This paper develops a low-computation area-decomposition based approach for tasking and planning for a team of Unmanned Aerial Vehicles (UAVs) involved in a search mission. For an area of interest captured as a convex polygon, a minimum bounding rectangle is calculated and decomposed into several partitions, which are assigned to different UAVs based on their flying and sensing capabilities. A novel Mixed Integer Linear Programming (MILP) based technique is developed to automatically task the UAVs to sweep the area of interest. If each individual UAV follows the created sweeping pattern and visit the generated waypoints, applying the developed method, it can be guaranteed that the entire area of interest is searched. The developed algorithm is applied to tasking and coordination of multiple UAVs searching an area on a farm and the implementation details and results are provided.**

## TABLE OF CONTENTS



## 1. INTRODUCTION

Unmanned aerial vehicles (UAVs) have been deployed in many different sensor-based applications such as bridge inspection [1], aerial mapping [2], reconnaissance [3], search and rescue [4], and precision agricultural farming [5]. Sensor-based application of UAVs and their applications to cover and survey an area have created great interest in civilians and military applications. Given the UAVs capabilities and their aerial vantage point, they can be deployed for surveying an area of interest in difficult terrains, where unmanned ground vehicles (UGVs) struggles.

There are different techniques for the control of a single UAV such as hierarchical control [6–8], hybrid control [9], neural networks [10], the differential geometry method [11], symbolic control [12,13], feedback control [14], nonlinear control [15], robust control [16], and model predictive control [17]. However, when we have a team of cooperative UAVs instead of a single sophisticated UAV system, we can distributively achieve the tasks with more flexibility. A major challenge is how to task the UAVs to collectively achieve the desired objectives. Most existing results either use a pre-assigned areas to task UAVs [18], rely either on heuristic approaches [19], or use auction-based techniques [20]. Here, a major issue is to go beyond heuristic approaches and develop an effective formal method that can optimally task a team of heterogeneous UAVs in a decentralized way.

To address these challenges, this paper develops a computationally effective tasking framework for coordination of a heterogeneous team of UAVs with different capabilities to search and survey an area of interest cooperatively. For this purpose, first a graphic user interface is developed by incorporating the Google Map to get the area of interest from a user. We assume that the area of interest is a convex polygon and the user determines the corners of the search area through the developed interface. If not, then we find the bounding convex polygon for the provided concave area. Once the area of interest is identified, to reduce the computation cost of the task assignment process for the UAVs, we find the smallest rectangle that surrounds the area of interest by applying a modified version of the calipers algorithm. Then, to task each UAV to search a part of the area of interest, a novel decomposition technique is developed to optimally decompose the area into smaller partitions, which will be assigned to different UAVs based on their flying capabilities (e.g., speed) and their sensing capabilities (e.g., sensors' footprint sizes).

Further, this paper develops a novel MILP-based technique to automatically generate a set of waypoints and an optimal sweeping pattern based on the footprint size of the UAV sensors, so that if the UAVs follow the designed sweeping pattern and the generated waypoints, it can be guaranteed that the entire area is surveyed. The optimal sweeping pattern for each partition is determined based on the geometry of the decomposed areas, the size of the sensors' footprint, and the minimum required overlap of the image frames. For this purpose, we will determine the sweeping direction that minimizes the number of turns of the UAVs and the number of coverage rows, while maintaining the minimum required overlap between coverage rows that is required for collecting sufficient information. This problem is formulated as an MILP [21,22] and the solution is found via the CPLEX solver [23, 24]. This results in a series of waypoints that should be visited in a particular order to cover each partition. The developed algorithm is implemented on multiple UAVs to survey an area of interest in NC A&T University's farm and detailed steps of this search scenario are provided.

The rest of the paper is organized as follows. In Section 2, the procedure for uploading the map and identifying the region of interest through the developed user interface is explained. Section 3 develops a computationally effective method for tasking of the UAVs and generating the waypoints that should

The authors are with the Department of Electrical and Computer Engineering, North Carolina Agricultural and Technical State University, Greensboro, NC 27411 USA.

Corresponding author: A. Karimoddini, Tel: +13362853313, Eamil: akarimod@ncat.edu.

978-1-5386-6854-2/19/$31.00 

be visited by each UAV member of the team. Section 4 develops an MILP-based planning mechanism to design an optimal sweeping pattern to visit all generated waypoints. In Section 6, the developed method is implemented to task a set of UAVs to survey a part of the NC A&T University's farm. The paper is concluded in Section 6.

## 2. IDENTIFYING AN AREA OF INTEREST

The developed toolbox first receives the map from the user through a graphical user-interface and analyzes it to understand the coordinates of the points (pixels) in the map. Then, the user can specify the area of interest as a polygon in the form of a set of points (vertices).

### *2.1 Analyzing the Uploaded Map*

The user is able to upload the map through the developed graphical user-interface. The map provides coordinates in degrees using the standard World Geodetic System 1984 (WGS84) reference coordinate system [25]. To find the coordinates of the points on the map in meter, we needed to know the width, $I_y$, and length, $I_x$, of the map in pixel. In addition, the longitude of the East and West side of the map, $E(x)$ and $W(x)$, as well as the latitude of the North and South of the map, $N(y)$ and $S(y)$, are needed. Having this information, visualized in Figure 1, we can calculate the coordinates of all points on the map in meter as it will be explained next. To find the width and height of the map in

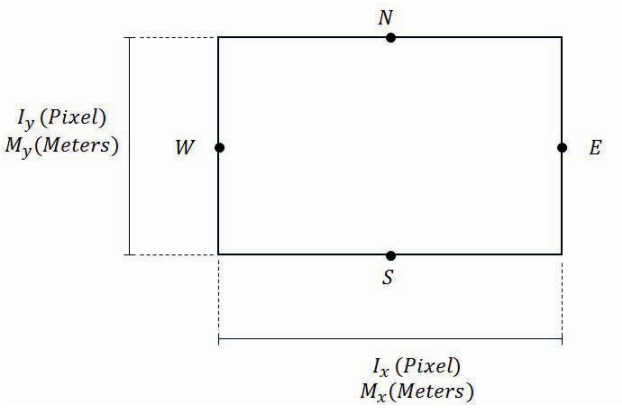



**Figure 1**: Dimensions of the uploaded map.

meter, $M_x$ and $M_y$, we use the following equation:

$$M_x = (6371 * \frac{|E(x) - W(x)|}{180} * \pi) * 1000$$
$$M_y = (6371 * \frac{|N(y) - S(y)|}{180} * \pi) * 1000 \quad (1)$$

where $6371\ km$ is the size of the radius of the earth. Now, for any point $w$ on the map with the coordinates of $(w^p(x), w^p(y))$ in image frame in pixel, we will find the corresponding coordinates $(w(x), w(y))$ in meter as follows:

$$w(x) = \frac{w^p(x)}{I_x} * M_x$$
$$w(y) = \frac{w^p(y)}{I_y} * M_y \quad (2)$$

where the origin is assumed to be at Northwest corner, the x-axis toward the east and the y-axis toward the south of the map.

### *2.2 Specifying the Area of Interest*

Now, using the uploaded map, the user can introduce the area of interest as a polygon in the form of a series of points on the map. Each vertex of the polygon is captured in the image frame in the form of $(V_i^p(x), V_i^p(y))$ in pixel. The vertices then are converted to meter using Eq (2).

## 3. OPTIMAL TASKING FOR COOPERATIVE SEARCH

This section presents the optimal tasking for UAVs cooperatively involved in a search mission. This includes finding the minimum area bounding rectangular box for the Region Of Interest (ROI), assigning a specific area to each UAV, and generating waypoints for each UAV to survey the area.

### *3.1 Finding the Minimal Area Bounding Box for an Area of Interest*

Once the area of interest is identified as a polygon, to ease the task assignment for the UAVs, we find the smallest rectangle that surrounds the area of interest. There are several different types of polygons including convex, concave, and complex as shown in Figure 2. In this paper, we will only consider regular convex and concave polygons. To bound a rectangular bounding box, we use an efficient algorithm called *rotating calipers*. In case of a concave polygon, we first enclose it by a convex polygon before applying the rotating calipers algorithm.

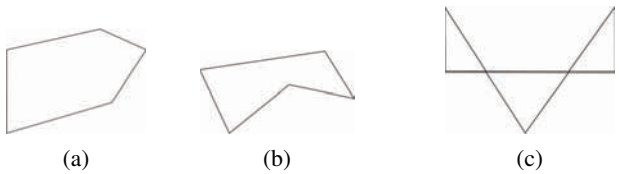
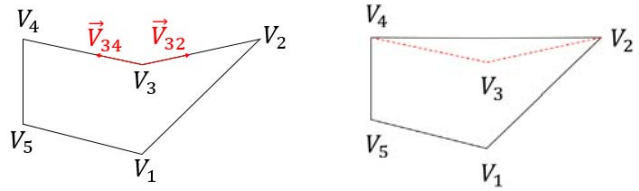



**Figure 2**: Different types of polygons: (a) convex polygon, (b) concave polygon, and (c) complex polygon.

*3.1.1 Enclosing a Concave Polygon by a Convex Polygon*—In the case that the ROI is a concave polygon, we first need to enclose it by a convex polygon. There are many algorithms that can be used for computing the convex hull of a concave polygon such as Giftwrapping [26], Graham scan [27], and Chan [28]. In this paper, however, we use a much simpler algorithm to obtain the convex hull of the polygon using the cross product operator to determine the vertices, which cause a polygon to become concave.

To illustrate the idea, consider a polygon shown in Figure 3(a), where the vertex $V_3$ makes it concave. Consider that the vertices are given in a counter-clockwise (CCW) direction without having 3 points collinear. For each edge from $V_i$ to $V_j$, consider the edge vector $\vec{V}_{i,j}$. We will remove the concave vertex $V_3$ by observing a negative cross product of the two edge vectors along with the z-axis (resulted from an obtuse interior angle between $\vec{V}_{3,2}$ and $\vec{V}_{3,4}$), and connect the remaining vertices. Hence, a convex quadrilateral with the vertices $V_1$, $V_2$, $V_4$, and $V_5$ is resulted as shown in Figure 3(b). This process is detailed in Algorithm 1.




(a) (b)

**Figure 3**: Concave to convex polygon transformation: (a) The given concave polygon, (b) After applying Algorithm 1, the concave vertex is removed, resulting in a convex polygon.

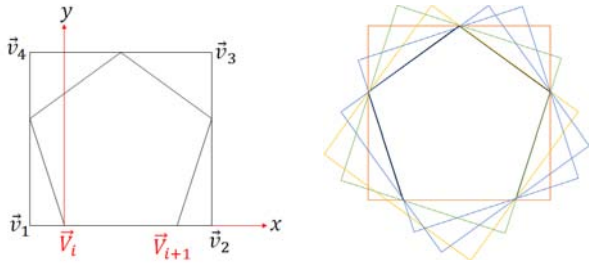
**Algorithm 1** Enclosing a Concave Polygon by a Convex Polygon

**Input:** Concave polygon, $Q$
**Input:** Vertices $V_i$, $i = 1, \cdots, n$, ordered CCW
$V_0 := V_n$
$V_{n+1} := V_1$
**for** $i = 1 : n$ **do**
  Find the vector edge $\vec{V}_{i-1,i}$ connecting $V_{i-1}$ to $V_i$
  Find the vector edge $\vec{V}_{i,i+1}$ connecting $V_i$ to $V_{i+1}$
  Find $\vec{V}_t = \vec{V}_{i-1,i} \times \vec{V}_{i,i+1}$
  **if** $\vec{V}_t$ is along with the negative direction of $Z$-axis **then**
    Remove $V_i$
Connect the remaining vertices to form a new polygon $P$
**Output:** Convex polygon, $P$

**Algorithm 2** Rotating Calipers Algorithm

**Input:** Convex polygon, $P$
**Input:** Vertices $V_i$, $i = 1, \cdots, n$, ordered CCW
$V_{n+1} := V_1$
$A_{min} = M$, where $M$ is a large positive number
**for** $i = 1 : n$ **do**
  Define $V_i$ as the origin of a new Cartesian coordinate system, whose x-axis pointing from $V_i$ toward $V_{i+1}$
  Find $\bar{V}_j$ as the transferred coordinates of $V_j$ to the new coordinate system, $j = 1, \cdots, n$
  $\bar{v}_1(y) = \bar{v}_2(y) := 0$
  $\bar{v}_1(x) = \bar{v}_4(x) := \min_{j=1}^{n} \bar{V}_j(x)$
  $\bar{v}_2(x) = \bar{v}_3(x) := \max_{j=1}^{n} \bar{V}_j(x)$
  $\bar{v}_3(y) = \bar{v}_4(y) := \max_{j=1}^{n} \bar{V}_j(y)$
  calculate $A = (\bar{v}_2(x) - \bar{v}_1(x))(\bar{v}_4(y) - \bar{v}_1(y))$
  **if** $A < A_{min}$ **then**
    $A_{min} = A$
    $\bar{v}_{1_{min}} = \bar{v}_1$, $\bar{v}_{2_{min}} = \bar{v}_2$
    $\bar{v}_{3_{min}} = \bar{v}_3$, $\bar{v}_{4_{min}} = \bar{v}_4$
Find $v_{1min}$, $v_{2min}$, $v_{3min}$, and $v_{4min}$ by transferring back the coordinates of $\bar{v}_{1_{min}}$, $\bar{v}_{1_{min}}$, $\bar{v}_{1_{min}}$, and $\bar{v}_{1_{min}}$ to the original coordinate system
**Output:** $A_{min}$, $v_{1min}$, $v_{2min}$, $v_{3min}$, and $v_{4min}$

*3.1.2 Minimal Area Bounding Box for a Convex Polygon*—The *rotating Calipers* algorithm [29] provides an efficient multi-purpose computational tool, which is generalized in a number of ways to solve several geometric problems, for approximating the bounding box area for a convex polygon. The algorithm provides a bounding rectangle with the minimum area, i.e., a rectangle (among many possible) that best represents the area of interest.




(a) Initial bounding box of a convex polygon.

(b) Rotating the base supporting line to obtain all candidate rectangles

**Figure 4**: Finding the minimal area bounding rectangle.

The *Calipers algorithm* starts with having 4 orthogonal supporting lines as shown in Figure 4a, so that the resulting rectangle has one of its edges coincide with the edge of the polygon [30, 31]. The supporting lines are then rotated until it overlaps over all the edges of the polygon to find the minimum bounding rectangle. This process constructs the smallest rectangle in linear time, i.e., $O(n)$ for $n-$sided polygon. This procedure is detailed in Algorithm 2.

*3.2 Cooperative Tasking for a Search and Coverage Mission*

The UAV that has a higher speed and has sensors with larger sensor footprint can cover an area faster. To ensure the total area coverage, the UAVs will survey the area in a distributed way. For this purpose, we first decompose the survey area into smaller partitions based on the flying capabilities of UAVs (e.g., speed) and their sensors' capabilities (e.g., footprint size). Then, each partition is assigned to one of the UAVs based on its capabilities, while ensuring that the whole area of interest is optimally and collectively searched by the team of UAVs.

*3.2.2 Area-decomposition Based Tasking*—Assume an area of $A$ to be covered by $n$ number of fixed-wing UAVs, say $UAV_1$, $UAV_2, \cdots, UAV_n$. For $UAV_i$ we define the capability $c_i$ as:

$$c_i = V_{mi} L_{xi} \tag{3}$$

where $V_{mi}$ is the maximum velocity of $UAV_i$ and $L_{xi}$ is the length of the sensor's footprint (lateral footprint's size). We then can calculate the relative UAVs' capability $cr_i$ as

$$cr_i = \frac{c_i}{\sum_{j=1}^{n} c_j} \tag{4}$$

for which it can be checked that $0 \leq cr_i \leq 1$, and $\sum_{i=1}^{n} cr_i = 1$.

Using the relative capabilities of UAVs defined in Eq. 4, we will then decompose the area $A$ into mutually exclusive smaller areas, $A_i$, as:

$$A_i = cr_i * A \tag{5}$$

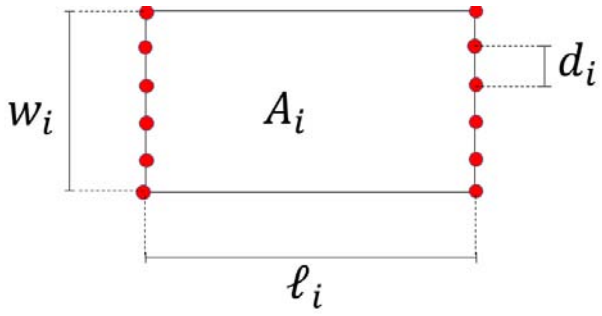



**Figure 5**: The generated waypoints for the $UAV_i$ covering the assigned area $A_i$ when the distance between the rows is $d_i$.

so that $\cup_{i=1}^{n} A_i = A$ and $A_i \cap A_j = \emptyset$, for all $i \neq j$.

*3.3 Waypoint Generation*

To accomplish the assigned tasks (searching the assigned areas), we generate a set of waypoints so that by visiting the generated waypoints with a proper order (sweeping pattern), we can ensure that the assigned area is searched. The waypoints for each partition are generated based on the geometry of the decomposed areas, the size of the sensor footprints, and the minimum required side overlap of the image frames.

The generated waypoints should require a minimum number of coverage rows and a minimum number of turns of the UAVs, while maintaining the minimum required overlap between coverage rows that is required for collecting sufficient information (e.g., for creating image mosaics). To have the minimum number of turns, we choose to sweep the area along with its height of the rectangle that describes the search area assigned to the UAV. To reduce the number of rows of the sweeping pattern, we respect the minimum required side overlap by choosing the number of rows as follows:

$$N_i = \frac{w_i}{L_{xi}(1 - O_s)} \tag{6}$$

where $w_i$ is the width of the $i_{th}$ rectangular partitioned assigned to $UAV_i$, and $O_s$ is the minimum required side overlap. Therefore, the distance between the rows are determined as:

$$d_i = \frac{w_i}{N_i} \tag{7}$$

based on which we can then generate the waypoints as shown in Figure 5.

## 4. Designing an Optimal Sweeping Pattern

Having the set of generated waypoints for each region, we will then determine the sweeping direction to visit the waypoints in a particular order to cover each partition and search the assigned area. For this purpose, an optimization-based planning approach is developed by employing the Mixed Integer Linear Programming (MILP) to generate optimal and feasible path for efficiently coordinating UAVs to cooperatively search an environment. Mathematically, having the assigned set of waypoints $\mathbf{W}^k = [W_1^k, W_2^k, \cdots, W_{n_k}^k]$ for the $k$th UAV, $k = 1, \cdots, N$, to cover the area $A_k$ as shown in Fig. 6, the problem is how to optimally generate the path for the UAVs to visit these waypoints, while respecting the sweeping pattern. The index $n_k$ refers to the number of waypoints for the $k$th UAV. Note that we chose $W_1^k$ and $W_{n_k}^k$ referring to the start and return base stations and the rest of waypoints are found based on calculations in Section 3. To solve this problem we employ the MILP technique [22] by formulating all requirements as linear constraints and integer decision variables.

To capture sweeping pattern requirements, let's define binary variables $E_{ij}^k$, for which $E_{ij}^k = 1$ if $kth$ UAV first visits $W_i^k$ and then immediately visits $W_j^k$, i.e. there is a directed edge from $W_i^k$ to $W_j^k$.

To ensure that the whole area is covered, we need the UAVs to start from $W_1^k$ and then visit all waypoints. This can be archived by the following two conditions:

1. Except the first waypoint, $W_1^k$, all other waypoints should have one and only one entering edge:

$$\sum_{i=1}^{n_k} E_{i,j}^k = 1 \tag{8}$$
$$j = 2, 3, 4, \cdots, n_k;\ k = 1, \cdots, N$$

For the first waypoint, however, there should be no entering edge:

$$\sum_{i=1}^{n_k} E_{i,1}^k = 0 \tag{9}$$
$$k = 1, \cdots, N$$

2. Except the last waypoint, $W_{n_k}^k$, all other waypoints should have one and only one exit edge:

$$\sum_{j=1}^{n_k} E_{i,j}^k = 1 \tag{10}$$
$$i = 1, 2, \cdots, n_k - 1;\ k = 1, \cdots, N$$

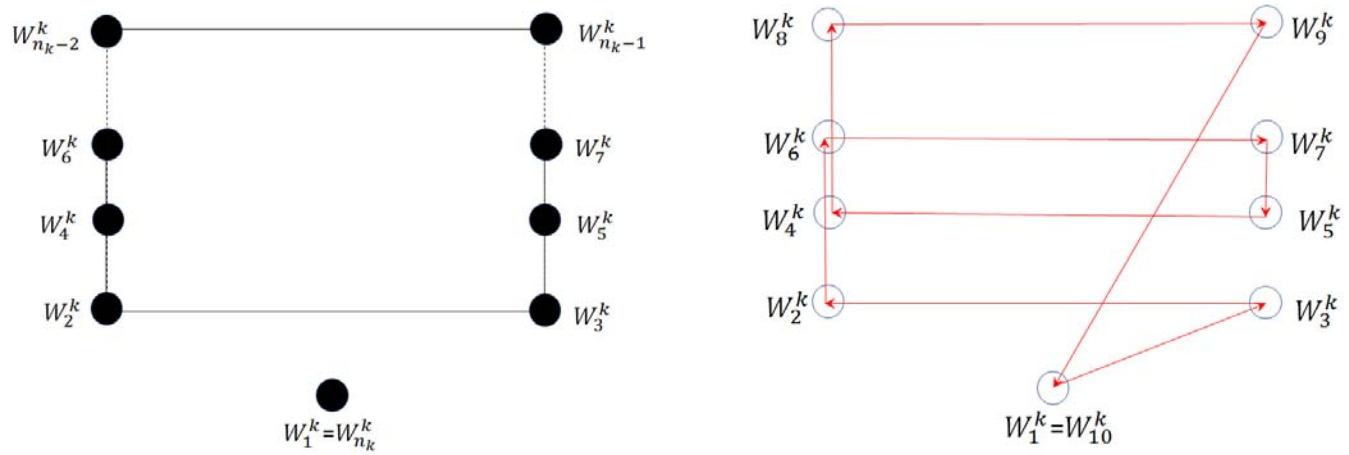
For the last waypoint, however, there should be no exit edge:

$$\sum_{j=1}^{n_k} E_{n_k,j}^k = 0 \tag{11}$$
$$k = 1, \cdots, N$$

3. The UAVs should leave the reached waypoint, i.e., there should be an exiting edge from the reached waypoints:

$$\sum_{i=1}^{n_k} \sum_{j=1}^{n_k} (E_{i,p}^k - E_{p,j}^k) = 0 \tag{12}$$
$$p = 2, 3, \cdots, n_k - 1;\ k = 1, \cdots, N$$

We also need to make sure that edges connect the waypoints




(a) (b)

**Figure 6**: (a) The set of waypoints $W^k = [W^k{}_1, W^k_2, \cdots, W^k_{n_k}]$ to be visited by *kth* UAV to form a sweeping pattern to cover the area. (b) An example of a sweeping pattern for 10 waypoints for *kth* UAV.

that are on the same row. For example, in Fig. 6, an edge from the waypoint $W^k_2$ to $W^k_3$ or from $W^k_3$ to $W^k_2$ should exist, and hence, either $E^k_{2,3} = 1$ or $E^k_{3,2} = 1$. Also, for any two waypoints one the same row, only one edge in one direction is allowed, i.e. only one of $E^k_{2,3} = 1$ or $E^k_{3,2} = 1$ can be true. This will avoid the back and forth movement on the same row. The following equation captures these requirements:

$$E^k_{i,i+1} + E^k_{i+1,i} = 1 \qquad (13)$$
$$i = 2, 4, 6, \cdots, n_k - 2;\ k = 1, \cdots, N$$

We also need to make sure that except for the first and last waypoints, only vertical or horizontal moves are allowed from each waypoint and no diagonal/cross move is allowed. For example, in Fig. 6, an edge from the waypoint $W^k_2$ to $W^k_3$ or $W^k_4$ is allowed but from $W^k_2$ to $W^k_5$ is not allowed. Similarly, an edge from the waypoint $W^k_3$ to $W^k_2$ or $W^k_5$ is allowed but not to $W^k_4$. Alternatively, these requirements can be expressed as: if there is an edge connecting two waypints on the same row (a horizontal move in Fig. 6), it should be followed by an edge which is not a row and not a self-loop (it should be followed by a vertical move in Fig. 6). These requirements can be captured over odd and even waypoints through the following two conditions:

$$E^k_{i,i+1} = \sum_{j=(\{n_k\}\cup\{3,5,\cdots\})\setminus(i+1)}^{n_k-1} E^k_{i+1,j} \qquad (14)$$
$$i = 2, 4, 6, \cdots, n_k - 2;\ k = 1, \cdots, N$$

$$E^k_{i,i-1} = \sum_{j=(\{n_k\}\cup\{2,4,\cdots\})\setminus(i-1)}^{n_k} E_{i-1,j} \qquad (15)$$
$$i = 3, 5, 7, \cdots, n_k - 1;\ k = 1, \cdots, N$$

Respecting all the aforementioned conditions, we expect that the UAVs start from the first waypoint and visit all the waypoints, while avoiding a subtour. For this purpose we implement the Miller Tucker Zemlin (MTZ) [32] subtour elimination technique by introducing new integer variables $u^k_i$, $k = 1, \cdots, N$, $i = 1, \cdots, n_k$, where $u^k_i$ captures the order of visiting $i$th waypoint by the $k$th UAV. Since the UAVs start from the first waypoint, we will have:

$$u^k_1 = 1 \qquad (16)$$
$$k = 1, \cdots, N$$

and for the rest of the waypoints we have:

$$2 \le u^k_i \le n_k \qquad (17)$$
$$i = 2, \cdots, n_k;\ k = 1, \cdots, N$$

Further, in general, if $E^k_{i,j} = 0$, the $i$th and $j$th waypoints may be visited in any order, respecting $u^k_i - u^k_j \le n_k - 1$. However, if $E^k_{i,j} = 1$, then the $j$th waypoint will be visited right after the $i$th waypoint, i.e., $u^k_i - u^k_j \le -1$. These two conditions can be captured as:

$$u^k_i - u^k_j + n_k E^k_{i,j} \le n_k - 1 \qquad (18)$$
$$i, j = 2, \cdots, n_k,\ i \ne j;\ k = 1, \cdots, N$$


**Figure 7**: The uploaded map for the NC A&T State University to be surveyed by three UAVs.


**Figure 8**: The Region of Interest determined by the user as a concave polygon.

and visit Our objective is that all UAVs visit all the waypoints based on the optimal swiping pattern and eventually come back to the base station with minimum traveled distance. This can be distributively achieved by letting $W_{n_k}^k = W_1^k$ and by minimizing the following cost functions:

$$min\ J_k = \sum_{i=1}^{n_k} E_{i,j}^k \mid W_i^k - W_j^k \mid \qquad (19)$$
$$k = 1, \cdots, N$$

subjected to the sweeping pattern requirements captured by (8)- (18).

This formulated MILP problem is translated to the mathematical language AMPL [33], which can then can be solved using commercially available software such as CPLEX [23, 24].


**Figure 9**: The Region of Interest is approximated by a convex polygon, shown in blue.

## 5. CASE STUDY

The developed method is applied to the tasking and planning problem for three UAVs surveying the North Carolina A&T State farm. The map is loaded via the developed toolbox as shown in Figure 7, for which we have $I_x = 1492$, $I_y = 788$, $E(x) = -79.732$, $W(x) = -79.748$, $N(y) = 36.064$, and $S(y) = 36.057$. Using this information and Equation 1, we can find $M_x = 1681.7$ m and $M_y = 717.78$ m, based on which we can find the coordinates of any other points on the map using Equation 2. Through the developed graphic user-interface, the ROI is determined by the user as a concave polygon in the form of a series of ordered vertices, shown in Figure 8. The ROI is then approximated by a convex polygon using Algorithm 1, shown in Figure 9. The converted convex polygon then is bounded by a minimum rectangle using Algorithm 2, and the result is shown in Figure 10.


**Figure 10**: The Region of Interest is bounded by a minimum rectangle, shown in green.

The desired imaging performance in this scenario is requiring the side overlap $O_s = 0.5$. The lateral size of the sensors' footprints for these three UAVs are $L_{x_1} = 21$ m, $L_{x_2} = 29$ m, and $L_{x_3} = 47$ m. Using Equations 3 and 4, the relative capabilities of these three UAVs are calculated as $cr_1 = 0.29827$, $cr_2 = 0.20473$, and $cr_3 = 0.49701$. The ROI is the decomposed to three disjoint rectangular areas $A_1$, $A_2$, and $A_3$, shown in Figure 11 and assigned to $UAV_1$, $UAV_2$, and $UAV_3$, respectively.


**Figure 11**: The Region of Interest is partitioned into three rectangular areas.

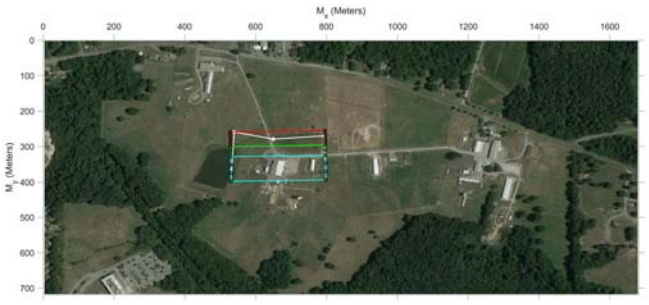



**Figure 12**: The generated waypoints shown as black circles.

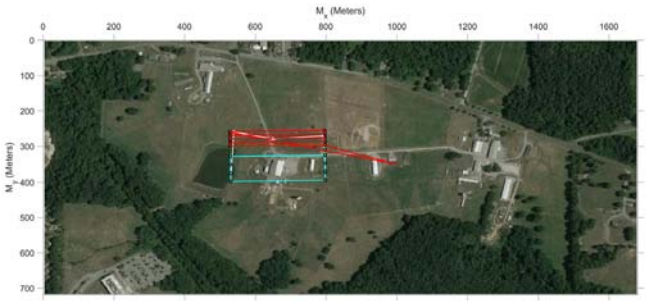



**Figure 13**: The sweeping pattern for the area $A_1$.

Knowing the UAV's sensor footprint and considering the minimum required side overlap of 0.5, the distance between the sweeping rows is calculated using Equation 6, based on which the waypoints for each UAV are found and shown in Figure 12. The optimal sweeping pattern then is then extracted using Equation 19 and is shown for the area $A_1$ in Figure 13. A video of the whole process starting from loading the map to generating the path for each UAV using the developed toolbox is available at https://youtu.be/0H1n6SvDZS0.

## 6. CONCLUSION

This paper developed a computationally effective framework for tasking and planning for a heterogeneous team of UAVs involved in a search mission. The developed took into account the UAV's flying and sensing capabilities, based on decomposed the search area and assigned to individual UAVs. Using the MILP technique the UAVs were automatically tasked to sweep the area of interest. The developed algorithm was applied to tasking and coordination of multiple UAVs searching an area on a farm and the implementation details and results were provided.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the support from NSF under the award number 1832110 and the Air Force Research Laboratory and OSD under agreement number FA8750-15-2-0116. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, Air Force Research Laboratory, OSD or the U.S. Government. Also, the authors would like to thank the technical support from Mr. Solomon Genene Gudeta, Mr. Tadewos Getahun Tadewos, and Mr. Md Khurram Monir Rabby.

## REFERENCES

[1] N. Metni and T. Hamel, "A uav for bridge inspection: Visual servoing control law with orientation limits," *Automation in construction*, vol. 17, no. 1, pp. 3–10, 2007.

[2] K. V. Stefanik, J. C. Gassaway, K. Kochersberger, and A. L. Abbott, "Uav-based stereo vision for rapid aerial terrain mapping," *GIScience & Remote Sensing*, vol. 48, no. 1, pp. 24–49, 2011.

[3] M. Bhaskaranand and J. D. Gibson, "Low-complexity video encoding for uav reconnaissance and surveillance," in *Military Communications Conference (2011-MILCOM)*, 2011, pp. 1633–1638.

[4] P. Doherty and P. Rudol, "A uav search and rescue scenario with human body detection and geolocalization," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2007, pp. 1–13.

[5] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review," *Precision agriculture*, vol. 13, no. 6, pp. 693–712, 2012.

[6] W. Findeisen, F. N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, and A. Wozniak, *Control and coordination in hierarchical systems*. John Wiley & Sons, 1980.

[7] J. S. Albus, A. J. Barbera, and R. N. Nagel, *Theory and practice of hierarchical control*. National Bureau of Standards, 1980.

[8] A. Karimoddini, G. Cai, B. Chen, H. Lin, and T. Lee, "Multi-layer flight control synthesis and analysis of a small-scale uav helicopter," in *IEEE Conference on Robotics Automation and Mechatronics (RAM)*, June 2010, pp. 321–326.

[9] A. Karimoddini, H. Lin, B. M. Chen, and T. H. Lee, "Hierarchical hybrid modelling and control of an unmanned helicopter," *International Journal of Control*, vol. 87, no. 9, pp. 1779–1793, 2014.

[10] R. Enns and J. Si, "Helicopter trimming and tracking control using direct neural dynamic programming," *IEEE Transactions on Neural Networks*, vol. 14, no. 4, pp. 929–939, 2003.

[11] A. Isidori, L. Marconi, and A. Serrani, "Robust nonlinear motion control of a helicopter," in *Robust Autonomous Guidance*. Springer, 2003, pp. 149–192.

[12] A. Karimoddini and H. Lin, "Hierarchical hybrid symbolic robot motion planning and control," *Asian Journal of Control*, vol. 17, no. 1, pp. 23–33, 2015.

[13] ——, "Hybrid symbolic control for robot motion planning," in *2013 10th IEEE International Conference on Control and Automation (ICCA)*, June 2013, pp. 1650–1655.

[14] K. Peng, G. Cai, B. M. Chen, M. Dong, K. Y. Lum, and T. H. Lee, "Design and implementation of an autonomous flight control law for a uav helicopter," *Automatica*, vol. 45, no. 10, pp. 2333–2338, 2009.

[15] H. Voos, "Nonlinear control of a quadrotor micro-uav

using feedback-linearization," in *IEEE International Conference on Mechatronics*, 2009, pp. 1–6.

[16] G. Cai, B. M. Chen, K. Peng, M. Dong, and T. H. Lee, "Modeling and control of the yaw channel of a uav helicopter," *IEEE transactions on industrial electronics*, vol. 55, no. 9, pp. 3426–3434, 2008.

[17] D. H. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *42nd IEEE conference on Decision and control*, vol. 4, 2003, pp. 3621–3626.

[18] G. S. Avellar, G. A. Pereira, L. C. Pimenta, and P. Iscold, "Multi-uav routing for area coverage and remote sensing with minimum time," *Sensors*, vol. 15, no. 11, pp. 27 783–27 803, 2015.

[19] M. Darrah, E. Fuller, T. Munasinghe, K. Duling, M. Gautam, and M. Wathen, "Using genetic algorithms for tasking teams of raven uavs," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 361–371, 2013.

[20] S. Sariel and T. Balch, "Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments," in *Proceedings of the AAAI-05 Workshop on Integrating Planning into Scheduling*. AAAI Palo Alto, CA, 2005, pp. 27–33.

[21] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *American Control Conference*, vol. 3, 2002, pp. 1936–1941.

[22] A. Richards and J. How, "Mixed-integer programming for control," in *American Control Conference*, 2005, pp. 2676–2683.

[23] S. CPLEX, "Ilog," *Inc., Armonk, NY*, 2003.

[24] I. I. CPLEX, "V12. 1: Users manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[25] S. Malys, "The wgs84 reference frame," *National Imagery and Mapping Agency*, 1996.

[26] R. A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Information processing letters*, vol. 2, pp. 18–21, 1973.

[27] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Info. Pro. Lett.*, vol. 1, pp. 132–133, 1972.

[28] T. M. Chan, "Optimal output-sensitive convex hull algorithms in two and three dimensions," *Discrete & Computational Geometry*, vol. 16, no. 4, pp. 361–368, 1996.

[29] G. T. Toussaint, "Solving geometric problems with the rotating calipers," in *Proc. IEEE Melecon*, vol. 83, 1983, p. A10.

[30] J. O'Rourke, "Finding minimal enclosing boxes," *International journal of computer & information sciences*, vol. 14, no. 3, pp. 183–199, 1985.

[31] H. Freeman and R. Shapira, "Determining the minimum-area encasing rectangle for an arbitrary closed curve," *Communications of the ACM*, vol. 18, no. 7, pp. 409–413, 1975.

[32] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM (JACM)*, vol. 7, no. 4, pp. 326–329, 1960.

[33] R. Fourer, D. M. Gay, and B. W. Kernighan, "A modeling language for mathematical programming," *Management Science*, vol. 36, no. 5, pp. 519–554, 1990.

## BIOGRAPHY

***Vinh K*** *received his Bachelor of Computer Engineering from North Carolina A&T State University, in 2016. He then started his Master studies at the North Carolina A&T State University. His research interests include Coordination and Control of Aerial and Ground Robotics, Flight Control Systems, and Optimal planning and coordination of UAVs. He is a member of Autonomous Cooperative Control of Emergent Systems of Systems (ACCESS) Laboratory and the Autonomous Control and Information Technology (ACIT) Institute.*

***Solomon Gebreyohannes*** *received the Ph.D. degree in electrical engineering from North Carolina A&T State University, Greensboro, NC, USA, in 2016. He is currently a Postdoctoral Research Scholar with the ACCESS Laboratory and the Autonomous Control and Information Technology Institute at North Carolina A&T State University, working on applying machine learning techniques for unmanned autonomous systems applications.*

***Ali Karimoddini*** *is an Associate Professor in the ECE Department at N.C. A&T State University. His research interests include Control and Robotics, Flight Control Systems, Human-machine Interactions, Cyber-physical Systems, Flight Control Systems, and Multi-agent Systems. He is the Director of the ACCESS Laboratory and the Deputy Director of the TECHLAV DoD Center of Excellence in Autonomy. His research has been supported by different federal funding agencies and industrial partners. He is a senior member of IEEE, and a member of AIAA, ISA, and AHS.*