Journal of Intelligent & Robotic Systems
https://doi.org/10.1007/s10846-019-01010-4

Energy-Constrained Multi-UAV Coverage Path Planning for an Aerial Imagery Mission Using Column Generation

Younghoon Choi[1] · Youngjun Choi[1] · Simon Briceno[1] · Dimitri N. Mavris[1]

Received: 2 October 2018 / Accepted: 1 March 2019

Abstract

This paper presents a new Coverage Path Planning (CPP) method for an aerial imaging mission with multiple Unmanned Aerial Vehicles (UAVs). In order to solve a CPP problem with multicopters, a typical mission profile can be defined with five mission segments: takeoff, cruise, hovering, turning, and landing. The traditional arc-based optimization approaches for the CPP problem cannot accurately estimate actual energy consumption to complete a given mission because they cannot account for turning phases in their model, which may cause non-feasible routes. To solve the limitation of the traditional approaches, this paper introduces a new route-based optimization model with column generation that can trace the amount of energy required for all different mission phases. This paper executes numerical simulations to demonstrate the effectiveness of the proposed method for both a single UAV and multiple UAV scenarios for CPP problems.

Keywords Coverage path planning · Multi-UAV missions · Column generation · Energy-constrained optimization

1 Introduction

The Coverage Path Planning (CPP) is a method of building optimal scanning paths in an Area of Interest (AOI) [17, 25]. The CPP approach has been utilized in various robotic applications such as lawn mowing [6], painting [2], cleaning [37], and harvesting [23]. In recent years, the research field of CPPs has been extended to Unmanned Aircraft Systems (UAS) applications such as image mosaicing [16, 36], post-earthquake assessment [31], and 3D terrain reconstruction [8, 34].

✉ Younghoon Choi
younghoon.choi@gatech.edu

Youngjun Choi
ychoi95@gatech.edu

Simon Briceno
briceno@gatech.edu

Dimitri N. Mavris
dimitri.mavris@aerospace.gatech.edu

[1] School of Aerospace Engineering, Georgia Institute of Technology North Avenue, Atlanta, GA 30332, USA

In the context of the path optimization, the typical path planning algorithms are categorized by five groups: geometric methods, stochastic methods, road map methods, potential field methods, and calculus of variation-based classical trajectory optimization methods [9]. For UAS-based imagery missions, classical exact methods, wavefront-based methods, and vehicle-routing-based methods have been successfully solved for CPP problems.

A classical exact method consists of three major processes: decomposing an AOI, selecting a line sweep direction, and backtracking. First, to decompose a whole AOI into multiple subregions, an exact cellular decomposition method is typically used. Two notable decomposition methods are trapezoidal decomposition methods [32] and boustrophedon cell decomposition (BCD) methods [10]. A trapezoidal decomposition method divides the AOI into smaller convex subregions called cells. The BCD method as an extension of the trapezoidal decomposition creates subregions by merging all adjacent cells between two critical points where subregions can be created vertically. Moreover, to address non-polygonal obstacles, a Morse decomposition method can generalize the BCD by employing a Morse function to determine critical points of non-convex obstacles [1]. Second, the line sweep direction has a substantial effect on reducing the total number of turns; a

Published online: 13 March 2019

turning motion generally requires more energy because a vehicle should decelerate, turn, and then accelerate when making a turn [16, 21]. The optimal line sweep direction should be parallel to an edge of the polygon of the environment, which has a minimum value of diameter function for a cell. Once the line sweep direction is determined, the optimal coverage path of each cell can be defined by the back-and-forth motion along the direction. Third, the backtracking algorithm determines the sequence of visiting cells, which is a problem called the Traveling Salesman Problem (TSP). To deal with the TSP of CPP problems, greedy algorithms, dynamic programming approaches, and evolutionary algorithms have been frequently used [25]. A limitation of the classical exact methods for multi-UAV missions is that they require the solution to a high-level decomposition problem to assign subregions to each UAV [28].

A wavefront-based method [38] uses a grid map that describes free space and obstacles in a given AOI. The method requires start/goal points to conduct a path transform approach that propagates a wavefront. The result of propagation determines a value of each grid cell as a weighted sum of both a distance from the goal grid cell and a measure of the degree of discomfort that the closest obstacle exerts. The values of grid cells forms contour patterns that slope towards the goal grid regarded as a numerical potential field. Starting with a pseudo-gradient ascending from the start grid, a coverage path is then constructed. The method has been favorably utilized for UAS-based image mosaicing missions with either a single UAV [30, 36] or multiple UAVs [5]. For a multi-UAV mission, the wavefront-based method requires the decomposition of an entire AOI into multiple subareas for each UAV before applying a wavefront-based method. Another limitation is that the algorithm could build non-optimal paths in an operational point of view even if each path for a UAV is optimal.

A vehicle routing problem (VRP) [35] is generally described with a graph consisting of nodes and arcs. Each node represents either a depot or a customer who has a demand, and arcs depict movements between two nodes. Each customer is visited by a vehicle to handle customer's demand. By solving the VRP, each vehicle is allocated to an ordered arc called a route satisfying the problem's constraints. An advantage of a vehicle-routing-based method is that it deals with multiple vehicles inherently while both classical exact methods and wavefront-based methods require to solve a high-level area decomposition problem for multi-vehicle problems. Because of the less complex structure, vehicle-routing-based methods have been employed to solve multi-UAV CPP problems. However, this vehicle-routing-based methods cannot obtain energy-efficient paths, back-and-forth paths along a line sweep direction. To address this limitation, Avellar et al. formulates a VRP problem with pre-determined nodes on the boundary of an AOI that enables UAVs to make back-and-force motions [3]. Nedjati et al. have also suggested alternative approaches that formulate a VRP problem with enforced flight direction constraints where if a UAV enters a grid, it should exit from the grid in the same horizontal or vertical direction to reduce the number of turns [31]. However, these approaches cannot still be generalized for the more irregular environment.

This paper proposes a novel vehicle-routing-based method for multi-UAV CPP problems that is capable of minimizing the number of UAVs required and total energy required by them to complete a given imagery mission. The proposed method deals with route-based design variables to model the CPP problem rather than the traditional arc-based ones. In order to solve the route-based optimization model, the column generation framework is utilized. The label correcting algorithm also is employed to accelerate the column generation process.

The main contributions of this paper are that: first, the proposed method has the capability of tracing energy consumption of mission profiles for aerial imagery missions consisting of takeoff, cruise, hovering, turning, and landing mission segments. Hence, the method can provide more accurate values of the amount of energy required by each UAV to perform a given multi-UAV mission compared to existing approaches. Second, the proposed method establishes solution routes with a fewer number of turns by dealing with all mission segments during optimization processes. This property of the method enables the construction of more energy-efficient routes for multi-UAV imagery missions.

The remainder of this paper is organized as follows: Section 2 introduces the CPP problem and models it as a UAV mission for multicopters. Section 3 proposes two optimization models for an energy-constrained multi-UAV CPP problem; one is formulated by arc-based design variables while the other is modeled by route-based design variables. Section 4 introduces column generation and utilizes it to solve the optimization model with route-based design variables as well as to impose UAV turns into the model. Section 5 presents numerical simulations to demonstrate effects of dealing with a mission profile including turning phases and impacts of a maximum energy capacity of UAVs on solution routes for CPP problems with multiple UAVs. This paper ends with conclusions in Section 6.

2 Problem Modeling

2.1 Coverage Path Planning Problem

The main purpose of solving a multi-UAV-based CPP problem is to build an optimal coverage route/path for each UAV that depends on a shape of an AOI and a size of sensor footprint as illustrated in Fig. 1. The footprint is determined by an operating altitude and the field of view of a sensor. In the context of the CPP problem, the optimal coverage route is based on back-and-forth routes or spiral routes as shown in Fig. 2, which requires the minimum number of turns since they can minimize total energy consumption [16, 21].

2.2 Estimation of Energy Consumption

In order to execute a UAS imagery mission, a typical mission profile consists of a takeoff, cruise, and hovering to obtain imagery data at each waypoint, turning, and landing. For instance, a mission profile with 3 waypoints is described in Fig. 3. The CPP flying trajectory may include several turning phases right after acquiring imagery data to reach the next waypoint or terminal depot position. For the simplified energy estimation, those five mission segments are categorized into two flight conditions: hovering and forward flight. The hovering phase entails takeoff/landing/turning/image acquisition segments, and the forward flight phase has cruise segment.

For multicopters, hovering power is larger than take-off/landing power at a low climb/descent speed in the normal working state or the windmill brake state [26]. A hovering, also, is required to acquire imagery data as well as to turn its body to head for a next waypoint or a depot. Thus, we use hovering power for mission segments except for cruise one. Based on the momentum theory [26], the power required to hover is described as

$$P_h = T\sqrt{\frac{T}{2\rho A}} = \frac{T^{3/2}}{\sqrt{2\rho A}} = \frac{W^{3/2}}{\sqrt{2\rho A}}, \tag{1}$$

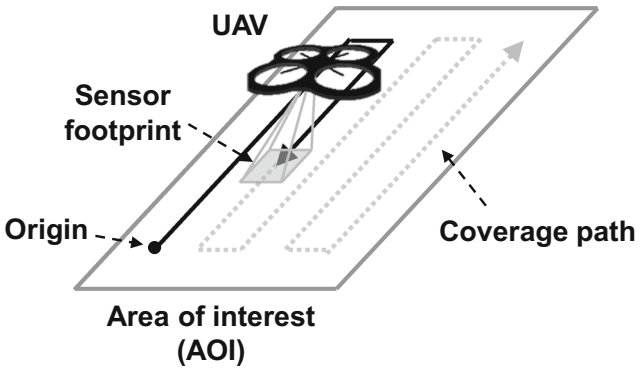


Fig. 1 An illustration of a coverage path planing problem

where T is the thrust required to hover, W is the weight of UAV, ρ is the density of the air, and A is the disk area. For multicopters having n rotors, the ideal power required to hover for each rotor can be calculated by

$$P_{h,(1/n)} = \frac{(W/n)^{3/2}}{\sqrt{2\rho(A/n)}}. \tag{2}$$

To estimate more practical power, the Figure of Merit (FM) is commonly used, if the information is available:

$$FM = \frac{P_{ideal}}{P_{actual}} < 1. \tag{3}$$

In a cruise mission segment, the thrust required in forward flight for each rotor can be calculated as

$$T_{(1/n)} = 2\rho A_{(1/n)} v_i \sqrt{(V_\infty \cos\alpha)^2 + (V_\infty \sin\alpha + v_i)^2}, \tag{4}$$

where V_∞ is a free stream velocity, α is an angle of attack, and v_i is an induced velocity. Then, from Eq. 4, the induced velocity in forward flight becomes

$$v_i = \frac{v_h^2}{\sqrt{(V_\infty \cos\alpha)^2 + (V_\infty \sin\alpha + v_i)^2}}, \tag{5}$$

where $v_h^2 = T_{(1/n)}/2\rho A_{(1/n)}$ is the induced velocity in the hovering phase. Using the advance ratio $\mu = V_\infty \cos\alpha / \Omega R$, where Ω is the rotational speed of a rotor, and R is the rotor radius, the inflow ratio is given by

$$\lambda = \frac{V_\infty \sin\alpha + v_i}{\Omega R} = \frac{V_\infty \sin\alpha}{\Omega R} + \frac{v_i}{\Omega R} = \mu \tan\alpha + \lambda_i, \tag{6}$$

where, λ_i is the induced velocity ratio in forward flight. Then, Eq. 5 can be written as

$$\lambda_i = \frac{\lambda_h^2}{\sqrt{\mu^2 + \lambda^2}} = \frac{C_T}{2\sqrt{\mu^2 + \lambda^2}}, \tag{7}$$

where $\lambda_h = \sqrt{C_T/2}$ from the hovering phase, and C_T is the rotor thrust coefficient, $C_T = T/\rho A(\Omega R)^2$. By substituting (7) for λ_i in Eq. 6, the inflow ratio is given by

$$\lambda = \mu \tan\alpha + \frac{C_T}{2\sqrt{\mu^2 + \lambda^2}}. \tag{8}$$

Once a solution of Eq. 8 is obtained, the power required in forward flight is calculated as

$$P_f = P_h \frac{\lambda}{\lambda_h}. \tag{9}$$

In order to solve (8) to obtain λ, a Newton-Raphson iteration can be used [26]. To use the iterative method, Eq. 8 can be written as an iteration equation with iteration number n such as

$$\lambda_{n+1} = \mu \tan\alpha + \frac{C_T}{2\sqrt{\mu^2 + \lambda_n^2}}. \tag{10}$$

Then, the iteration scheme is given by

$$\lambda_{n+1} = \lambda_n - f(\lambda_n)/f'(\lambda_n), \tag{11}$$

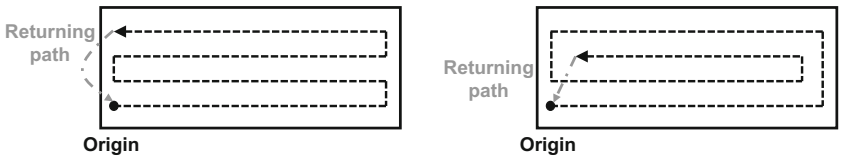


Fig. 2 Preferred routes for the coverage path planning problems

(a) A back-and-forth route (b) A spiral route

where $f(\lambda_n)$ and $f'(\lambda_n)$ are defined as

$$f(\lambda_n) = \lambda_n - \mu \tan\alpha - \frac{C_T}{2\sqrt{\mu^2 + \lambda_n^2}} = 0, \tag{12}$$

and

$$f'(\lambda_n) = 1 + \frac{C_T \lambda_n}{2}(\mu^2 + \lambda_n^2)^{-3/2}, \tag{13}$$

respectively. This iteration stops when

$$\epsilon = \left\| \frac{\lambda_{n+1} - \lambda_n}{\lambda_{n+1}} \right\| < 0.0005 = 0.05\%. \tag{14}$$

3 Optimization Model

A VRP is a kind of Mixed Integer Linear Programming (MILP) problems including both real variables and integer variables with a linear objective function and linear constraints. The VRP determines optimal routes for each vehicle which are modeled by integer variables. Each route is constrained by vehicle characteristics such as its payload capacity and maximum range. The VRP models utilize arcs or routes as design variables to model routes, which are called an arc-based optimization model and a route-based optimization model, respectively. To address CPP problems, we propose an arc-based and energy-constrained optimization model (ABECOM) which is an extension of arc-based and distance-constraint model [7, 24], and then the model is converted to a route-based and energy-constrained optimization model (RBECOM) to impose an effect of turning motions on the energy-based cost function.

3.1 Arc-Based Optimization Model for Energy-Constrained CPP Problems

An ABECOM is described by an input graph describing a vehicle network, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, and a fleet of vehicles, $\mathcal{V} = \{1, \cdots, l\}$. A graph, $\mathcal{G}$, contains a set of nodes, $\mathcal{N}$, and a set of arcs, $\mathcal{A}$: a set of nodes, $\mathcal{N} = \{0, 1, 2, \cdots, n, n+1\}$, consists of the starting depot, 0, the returning depot, $n+1$, and waypoints, $\mathcal{W} = \{1, 2, \cdots, n\}$. Let $\mathcal{N}_O$ be the starting depot and waypoints, $\mathcal{N}_O = \{0, 1, 2, \cdots, n\}$, and $\mathcal{N}_D$ be waypoints and the returning depot, $\mathcal{N}_D = \{1, 2, \cdots, n, n+1\}$, then a set of arcs is defined as $\mathcal{A} = \{(i, j) : \forall i \in \mathcal{N}_O, \forall j \in \mathcal{N}_D, i \neq j\}$. In a UAS imagery mission, the locations of waypoints, $\mathcal{W}$, are selected by image sensor's footprint that depends on both a sensor specifications and a mission altitude. A cost coefficient c_{ij}^k of each arc corresponds with energy required for all operations such as a forward flight from node i to node j by vehicle k, $(i, j) \in \mathcal{A}$ and $k \in \mathcal{V}$, mission operations at node j, $j \in \mathcal{W}$, a takeoff, $i = 0$ and $j \in \mathcal{W}$, and a landing, $i \in \mathcal{W}$ and $j = n+1$. Each vehicle has maximum energy, $\mathcal{E}^k$, available to use during a flight, $k \in \mathcal{V}$. For an arc-based model, a design variable, x_{ij}^k, is defined such that if vehicle k, $k \in \mathcal{V}$, travels along an arc (i, j), $(i, j) \in \mathcal{A}$, then x_{ij}^k is defined as 1, otherwise 0. With these definitions, an ABECOM can be written by

$$\text{Minimize} \quad \sum_{k \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k x_{ij}^k \tag{15}$$

Subject to

$$\sum_{k \in \mathcal{V}} \sum_{j \in \mathcal{N}_D} x_{ij}^k = 1 \quad (\forall i \in \mathcal{W}) \tag{16}$$

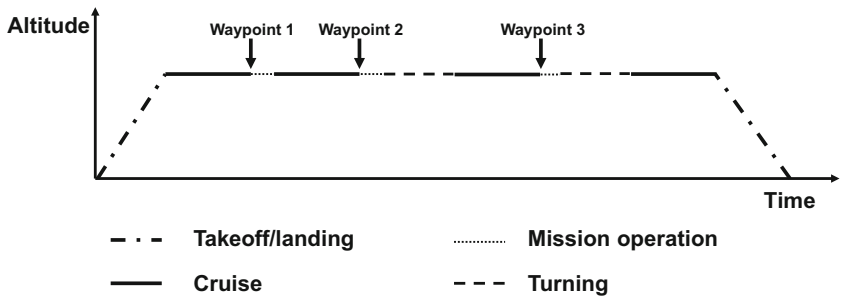


Fig. 3 A mission profile of a UAV for an imagery mission

$$\sum_{j\in\mathcal{N}_D} x_{0j}^k = 1 \quad (\forall k \in \mathcal{V}) \tag{17}$$

$$\sum_{i\in\mathcal{N}_O} x_{ih}^k - \sum_{j\in\mathcal{N}_D} x_{hj}^k = 0 \quad (\forall h \in \mathcal{W},\ \forall k \in \mathcal{V}) \tag{18}$$

$$\sum_{i\in\mathcal{N}_O} x_{i(n+1)}^k = 1 \quad (\forall k \in \mathcal{V}) \tag{19}$$

$$\sum_{j\in\mathcal{N}_D} s_{hj}^k - \sum_{i\in\mathcal{N}_O} s_{ih}^k - \sum_{j\in\mathcal{N}_D} (e_F^k d_{hj}^k + e_M^k t_j^k) x_{hj}^k = 0 \quad (\forall h \in \mathcal{W},\ \forall k \in \mathcal{V}) \tag{20}$$

$$s_{0j}^k = (e_F^k d_{0j}^k + e_M^k t_j^k + e_T^k t_T^k) x_{0j}^k \quad (\forall j \in \mathcal{N}_D,\ \forall k \in \mathcal{V}) \tag{21}$$

$$s_{ih}^k \le (\mathcal{E}^k - e_F^k d_{h(n+1)}^k - e_L^k t_L^k) x_{ih}^k \quad (\forall i \in \mathcal{N}_O,\ \forall h \in \mathcal{W},\ \forall k \in \mathcal{V}) \tag{22}$$

$$s_{i(n+1)}^k \le \mathcal{E}^k x_{i(n+1)}^k \ (\forall i \in \mathcal{N}_O,\ \forall k \in \mathcal{V}) \tag{23}$$

$$s_{hj}^k \ge (e_F^k (d_{0h}^k + d_{hj}^k) + e_M^k (t_h^k + t_j^k) + e_T^k t_T^k) x_{hj}^k \quad (\forall h \in \mathcal{W},\ \forall j \in \mathcal{N}_D, \forall k \in \mathcal{V}) \tag{24}$$

$$x_{ij}^k \in \{0, 1\} \quad (\forall (i, j) \in \mathcal{A},\ \forall k \in \mathcal{V}), \tag{25}$$

where s_{ij}^k is a flow variable that traces energy consumed by vehicle k from the starting depot to node j, $(i, j) \in \mathcal{A}$, $k \in \mathcal{V}$, and d_{ij}^k is a flight distance from node i to node j by vehicle k, $(i, j) \in \mathcal{A}$ and $k \in \mathcal{V}$, and e_F^k is energy per unit distance required by vehicle k, $k \in \mathcal{V}$, for a forward flight, and e_M^k is energy per unit time required by vehicle k, $k \in \mathcal{V}$, for mission operations, and t_j^k/t_h^k is time required by vehicle k for mission operations at node j/h, $j \in \mathcal{N}_D$, $h \in \mathcal{W}$, $k \in \mathcal{V}$. The variable e_T^k/e_L^k is energy per unit time required by vehicle k for a takeoff/landing, and t_T^k/t_L^k is time required by vehicle k for a takeoff/landing respectively.

The objective function of the ABECOM, Eq. 15, is the total energy required for a given mission which is proportional to a total flight distance of the UAVs. Operational strategies for an aerial imaging mission are treated as constraints. Every waypoint should be visited exactly once by a UAV, Eq. 16. Each route should start and end at the depot, Eqs. 17 and 19. If a UAV visits a waypoint, then it should leave for another place, Eq. 18. The Sub-tour elimination condition is modeled as Eq. 20. The initial required energy for each UAV is defined by Eq. 21. The upper bounds of energy consumed at intermediate and final parts of each route are constrained by Eqs. 22 and 23 respectively whereas the lower bound of energy consumed up to node j is restricted by Eq. 24. The integrity constraint of x_{ij}^k is shown in Eq. 25.

3.2 Route-Based Optimization Model for Energy-Constrained CPP Problems

The ABECOM searches an optimal solution with minimum energy consumption to cover an entire AOI. However, imposing turning impacts based on the ABECOM is very challenging because of the limitation of arc-based formulation [7]. Before directly dealing with the limitation, we present the RBECOM comprised of a master problem and subproblems in this subsection, which is an equivalent reformulation of the ABECOM.

A standard MILP problem including n design variables and m constraints can be written as a matrix expression such that

$$\begin{aligned} &\text{Minimize} && \mathbf{c}^T\mathbf{x} && (26)\\ &\text{Subject to} && \mathbf{A}\mathbf{x} \ge \mathbf{b} && (27)\\ &&& \mathbf{x} \ge 0 && (28)\\ &&& \mathbf{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}, && (29) \end{aligned}$$

where $\mathbf{c}^T$ is a cost coefficient vector, $\mathbf{c}^T \in \mathbb{R}^n$, and $\mathbf{A}/\mathbf{b}$ are a constraint matrix/vector, $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{x}$ is a design variable vector consisting of p integer numbers and $n - p$ real numbers. A constraint matrix of a VRP has a block angular structure such that some constraints are applied to whole design variables whereas the others have an effect on a subset of the design variables. The block angular structure system can be decomposed by the Dantzig-Wolfe decomposition (DWD) [11] into two problems: coupled and non-coupled problems. Note that in the ABECOM, only the constraint that every customer is visited by a UAV, Eq. 16, is coupled with vehicles, $\mathcal{V}$. Thus, the ABECOM can be decomposed into a vehicle-coupled problem called a master problem and l vehicle-non-coupled problems called subproblems as shown in Fig. 4. Given $|\cdot|$ is the number of elements of a set, $l = |\mathcal{V}|$.

3.2.1 The Master Problem

The master problem of the ABECOM consists of Eqs. 15, 16, and 25, which is a convex combinatorial optimization problem. The master problem can be reformulated by the Minkowski-Weyl theorem; a matrix $\mathbf{A}$ and a vector $\mathbf{b}$ such that $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \le \mathbf{b}\}$ exist. That is, $\mathcal{S}$ is a polyhedron, if and only if there is $\mathbf{B} \in \mathbb{R}^{n\times p}$ and $\mathbf{C} \in \mathbb{R}^{n\times q}$ such that $\mathcal{S} = conv(\mathbf{B}) + cone(\mathbf{C})$ [22]. In network flow problems, an extreme point of the polytope in its design space corresponds to a route in the network [12]. Thus, the arc-based design variables of the master problem in the ABECOM can be stated as the convex combination of route-based design variables. Let $\mathcal{P}^k$ be a set of feasible routes for vehicle k, $k \in \mathcal{V}$, and x_{ijp}^k be 1 if vehicle k, $k \in \mathcal{V}$, travels along arc (i, j), $(i, j) \in \mathcal{A}$, on route p, $p \in \mathcal{P}^k$, otherwise

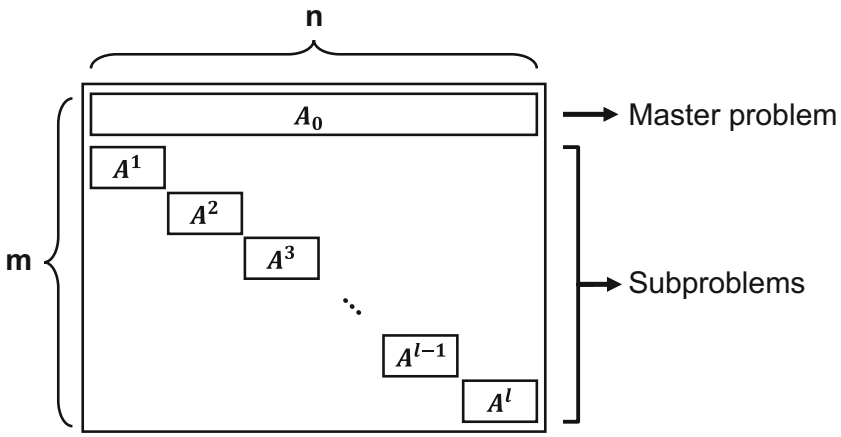


Fig. 4 Block angular structure of a constraint matrix

0. Let y_p^k be the number of times vehicle k, $k \in \mathcal{V}$, travels on route p, $p \in \mathcal{P}^k$. The arc-based design variable x_{ij}^k can be described as

$$x_{ij}^k = \sum_{p\in\mathcal{P}^k} x_{ijp}^k y_p^k \quad (\forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{V}),$$
$$\sum_{p\in\mathcal{P}^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}),$$
$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}^k,\ \forall k \in \mathcal{V}). \tag{30}$$

Using route-based design variables, an integer master problem (IMP) can be expressed as

$$f_{IMP} := \text{Minimize} \sum_{k\in\mathcal{V}} \sum_{p\in\mathcal{P}^k} \sum_{(i,j)\in\mathcal{A}} c_{ij}^k x_{ijp}^k y_p^k \tag{31}$$

Subject to

$$\sum_{k\in\mathcal{V}} \sum_{p\in\mathcal{P}^k} \sum_{j\in\mathcal{N}_D} x_{ijp}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \tag{32}$$
$$\sum_{p\in\mathcal{P}^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \tag{33}$$
$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}^k,\ \forall k \in \mathcal{V}) \tag{34}$$
$$\sum_{p\in\mathcal{P}^k} x_{ijp}^k y_p^k = x_{ij}^k \quad (\forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{V}) \tag{35}$$
$$x_{ij}^k \in \{0,1\} \quad (\forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{V}). \tag{36}$$

From the relation between the arc-based design variables and the route-based design variables, Eq. 30, two convexity constraints are added: an affine constraint, Eq. 33 and a conic constraint, Eq. 34. Linking constraints between the arc-based and route-based variables are stated in Eq. 35. To noticeably address the properties of routes, a cost coefficient for each route, c_p^k, and the number of times vehicle k visits customer i, a_{ip}^k, can be defined as

$$c_p^k := \sum_{(i,j)\in\mathcal{A}} c_{ij}^k x_{ijp}^k \quad (\forall p \in \mathcal{P}^k,\ \forall k \in \mathcal{V}), \tag{37}$$
$$a_{ip}^k := \sum_{j\in\mathcal{N}_D} x_{ijp}^k \quad (\forall i \in \mathcal{N}_O,\ \forall p \in P^k,\ \forall k \in \mathcal{V}). \tag{38}$$

With route-based terms, the IMP of a RBECOM can be rewritten as

$$f_{IMP} := \text{Minimize} \sum_{k\in\mathcal{V}} \sum_{p\in\mathcal{P}^k} c_p^k y_p^k \tag{39}$$

Subject to

$$\sum_{k\in\mathcal{V}} \sum_{p\in\mathcal{P}^k} a_{ip}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \tag{40}$$
$$\sum_{p\in\mathcal{P}^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \tag{41}$$
$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}^k,\ \forall k \in \mathcal{V}) \tag{42}$$
$$\sum_{p\in\mathcal{P}^k} x_{ijp}^k y_p^k = x_{ij}^k \quad (\forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{V}) \tag{43}$$
$$x_{ij}^k \in \{0,1\} \quad (\forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{V}). \tag{44}$$

The IP problem can be solved like LP problems with a branch-and-bound approach that is a framework to systematically solve IP problems using a tree data structure. In a branch-and-bound approach, an IP optimization model is commonly transformed to an LP model using a linear relaxation. We note that the LP solution is not guaranteed to be feasible for the IP model. To address this issue, a branch-and-bound approach provides a means for an optimality check of the solution for its IP model. A LP-relaxed master problem (LMP) is stated as

$$f_{LMP} := \text{Minimize} \sum_{k\in\mathcal{V}} \sum_{p\in\mathcal{P}^k} c_p^k y_p^k \tag{45}$$

Subject to

$$\sum_{k\in\mathcal{V}} \sum_{p\in\mathcal{P}^k} a_{ip}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \tag{46}$$
$$\sum_{p\in\mathcal{P}^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \tag{47}$$
$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}^k,\ \forall k \in \mathcal{V}). \tag{48}$$

By relaxing (44), there is no longer a need of linking constraints, Eq. 43.

3.2.2 The Subproblem

After implementing the DWD, the RBECOM has l identical subproblems which are independent on vehicles, $l = |\mathcal{V}|$. Each subproblem solves a shortest path problem with resource constraints (SPPRC), which is written as

$$c^k := \text{Minimize} \sum_{(i,j)\in\mathcal{A}} (c_{ij}^k - \pi_i)x_{ij}^k - \pi_0^k \tag{49}$$

Subject to

$$\sum_{j\in\mathcal{N}_D} x_{0j}^k = 1 \tag{50}$$

$$\sum_{i\in\mathcal{N}_O} x_{ih}^k - \sum_{j\in\mathcal{N}_D} x_{hj}^k = 0 \quad (\forall h \in \mathcal{W}) \tag{51}$$

$$\sum_{i\in\mathcal{N}_O} x_{i(n+1)}^k = 1 \tag{52}$$

$$\sum_{j\in\mathcal{N}_D} s_{hj}^k - \sum_{i\in\mathcal{N}_O} s_{ih}^k - \sum_{j\in\mathcal{N}_D} (e_F^k d_{hj}^k + e_M^k t_j^k)x_{hj}^k = 0 \quad (\forall h \in \mathcal{W}) \tag{53}$$

$$s_{0j}^k = (e_F^k d_{0j}^k + e_M^k t_j^k + e_T^k t_T^k)x_{0j}^k \ (\forall j \in \mathcal{N}_D) \tag{54}$$

$$s_{ih}^k \le (\mathcal{E}^k - e_F^k d_{h(n+1)}^k - e_L^k t_L^k)x_{ih}^k \quad (\forall i \in \mathcal{N}_O, \ \forall h \in \mathcal{W}) \tag{55}$$

$$s_{i(n+1)}^k \le \mathcal{E}^k x_{i(n+1)}^k \ (\forall i \in \mathcal{N}_O) \tag{56}$$

$$s_{hj}^k \ge (e_F^k (d_{0h}^k + d_{hj}^k) + e_M^k (t_h^k + t_j^k) + e_T^k t_T^k)x_{hj}^k \quad (\forall h \in \mathcal{W}, \ \forall j \in \mathcal{N}_D) \tag{57}$$

$$x_{ij}^k \in \{0, 1\} \quad (\forall (i, j) \in \mathcal{A}), \tag{58}$$

where π_i and π_0^k are dual variables from Eqs. 46 and 47 respectively. Constraints (50)–(52) describe vehicle flow from/to the depot. Subtour elimination constraint can be expressed by Eq. (53). Energy consumed by each vehicle is constrained by Eqs. 54–57. The integrity constraints of x_{ij}^k are stated in Eq. 58.

4 Column Generation of the RBECOM for CPP Problems

The VRP is typically modeled by a MILP optimization model. Column generation is an approach to systematically obtain a solution of an LP problem. Its fundamental ideas are presented with the DWD [11] which decomposes an optimization model with a block-angular-structural constraint matrix into a master problem and subproblems as described in Section 3.2. When solving practical problems using column generation, a physical meaning of each extreme point needs to be clear. For instance, the first practical problem solved by column generation is a cutting stock problem [19, 20]. Each extreme point of the cutting stock problem indicates a cutting pattern that is how to cut a stock into pieces of specified size. In network flow problems, each extreme point means a flow pattern, a path/route. With this physical meaning, the column generation has been utilized to solve network flow problems such as urban transit crew scheduling problems [14] and vehicle routing problems with time windows [13]. Recently, the column generation becomes a main approach to handle huge IP problems in logistics and operational research areas [4]. For UAS-based surveillance missions, the VRPs have been solved by using the column generation method [29, 39].

The extreme point of the CPP problem represents a route that is used as a design variable, y_p^k, in the master problem. The number of y_p^k depends on both the number of vehicles, $|\mathcal{V}|$, and routes, $|\mathcal{P}|$. The RBECOM could need more computing power than the ABECOM even if solving the same problem. For instance, let's assume a simple network topology illustrated in Fig. 5. The objective of this problem is to find an optimal route from the origin to the destination. To handle this problem, the ABECOM calls for 10 design variables while an RBECOM requires 20 ones. This is because the network topology consists of 10 arcs whereas the number of feasible combinations in the given topology is 20.

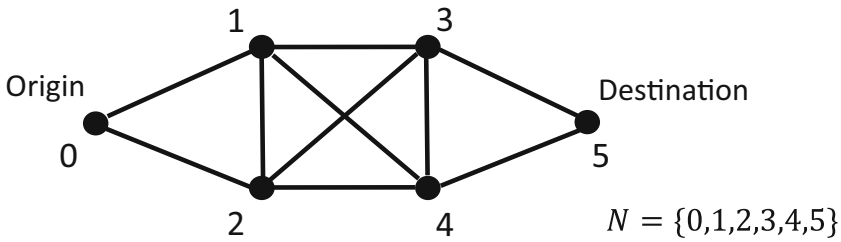


Fig. 5 A simple network topology consisting of 6 nodes and 10 edges

To address this computational drawback, one approach is to solve a route-based optimization problem including a subset of routes instead of the whole possible routes such that $\mathcal{P}' \subset \mathcal{P}$ which is called a restricted problem. Let P_0' be the initial subset of routes, and P_i' be a subset of routes for i-th iteration as illustrated in Fig. 6. The column generation expands the design space of design variables on every iteration by solving a restricted LP-relaxed master problem and its l subproblems. It stops based on the existence of profitable routes. The column generation framework is shown in Fig. 7. Each step is described below.

4.1 Initialization

The initial step of the column generation creates initial solution routes for a CPP problem which are a subset of feasible extreme points. Finding initial feasible routes is commonly solved by two methods: the trivial solution method and the savings algorithm [18]. The trivial solution method creates routes such that each route visits just one waypoint (depot-waypoint-depot routes). The savings algorithm is a greedy method as a simple heuristic approach. Note that the number of initial feasible routes can be used to fix the number of vehicles, $|\mathcal{V}|$. In the RBECOM, the number of vehicles determines the number of subproblems.

4.2 Master Problem

The master problem can be modeled by either a set partitioning problems or a set covering problem that is a relaxation of a set partitioning problem [4, 11]. The master problem as a set partitioning problem only considers elementary routes such that each waypoint is visited exactly once. In contrast, the master problem as a set covering problem for CPP problems allows non-elementary routes where each waypoint is visited at least once. In order to find an exact solution of a CPP problem, this authors employ the set partitioning master problem.

The column generation works with a restricted master problem having a subset of routes, $\mathcal{P}' \subset \mathcal{P}$. The solution routes from the initialization step are injected to a restricted integer master problem (RIMP) as an initial subset or routes,

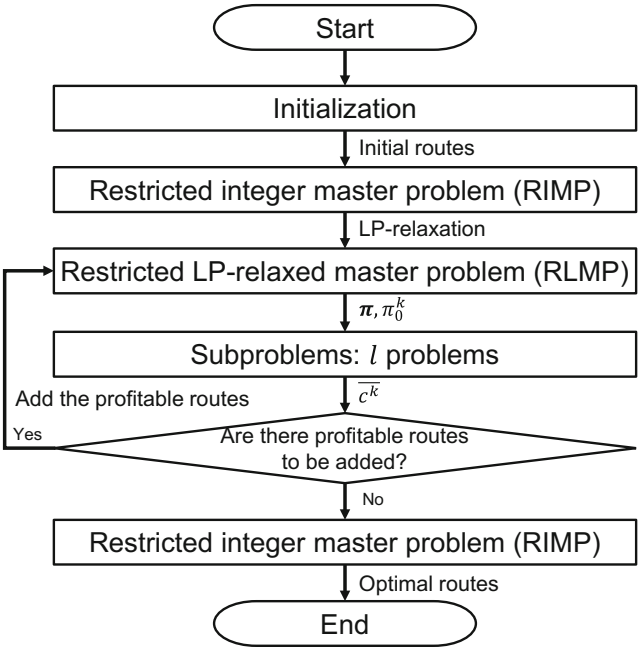


Fig. 7 Column generation framework

$\mathcal{P}'_0 \subset \mathcal{P}$. The RIMP is defined as

$$f_{RIMP} := \text{Minimize} \sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}'^k} c_p^k y_p^k \tag{59}$$

Subject to

$$\sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}'^k} a_{ip}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \tag{60}$$

$$\sum_{p \in \mathcal{P}'^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \tag{61}$$

$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}'^k,\ \forall k \in \mathcal{V}) \tag{62}$$

$$\sum_{p \in \mathcal{P}'^k} x_{ijp}^k y_p^k = x_{ij}^k \quad (\forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{V}) \tag{63}$$

$$x_{ij}^k \in \{0,1\} \quad (\forall (i,j) \in \mathcal{A},\ \forall k \in \mathcal{V}). \tag{64}$$

In order to solve the IMP, Eqs. 39–44, by using the RIMP, Eqs. 59–64, there is a need to find profitable routes for decreasing the objective function value. The process to find

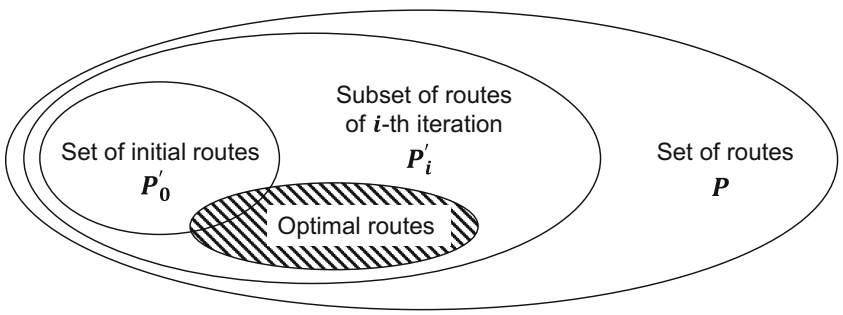


Fig. 6 Change in the domain of routes during the column generation process

the routes is similar with iterations of the simplex method using a non-basic variable to price out and enter the basis. In order for a pricing process of the column generation, the restricted LP-relaxed mater problem (RLMP) is exploited, which is defined as

$$f_{RLMP} := \text{Minimize} \sum_{k\in\mathcal{V}} \sum_{p\in\mathcal{P}'^k} c_p^k y_p^k \tag{65}$$

Subject to

$$\sum_{k\in\mathcal{V}} \sum_{p\in\mathcal{P}'^k} a_{ip}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \tag{66}$$

$$\sum_{p\in\mathcal{P}'^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \tag{67}$$

$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}'^k,\ \forall k \in \mathcal{V}). \tag{68}$$

The dual variables of the RLMP are used to calculate the reduced cost of routes in the subproblem to be introduced below.

4.3 Subproblem

The main purpose of the subproblem is to find profitable routes by calculating the reduced cost for each route. Because the RLMP is a set partitioning problem, the subproblem essentially works with only elementary paths, which is called an elementary shortest path problem with resource constraints (ESPPRC) that is a special case of the SPPRC. After applying the DWD, l subproblems are defined. The non-negative dual variables of the RLMP, $\boldsymbol{\pi}$ and $\pi_0^k,\ k \in \mathcal{V}$, enable each subproblem to determine that

$$\bar{c}^k := \text{Minimize}\{(\mathbf{c}^{kT} - \boldsymbol{\pi}^T \mathbf{A}^k)\mathbf{x}^k - \pi_0^k\}, \tag{69}$$

which is the matrix expression of the objective function of the subproblem, Eq. 49. The affine constraint, Eq. 67, can decompose into $\sum_{p\in\mathcal{P}'^k} y_p^k \geq 1$ and $\sum_{p\in\mathcal{P}'^k} y_p^k \leq 1$. Then, $\sum_{p\in\mathcal{P}'^k} y_p^k \leq 1$ enables to have both an upper bound and a lower bound on f_{LMP}, Eq. 45, such that

$$f_{RLMP} + \sum_{k\in\mathcal{V}} \bar{c}^k \leq f_{LMP} \leq f_{RLMP}, \tag{70}$$

where $f_{RLMP} = \boldsymbol{\pi}^T \boldsymbol{b} + \sum_{k\in\mathcal{V}} \pi_0^k$ by the LP duality. In each iteration, f_{RLMP} can decrease up to the aggregate of the smallest reduced costs $\bar{c}^k$ for each subproblem, $k \in \mathcal{V}$ [12]. Because of the condition, $f_{LMP} \leq f_{IMP}$, $f_{RLMP} + \sum_{k\in\mathcal{V}} \bar{c}^k$ is a lower bound on f_{IMP}. If $\bar{c}^k \geq 0$, there is no negative reduced cost column, which means there is no profitable route for vehicle k. At an optimal point, $\bar{c}^k = 0$ for all $k \in \mathcal{V}$, $f_{LMP} = f_{RLMP}$.

4.4 Stop Criterion

If there is no profitable route for all vehicles, $\bar{c}^k \geq 0$ for all $k \in \mathcal{V}$, the iterative process terminates. Otherwise, the routes having a negative reduced cost are added to the RLMP as new columns of the constraints matrix. For instance, suppose the RLMP has p routes at the i-th iteration. After updating the master problem in the $(i+1)$-th iteration, it has p' routes, $p' > p$, as described in Fig. 8. After the iterative process terminates, all routes of the RLMP are transferred to the the RIMP.

4.5 Label Correcting Algorithm and Turning Effects

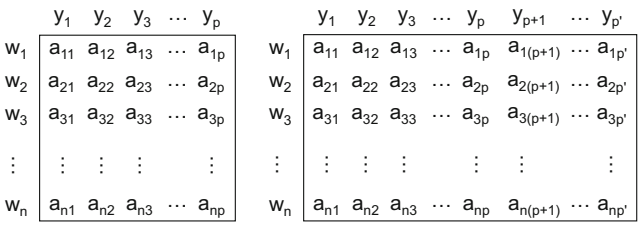
In order to accelerate the column generation process, the label correcting algorithm [15] can be utilized to solve subproblems, which finds all routes with a negative reduced cost in the subproblem. The algorithm is a dynamic programming method that divides a given problem into simpler subproblems in a recursive manner. In general approaches, even if column generation works with route-based variables, arc-based variables needs to be used to reorganize routes because the list of waypoints visited can be obtained from a_{ip}^k and y_p^k, but the order of them cannot be traced without arc-based variables. When utilizing the label correcting algorithm, the RLMP can be solved based on just route-based variables by storing path information in each label. This algorithm can add turning penalties to cost coefficients of the RLMP and subproblems while the linearity of the RLMP remains [7].

In the label correcting algorithm, a label of vehicle k, $k \in \mathcal{V}$, at node z, $z \in \mathcal{N}$, can be defined as

$$\mathbf{L}_z^k = (\mathbf{R}_z^k, \mathbf{Q}_z^k, C_z^k),\ (\forall z \in \mathcal{N}, \forall k \in \mathcal{V}), \tag{71}$$

where $\mathbf{Q}_z^k$ is a ordered list of arcs the label moves, and C_z^k is a route cost along the label, and $\mathbf{R}_z^k$ is resources used by vehicle k, $k \in \mathcal{V}$, from a depot to node z, $z \in \mathcal{N}$ which is given by

$$\mathbf{R}_z^k = (E_z^k, \mathbf{U}^k = [U_0^k, \cdots, U_{n+1}^k], \sum_{z\in\mathcal{N}} U_z^k), \tag{72}$$



(a) The i-th iteration (b) The (i+1)-th iteration

Fig. 8 Shapes of constraint matrix at the i-th and the (i+1)-th iterations of column generation

where $\mathbf{U}^k$ is a vector of unreachable nodes of vehicle k: if node i is already visited, $u_z^k = 1$. Otherwise, $u_z^k = 0$. In addition, E_z^k is energy used by vehicle k, $k \in \mathcal{V}$, from a depot to node z, $z \in \mathcal{N}$, which is given by

$$E_z^k = \sum_{(0,j)\in\mathbf{Q}_z^k} e_T^k t_T^k + \sum_{(i,j)\in\mathbf{Q}_z^k} e_F^k d_{ij}^k + \sum_{(i,h),(h,j)\in\mathbf{Q}_z^k} T_{ihj}^k + e_M^k t_h^k + \sum_{(i,n+1)\in\mathbf{Q}_z^k} e_L^k t_L^k, \tag{73}$$

where T_{ihj}^k is energy required to turn a vehicle k into a direction of arc (h, j) from a direction of arc (i, h). Using energy per unit time required by vehicle k, $k \in \mathcal{V}$, to turn, e_R^k, T_{ihj}^k can be calculated by

$$T_{ihj}^k = e_R^k \frac{\theta_{ihj}^k}{r^k}, \tag{74}$$

where θ_{ihj}^k is the angle between arc (i, h) and arc (h, j) for vehicle k, and r^k is a turning rate of vehicle k.

The label correcting algorithm is working with a partial order of resources to accelerate run time. Only non-dominated labels are propagated to neighbor nodes. When $C_z^k = E_z^k$, the dominance rule can be described as shown in Algorithm 1. The main idea is that when there are labels which visit same nodes, only non-dominated labels are transmitted to next iterations. For the whole procedures of the label correcting algorithm and numerical examples, refer [7, 15].

Algorithm 1 Pseudo code for evaluating dominance

Input: L_{z1}^{k1}, L_{z2}^{k2}
Output: $True$ if L_{z1}^{k1} is dominated by L_{z2}^{k2}, otherwise $False$
if $(k_1 \neq k_2)$ or $(\sum_{z\in\mathcal{N}} U_{z1}^{k1} \neq \sum_{z\in\mathcal{N}} U_{z2}^{k2})$ **then**
  return $False$
else
  for i in $\mathcal{N}$ **do**
    if $U_i^{k_1} \neq U_i^{k_2}$ **then**
  return $False$
    end if
  end for
  if $E_{z1}^{k1} \leq E_{z2}^{k2}$ **then**
  return $False$
  else
  return $True$
  end if
end if

5 Numerical Simulation

In this section, we discuss numerical simulations to demonstrate the performance of both the ABECOM and the RBECOM with respect to energy consumption. To account for energy required for a given scenario, the UAS platform is assumed to be a DJI Frame Wheel 450 with 9.4-inch rotors whose mass is 3.5 lb. We also assume that each vehicle moves at speed 10 ft/sec with an angle of attack of 5 degrees. Its figure of merit (FM) is assumed to be 0.6 during the mission based on analogous systems [33]. With these values, the power coefficients are calculated such that $P_h = 114.00$ W/sec, and $P_f = 109.74$ W/sec. To address time required for mission operations, each vehicle spends 2 seconds to acquire imagery data at each waypoint. After that, the vehicle turns at speed 22.5 deg/sec to head into its next waypoint if needed. Each vehicle also requires 5 seconds for either a takeoff or a landing.

The numerical simulations consist of two experiments: the first experiment is to compare the optimal routes from the ABECOM and the RBECOM with a single UAV scenario. The second experiment is to discuss the optimal area decomposition of a given AOI in multi-UAV scenarios. All the numerical simulations are executed with Intel® Core™ i7-7700HQ processor and 32 Gb memory. In addition, Gurobi 8.0 is used as a MIP solver to obtain solutions from the optimization models.

5.1 Single-UAV CPP Scenarios

The main objective of the single-UAV missions is to compare two optimal routes with respect to energy consumption. Note that the total energy of the ABECOM is computed as a post-process. In the single-UAV missions, two AOIs are considered as example cases:a L-shape AOI and a non-convex-shape one. In the examples, the length of the square grid cell is assumed to be 10 ft.

The simulation results are summarized in Table 1. As we expected, the solutions of the ABECOM requires less energy for cruise than that of the RBECOM. This result implies the ABECOM finds better routes in terms of minimizing flight distance. When considering energy required for a turn, however, the solutions of the ABECOM have more turns than that of the RBECOM. Consequently, the total energy of the ABECOM becomes larger than that of the RBECOM. Each solution route of both models is visualized in Fig. 9. In each figure, the blue area is the given AOI, and each UAV moves from/to the depot grid. The routes of the ABECOM have short returning paths, that is, the last grid a UAV takes imagery data is an adjacent grid of the depot grid. On the other hand, the routes of the RBECOM seem to have a combination of back-and-forth and spiral paths that are known for energy-efficient paths.

Table 1 Effects of optimization models on single-UAV CPP scenarios

AOI	Model	Required energy (W)					Number	Solution
		Takeoff/landing	Cruise	Mission Ops.	Turning	Total	of turns	route
L-shape	ABECOM	1140.00	1755.84	3420.00	4104.00*	10419.84*	9	Fig. 9a
	RBECOM	1140.00	2041.77	3420.00	2565.30	9167.08	5	Fig. 9b
Non-convex-shape	ABECOM	1140.00	2788.96	5472.00	5244.00*	14644.96*	11	Fig. 9c
	RBECOM	1140.00	2944.15	5472.00	4332.00	13888.15	9	Fig. 9d

∗: the value is calculated by a post-processing based on its route

Even though solutions of the RBECOM have long returning paths as illustrated in Fig. 9a and b, their amount of total energy required to fully execute the given mission is less than that of the ABECOM because of the less number of turns.

The RBECOM creates an optimal solution by minimizing both the length of a returning path and the number of turns. Thus, the solution route depends on the relative magnitude of cruise energy and turning energy. The simulation results summarized in Table 2 show that the turning rate and the cruise distance are in inverse proportion to each other. It means that when it has low turning rates, the optimal route tends to have less turning angles to minimize the total energy rather than decreasing the cruise distance. The result is more clearly shown by their shapes of solution routes as illustrated in Fig. 10. The solution routes of low-turning-rate scenarios in Fig. 10a and b visually illustrate that they consist of a combination of back-and-forth and spiral routes to reduce the number of turns and total turning angle.

On the other hand, the solution route with high-turning-rate scenarios shown in Fig. 10a and b seems like the optimal route of the ABECOM that minimizes the flight distance without considering turning effects. This response is expected because the objective of RBECOM with high-turning-rate scenarios is to minimize the flight distance. In summary, the dominant factor of the RBECOM is determined by relative quantities of the turning rate and the flight distance which come from the given scenario.

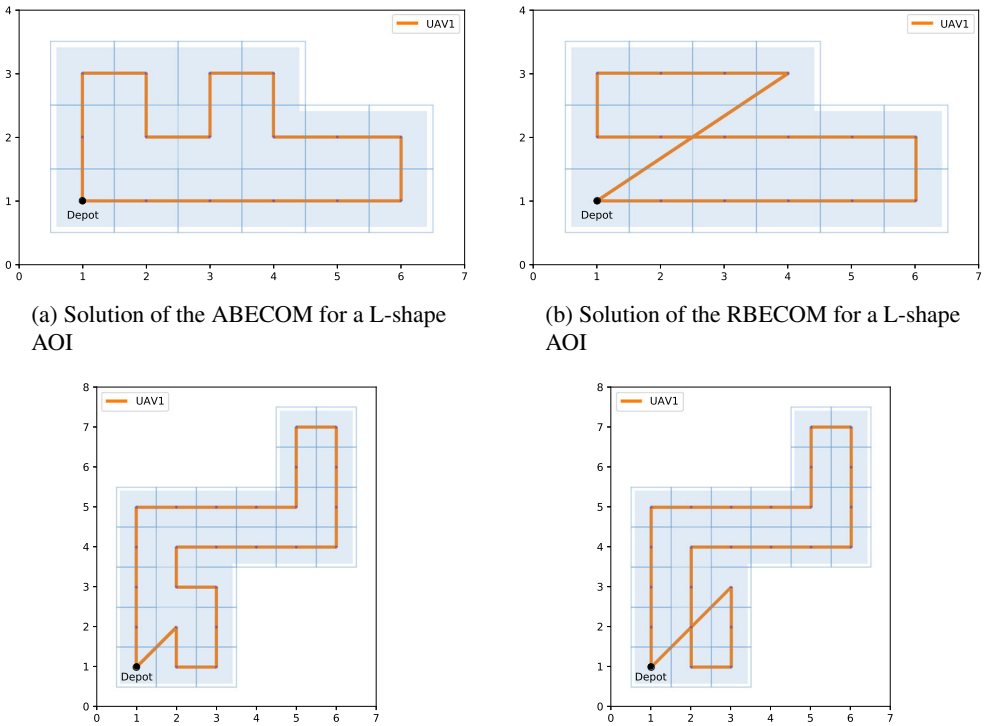


Fig. 9 Comparison of solution routes of both the ABECOM and the RBECOM

(a) Solution of the ABECOM for a L-shape AOI

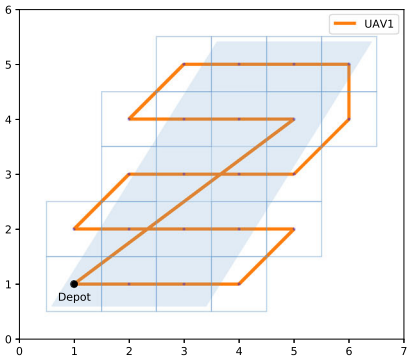
(b) Solution of the RBECOM for a L-shape AOI

(c) Solution of the ABECOM for a non-convex-shape AOI

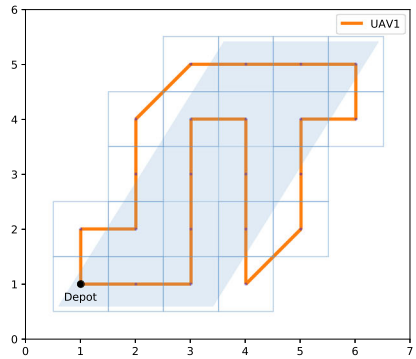
(d) Solution of the RBECOM for a non-convex-shape AOI

Table 2 Effects of turning rate on single-UAV CPP scenarios

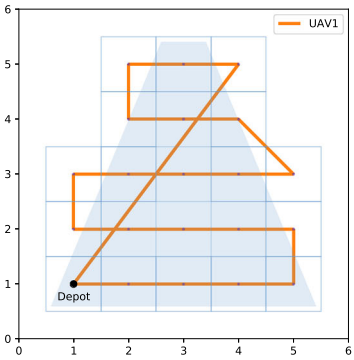
AOI	Turning rate (deg./sec.)	Required energy (W)			Number of turns	Total turning angle (deg.)	Solution route
		Cruise	Turning	Total			
Parallelogram-shape	22.50	3035.06	4373.19	13336.26	10	863.13	Fig. 10a
	45.00	2505.19	2508.00	10941.19	12	990.00	Fig. 10b
Trapezoid-shape	45.00	2788.96	2145.40	10634.36	9	846.87	Fig. 10c
	90.00	2350.00	1368.00	9418.00	13	1080.00	Fig. 10d



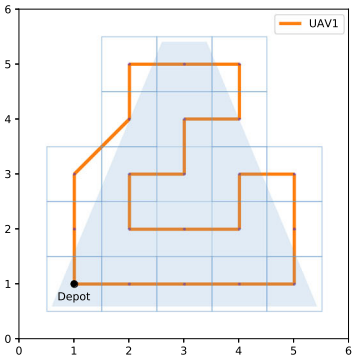
(a) Solution route for a parallelogram-shape AOI with turning rate 22.50 deg./sec.



(b) Solution route for a parallelogram-shape AOI with turning rate 45.00 deg./sec.



(c) Solution route for a trapezoid-shape AOI with turning rate 45.00 deg./sec.



(d) Solution route for a trapezoid-shape AOI with turning rate 90.00 deg./sec.

Fig. 10 Effects of turning rate on solution routes of the RBECOM

Table 3 Effects of maximum energy capacities of UAVs on multi-UAV CPP scenarios

AOI	Maximum energy capacity (W)	Required energy (W)						Number of UAVs	Solution routes
		UAV1	UAV2	UAV3	UAV4	UAV5	All routes		
Li's shape	6000	4594.79	5812.60	3764.30	5193.51	5635.99	25001.19	5	Fig. 11a
	7500	5356.60	6636.22	4285.03	7396.17		23674.01	4	Fig. 11b
	8000	6488.08	7376.01	7989.23			21853.32	3	Fig. 11c
	11000	10604.85	10103.47				20708.32	2	Fig. 11d

5.2 Multi-UAV CPP Scenarios

In the context of a large-scale CPP, a single vehicle may not completely cover the given AOI. In this case, the area decomposition of the AOI is required to define small subareas that can be covered by a UAV. However, the vehicle-routing-based method can solve the large AOI mission regardless of size and shape of the given AOI. In other words, this method does not require any area subdivisions. In most multi-UAV missions, an operational plan with fewer UAVs is preferred because it generally requires less operating cost to perform a given mission. In order to reflect this concept, we add a fixed large number, M, into the route energy of the RBECOM, Eq. 73, as follows

$$E_z^k = M + \sum_{(0,j)\in\mathbf{Q}_z^k} e_T^k t_T^k + \sum_{(i,j)\in\mathbf{Q}_z^k} e_F^k d_{ij}^k + \sum_{(i,h),(h,j)\in\mathbf{Q}_z^k} T_{ihj}^k + e_M^k t_h^k + \sum_{(i,n+1)\in\mathbf{Q}_z^k} e_L^k t_L^k. \quad (75)$$

By adding M, the master problem looks for a solution requiring fewer number of routes to complete a given mission first. Then, the master problem finds a combination of routes that requires the least amount of total energy. This approach prevents the RBECOM from increasing the required number of UAVs when the given AOI can be entirely covered by fewer UAVs.

The numerical simulations of multi-UAS imagery missions are executed to demonstrate the effects of the maximum energy capacity on both the total energy required to complete the mission and the number of required UAVs. In order to perform multi-UAV CPP simulations, the shape of Li's AOI [27] is utilized with homogeneous UAVs. The simulation results are summarized in Table 3. The results present that as the maximum energy capacity increases, the amount of energy required decreases as well as fewer UAVs can complete the given mission. The solution routes for each simulation are illustrated in Fig. 11. Each solution is feasible for the given constraints. The result of area decomposition in Fig. 11c is intuitively understandable, which vertically divides the AOI into 3 subareas. However, in the other three results, it is hard to find patterns for area decompositions. This behavior is expected because the RBECOM decomposes the AOI based on the vehicle's maximum energy capacity rather than its shape. This result implies that the traditional area decompositions based on a configuration of a given AOI could lead a non-optimal solution for multi-UAV missions. In this context,

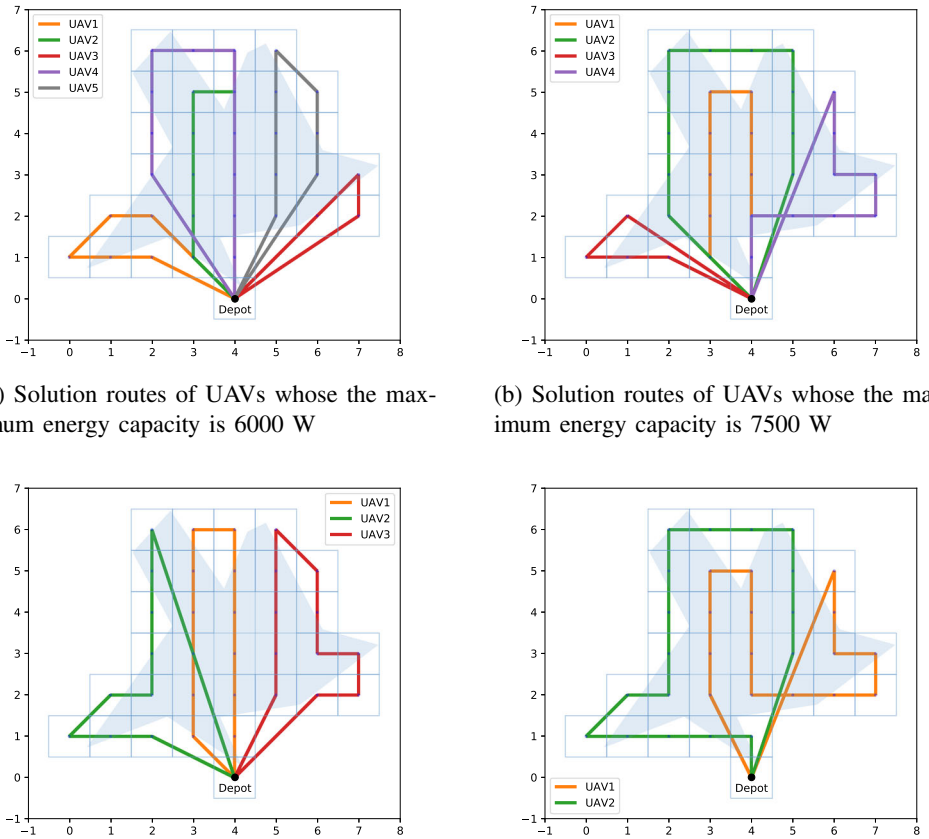


Fig. 11 Solution routes of multi-UAV CPP scenarios

(a) Solution routes of UAVs whose the maximum energy capacity is 6000 W

(b) Solution routes of UAVs whose the maximum energy capacity is 7500 W

(c) Solution routes of UAVs whose the maximum energy capacity is 8000 W

(d) Solution routes of UAVs whose the maximum energy capacity is 11000 W

the vehicle-routing-based methods have an advantage to achieve more energy-efficient solution.

6 Conclusion

This paper introduces a novel vehicle-routing-based method to solve an energy-constrained multi-UAV CPP problem. The proposed method converts the CPP problem into a MILP problem with energy constraints. The MILP problem is reformulated by the DWD to define a master problem and subproblems; the master problem determines an optimal combination of routes in terms of the amount of energy required to perform a given mission while the subproblems specify profitable routes that could reduce the total energy required. These two problems are integrated into a column generation. The column generation process entails a label correcting algorithm to identify profitable routes rather than the best route of each subproblem. The algorithm additionally works based on energy consumption calculated by mission profiles including turning phases which cannot be reflected by the conventional VRP models. In numerical simulations with single UAV CPP missions, the RBECOM generates the solution that has more energy-efficient routes. Its energy consumption decreases up to 12.02% and 5.17% compared to that of the ABECOM in the given AOIs respectively: the L-shape AOI and the non-convex one. In multi-UAV simulation scenarios, the RBECOM decomposes the given AOI into subareas to generate complete coverage paths for each UAV. The result of the area decomposition is based on the maximum energy capacity of UAVs rather than the shape of the AOI. The potential extension of this work can be the improvement of energy model based on flight dynamics and actual physical flight experiments.

Acknowledgements This paper is a major enhancement of the ICUAS 2018 accepted paper.

References

1. Acar, E., Choset, H., nad, P.N., Atkar, A.R., Hull, D.: Morse decompositions for coverage tasks. Int. J. Robot. Res. **21**(4), 331–344 (2002)
2. Atkar, P., Greenfield, A., Conner, D., Choset, H., Rizzi, A.: Uniform coverage of automotive surface patches. Int. J. Robot. Res. **24**(11), 87–102 (1988)
3. Avellar, G.S.C., Pereira, G.A.S., Pimenta, L.C.A., Iscold, P.: Multi-UAV routing for area coverage and remote sensing with minimum time. Sensors **15**, 27783–27803 (2015)
4. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H.: Branch-and-price: column generationn for solving huge integer probrams. INFORMS **46**(3), 316–329 (1998)
5. Barrientos, A., Colorado, J., del Cerro, J., Martinez, A., Rossi, C., Sanz, D., Valente, J.: Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. J. Field Robot. **28**(5), 667–689 (2011)
6. Cao, Z., Huang, Y., Hall, E.: Region filling operations with random obstacle avoidance for mobile robotics. J. Robot. Syst. **5**(2), 87–102 (1988)
7. Choi, Y., Choi, Y., Briceno, S., Mavris, D.N.: Coverage path planning for a UAS imagery mission using column generation with a turn penalty. In: The 2018 International Conference on Unmanned Aircraft Systems. Dallas (2018)
8. Choi, Y., Choi, Y., Briceno, S., Mavris, D.N.: Three-dimensional uas trajectory optimization for remote sensing in an irregular terrain environment. In: The 2018 International Conference on Unmanned Aircraft Systems. Dallas (2018)
9. Choi, Y., Jimenez, H., Mavris, D.N.: Two-layer obstacle collision avoidance with machine learning for more energy-efficient unmanned aircraft trajectories. Robot. Auton. Syst. **98**, 158–173 (2017)
10. Choset, H., Pignon, P.: Coverage path planning: the boustrophedon decomposition. In: Proceedings of the International Conference on Field and Service Robotics. Canberra (1997)
11. Dantzig, G.B., Wolfe, P.: Decomposition principle for linear programmings. Oper. Res. **8**, 101–111 (1960)
12. Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds.): Column Generation. Springer, Berlin (2005)
13. Desrochers, M., Desposiers, J., Solomon, M.: A new optimization algorithm for the vehicle routing problem with time windows. Oper. Res. **40**(2), 342–354 (1992)
14. Desrochers, M., Soumis, F.: A column generation approach to the urban transit crew scheduling problem. Transp. Sci. **23**(1), 1–13 (1989)
15. Feillet, D., Dejax, P., Gendreau, M., Gueguen, C.: An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problem. Networks **44**(3), 216–229 (2004)
16. Franco, C.D., Buttazzo, G.: Coverage path planning for UAVs photogrammetry with energy and resolution constraints. J. Intell. Robot. Syst. **83**, 445–462 (2016)
17. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. Robot. Auton. Syst. **61**, 1258–1276 (2013)
18. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. **12**(4), 568–581 (1964)
19. Gilmore, P.C., Gomory, R.E.: A linear programming approach to the cutting-stock problem. Oper. Res. **9**(6), 849–859 (1961)
20. Gilmore, P.C., Gomory, R.E.: A linear programming approach to the cutting-stock problem-part 2. Oper. Res. **11**(6), 863–888 (1963)
21. Huang, W.H.: Optimal line-sweep-based decompositions for coverage algorithms. In: Proceedings of the 2001 IEEE International Conference on Robotics & Automation. Seoul (2001)
22. Jünger, M., Liebling, T., Naddef, D., Nemhauser, G., Pulleyblank, W., Reinelt, G., Rinaldi, G., Wolsey, L. (eds.): 50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art. Springer, Berlin (2010)
23. Jin, J., Tang, L.: Optimal coverage path planning for arable farming on 2D surfaces. Trans. ASABE **53**(1), 283–295 (2010)
24. Kara, I.: Arc based integer programming formulations for distance constrained vehicle routing problem. In: LINDI 2011 - 3rd IEEE International Symposium on Logistics and Industrial Informatics. Budapest (2011)

25. Khan, A., Noreen, I., Habib, Z.: On complete coverage path planning algorithms for non-holonomic mobile robots: survey and challenges. J. Inf. Sci. Eng. **33**, 101–121 (2017)
26. Leishman, J.G. (ed.): Principles of Helicopter Aerodynamics, 2nd edn. Cambridge University Press, Cambridge (2006)
27. Li, Y., Chen, H., Er, M.J., Wang, X.: Coverage path planning for UAVs based on enhanced exact cellular decomposition method. Mechatronics **21**, 876–885 (2011)
28. Maza, I., Ollero, A.: Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In: Alami, R., Chatila, R., Asama, H. (eds.) Distributed Autonomous Robotic Systems, vol. 6, pp. 221–230. Springer Japan, Tokyo (2007)
29. Mufalli, F., Batta, R., Nagi, R.: Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans. Comput. Oper. Res. **39**, 2787–2799 (2012)
30. Nam, L.H., Huang, L., Li, X.J., Xu, J.F.: An approach for coverage path planning for UAVs. In: 2016 IEEE 14th International Workshop on Advanced Motion Control (AMC) (2016)
31. Nedjati, A., Izbirak, G., Vizvari, B., Arkat, J.: Complete coverage path planning for a multi-UAV response system in post-earthquake assessment. Robotics, **26**(5) (2016)
32. Okasanen, T., Visala, A.: Coverage path planning algorithms for agricultural field machines. J. Field Robot. **26**(8), 651–668 (2009)
33. Russell, C., Jung, J., Willink, G., Glasner, B.: Wind tunnel and hover performance test results for multicopter UAS vehicles. In: AHS 72^{nd} Annual Forum. West Palm Beach (2016)
34. Torres, M., Pelta, D.A., Verdegay, J.L., Torres, J.C.: Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. Expert Syst. Appl. **55**, 441–451 (2016)
35. Toth, P., Vigo, D. (eds.): Vehicle Routing - Problems, Methods, and Applications, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2014)
36. Valente, J., Sanz, D., Cerro, J.D., Barrientos, A., de Frutos, M.A.: Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields. Precision Agric. **14**, 115–132 (2013)
37. Viet, H., Dang, V., Laskar, M., Chung, T.: BA*: An online complete coverage algorithm for cleaning robots. Int. J. Appl. Intell. Neural Netw. Complex Problem Solving Technol. **39**, 217–235 (2013)
38. Zelinsky, A., Jarvis, R., Byrne, J., Yuta, S.: Planning paths of complete coverage of an unstructured environment by a mobile robot. In: Proceedings of International Conference on Advanced Roboitcs, pp. 533–538 (1993)
39. Zillies, J., Westphal, S., Thakur, D., Kumar, V., Pappas, G., Scheidt, D.: A column generation approach for optimized routing and coordination of a UAV fleet. In: 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). EPFL, Lausanne (2016)

Younghoon Choi is a Ph.D. candidate at the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology. He received his B.S. and M.S. degree in Aerospace and Mechanical Engineering from Korea Aerospace University in 2006 and 2008, respectively. His research areas are path planning and operational planning for autonomous vehicles, aerospace system design, and intelligent algorithms.

Youngjun Choi is a research faculty at the School of Guggenheim Aerospace Engineering, the Georgia Institute of Technology. Dr. Choi received Ph.D. from that institution in 2016. He was a visiting researcher at the Centre for Power Transmission and Motion Control at the University of Bath and worked at the Agency for Defense Development. His research areas are guidance, navigation, and control for autonomous vehicles, machine learning algorithms, and robotics.

Dr. Simon Briceno is a Senior Research Engineer and the Transformational Aviation Concepts Division Chief at the Aerospace Systems Design Laboratory at the Georgia Institute of Technology. Over the past 10 years he has led research in rotary and fixed wing systems design, transformative air vehicle concepts design and operations, and unmanned air-systems design. Dr. Briceno received a Ph.D. in Aerospace Engineering from Georgia Tech.

Dimitri N. Mavris is S.P. Langley Regents Professor at the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology. He is director of the Aerospace Systems Design Laboratory where over 40 research faculty and 200 graduate students conduct research in civil aviation, propulsion and power, advanced concepts, systems engineering, and defense and space.