



Unmanned Aircraft System Path Planning for Visually Inspecting Electric Transmission Towers

Hyeoncheol Baik¹ · Jorge Valenzuela¹

Received: 10 May 2018 / Accepted: 2 October 2018
© Springer Nature B.V. 2018

Abstract

To detect faults on the power transmission and distribution systems, current electric utilities perform a visual inspection by dispatching line crews and helicopters. This practice has disadvantages such as high operation costs and safety concerns. To resolve these issues, power utilities are considering the use of an unmanned aircraft system (UAS). In this paper, we formulate an optimization model to find an efficient flight path for a UAS for visually inspecting a transmission tower. The objective of the model is to maximize a function involving three performance ratios, namely, flight time, image quality, and tower coverage. The optimization model is non-linear, non-differentiable, and multi-modal. We solve the problem by using a particle swarm optimization (PSO) based-algorithm and a simulated annealing (SA) based-algorithm and compare their results. We test the model under three inspection strategies. The experimental results show that the PSO-based algorithm outperforms the SA-based algorithm. They also show that the proposed model can provide a flight path that comprises a good balance over the three performance ratios.

Keywords Camera calibration · Particle swarm optimization · Simulated annealing · Transmission tower inspection · Unmanned aircraft system · Drone

Nomenclature

Indices

i	Particle (flight path) in the PSO
j	Waypoint in the particle
t	Time step

Sets

Φ	Set of all coordinates within the flying area
Φ_S	Set of surface coordinates of the tower
Φ_N	Set of coordinates of the no-fly zone
Φ_C	Set of candidates' coordinates for a waypoint
	$\Phi_C = \Phi - \Phi_N$

Parameters

\mathbf{D}	Maximum x , y , and z of the flying area
Ω	Weight vector of the objective function ($\omega_1, \omega_2, \omega_3$)

\mathbf{e}	Unit vector of each coordinate in a search space
FS	Flying speed of the UAS (m/s)
H	Hovering time of the UAS at a waypoint (sec)
FT^{max}	Maximum flight time (sec)
FD^{max}	Maximum flight distance (m)
N^s	Total number of surface elements of the tower
$(X^{tc}, Y^{tc}, 0)$	Tower center-base coordinates (m)
G^{max}	Maximum ground sampling distance (mm)
FL	Focal length of the camera (mm)
L	Height of the image sensor (mm)
N^p	Vertical number of pixels in the image
N^{max}	Maximum number of waypoints
N	Number of waypoints
M	Number of particles
C_1, C_2	Learning factors
W	Inertial weight
T_0	Initial temperature
N_S	Number of cycles
N_T	Number of step vector adjustments
N_ϵ	Number of successive temperature reductions
N_{eval}	Maximum number of objective function evaluations
r_T	Reduction coefficient
ϵ	Tolerance for stopping iterations

✉ Jorge Valenzuela
valenjo@auburn.edu

Hyeoncheol Baik
hbaik@auburn.edu

¹ Department of Industrial and Systems Engineering, Auburn University, Auburn, AL 36849, USA

Variables

\mathbf{fp}_i	Position of flight path (particle) i
$\mathbf{wp}_{i,j}$	Coordinates of waypoint j in flight path (particle) i
\mathbf{fv}_i	Velocity of particle i
$\mathbf{wv}_{i,j}$	Velocity of waypoint j in particle i
\mathbf{R}	Rotation matrix
\mathbf{T}	Translation matrix
\mathbf{TR}	Transformation matrix
\mathbf{P}_i	Position of particle i
\mathbf{V}_i	Velocity of particle i
\mathbf{PB}_i^t	Best position of particle i at time t
\mathbf{GB}^t	Position of the global best particle at time t
\mathbf{s}	Step vector
\mathbf{x}	Decision variable vector
\mathbf{x}'	New decision variable vector
pr_T	Total performance ratio of a flight path
pr_1	Performance ratio for the total flight time
pr_2	Performance ratio for the image quality
pr_3	Performance ratio for the tower coverage
ft_i	Flight time of particle i
fd_i	Flight distance of particle i
wd_i	Flight distance from the UAS home-base to the last waypoint of particle i
S^{rc}	Real coordinates (3D) of a tower surface element
S^{cc}	Camera coordinates (3D) of a tower surface element
S^{ic}	Image coordinates (2D) of a tower surface element
(x, y, z)	Real coordinates
(u, v, w)	Camera coordinates
(x', y')	Image coordinates
n_i^s	Total surface elements covered by the images taken during flight path (particle) i
$g_{i,j}$	Ground sampling distance of the images taken at waypoint j in particle i
v^{max}	Maximum velocity of a particle
r, r_1, r_2	Random numbers in [0,1]
T	Temperature
p	Acceptance probability
m	Number of accepting a new decision variable vector

1 Introduction

Over the decades, electric utilities have relied on foot patrol and helicopters to perform a visual inspection on power transmission and distribution (T&D) systems. These practices are time-consuming, expensive and highly dangerous. Operating helicopters can cost up to \$1,500 an hour [1] and more than \$2 per kilometer [2]. According to a report from the U.S. Bureau of Labor Statistics, electrical power-line installers and repairers ranked in the top 10 lists of

civilian jobs with high fatal work injury rates in 2015 [3]. In addition, a visual inspection is the main mean of identifying defects on T&D systems [4]. For these reasons, utilities are seeking innovative technologies that can replace line crews and helicopters in power transmission and distribution (T&D) inspection. One promising alternative is the use of an unmanned aircraft system (UAS, aka drone) [5]. The federal aviation administration (FAA) defines a UAS as an aircraft maneuvered by an operator on the ground instead of a human pilot onboard [6].

UAS applications in power T&D systems have shown several benefits since early UAS researchers proposed a schematic concept and proved it with laboratory tests [7, 8]. First, a UAS can be equipped with diverse types of sensors for an inspection. The electric power research institute (EPRI) elaborates possible inspection scenarios using the various sensors with a comprehensive summary of up-to-date sensing technology [9, 10]. Supporting their proposal, EPRI demonstrates that a UAS can be used effectively to assess storm damage [11]. Second, a UAS can be used as a mobile platform to collect and relay data [9, 10, 12]. Third, a UAS can execute two different levels of inspections (macro/micro) depending on the wing type of a UAS: fixed-wing or rotary-wing. Since a fixed-wing UAS cannot hover itself, it provides a macro-level inspection which, for example, is surveying power lines in a large area [13]. On the other hand, a rotary-wing UAS can perform a micro-level inspection such as checking for mechanical failures on transmission tower components [14]. Sometimes, it is possible to use cooperative operation of both types [15]. Fourth, a UAS can improve routine inspections such as vegetation monitoring and unscheduled inspections such as weather-related damages. Lastly, a UAS allows us to implement an automatic inspection system with the aid of image processing [16, 17].

However, using a UAS is challenging because there are still major barriers blocking its full implementation. Technically, it has a limited battery capacity and may not be as capable operating in a large area when compared to a manned helicopter [7]. In addition, a stable flight control mechanism should be embedded to keep a UAS away from a strong electromagnetic field and any obstacles [18, 19]. In [20], the authors point out that little research has been done on the flying robot because it deals with more complex technical problems than the climbing robot. Regulatory issues are another factor constraining the operation scope. For example, in the United States, a UAS must fly within the visual line-of-sight (VLOS) of the remote pilot or observer during daylight hours according to a rule enacted by the FAA [21]. Other issues include privacy intrusions, liability of an accident, and hacking. We emphasize that the limited battery capacity and VLOS requirement reinforce the need

to determine the flight path of a UAS for operational efficiency.

To the best of our knowledge, no research has addressed the flight path planning optimization problem of a UAS for inspecting transmission towers, although there are some research works for inspecting other objects such as structure and terrain [22–24]. Thus, our research question is how to determine an optimal flight path considering three areas of performance: (i) flight time, (ii) image quality, and (iii) tower coverage that is the total surface area of the transmission tower captured by the camera of a UAS. Note that our model will help to build an autonomous system which is the goal of most literature research [5, 16, 17, 19].

The remainder of this paper is organized as follows. In Section 2, we describe how to mathematically model this flight path planning problem. Section 3 shows how to apply the particle swarm optimization (PSO) and simulated annealing (SA) algorithm to find the optimal flight path. Section 4 discusses the numerical experiments and results analysis. Finally, Section 5 states our conclusions.

2 The Proposed Model

To perform a micro-level inspection of a transmission tower by hovering, we use a rotary-wing UAS that navigates autonomously through waypoints input by a remote pilot. A waypoint is a position that the UAS must visit on a flight path. The optimization problem is to find the optimal flight path considering flight time, image quality, and tower coverage performance ratios. The flight time ratio is computed with respect to the maximum flight time FT^{max} obtained from the hovering time at N^{max} waypoints plus the corresponding cruising time. It is assumed that the UAS flies at a constant speed, FS m/s, hovers at a waypoint for a constant time of H seconds and is not affected by wind. The battery charge should allow the UAS to fly for FT^{max} seconds. To compute the image quality ratio, we assume that one standard digital camera is mounted on the UAS and cannot be tilted. By using a camera gimbal, the UAS keeps the camera upright with respect to the horizon regardless of its motion. We use a prime lens on the camera to take images at a constant resolution. Thus, the distance from the lens to the image sensor is FL millimeters. For the image size, we choose the 4:3 aspect ratio. The vertical number of pixels is given by N^p . To comply with FAA rules, the UAS must weigh less than 25 kilograms including the camera [21]. The tower coverage ratio is calculated by representing the tower surface by a discrete set of N^s triangular and rectangular elements. These surface elements are denoted by the set Φ_s . Thus, the tower coverage is obtained by totaling the different surface elements identified in each of the images taken by the UAS during its flight.

The UAS is restricted to flying within the area given by the set Φ which is a cuboid delimited by \mathbf{D} . This area should meet the VLOS requirement and consider the no tree zone within a transmission line right-of-way. We designate the position $(X^{tc}, Y^{tc}, 0)$ to be the center-base of the tower and the position $(0, 0, 0)$ to be the home-base of the UAS. To consider the electromagnetic field around the tower and the tower itself, we define the no-fly zone Φ_N . It is assumed that the UAS takes off from the home-base and returns to the same location. At each waypoint, the UAS hovers and takes an image. We assume that the UAS flies along a straight line connected by two waypoints. If the UAS encounters the no-fly zone, it detours around the boundaries of the no-fly zone. We also assume that there is no other obstacle within the flying area.

2.1 Tower Model

The tower surface is represented by a discrete set of N^s triangular and rectangular elements. As an example, we choose the 132 kV transmission tower that is used to transmit extra-high voltage over a long distance. Figure 1 shows the tower dimensions obtained from [25] and Fig. 2 depicts the tower model and its surface elements.

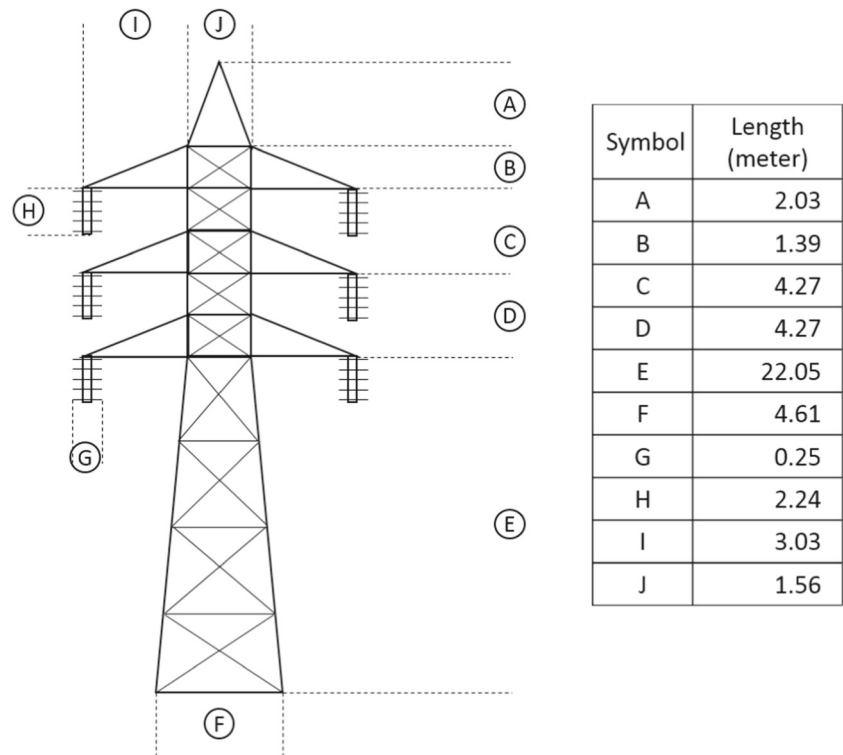
2.2 Flight Time Ratio

We define the flight time ratio of path i as the difference between the maximum flight time (FT^{max}) and the flight time of path i relative to FT^{max} (see Eq. 8). The flight time consists of hovering and cruising times. The hovering time is calculated by multiplying H by N . The cruising time is obtained by dividing the Euclidean distance between waypoints by the flight speed (FS). Figure 3 illustrates the process of determining FT^{max} when the maximum number of waypoints N^{max} is 3. The process starts by dividing the flying area into N^{max} cuboids. The first waypoint is assigned to be the vertex opposite from the UAS home-base. The second waypoint is the vertex that has the longest distance from the first waypoint. This process continues until the remaining waypoints are determined. The maximum flight distance (FD^{max}) is the sum of the distances between two consecutive waypoints. The cruising time is obtained by dividing FD^{max} by the flight speed FS . The hovering time is computed by multiplying H by N^{max} . The FT^{max} is the sum of the cruising and hovering times.

2.3 Image Quality Ratio

The most common way to measure image quality is to use pixel resolution, which is a camera dependent specification (e.g. 12 megapixels). However, pixel resolution does not provide a good way to measure the image quality of a

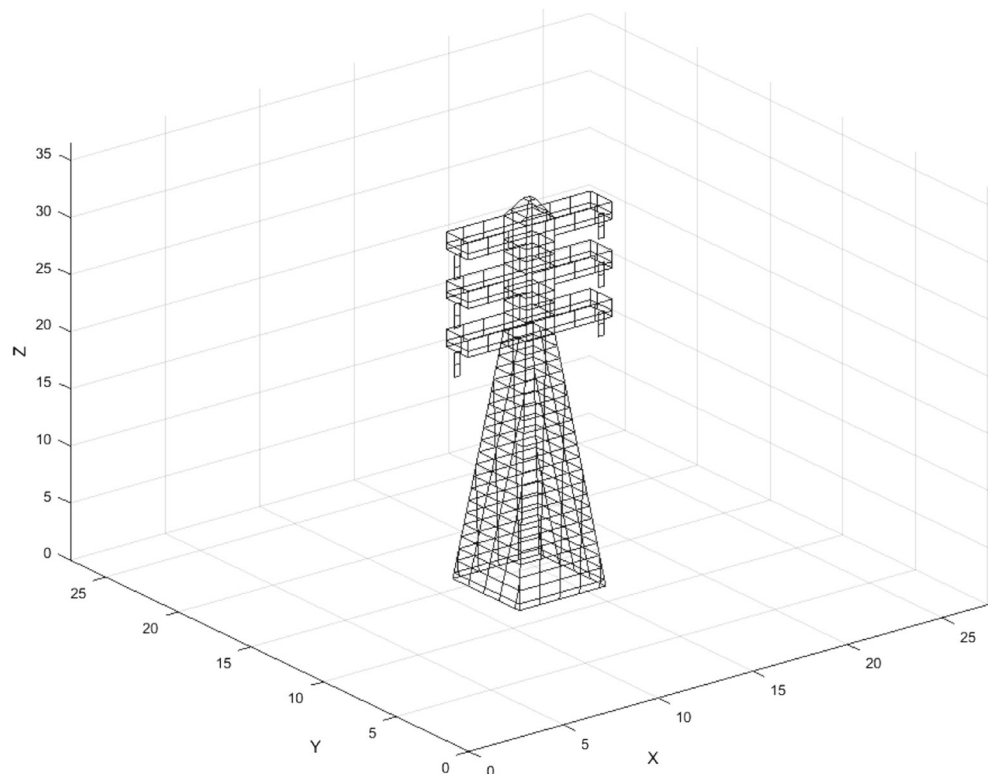
Fig. 1 132 kV transmission tower dimension



feature in the image. Instead, we use spatial resolution that refers to the size of the smallest possible object discernible in the image. To measure the spatial resolution, we use the ground sampling distance (GSD) that is the distance

between two pixels' centers measured on the ground as shown in Fig. 4 [26]. The unit of measure is millimeter per pixel. A lower GSD provides a higher image quality. In our situation, the ground corresponds to the vertical plane

Fig. 2 Transmission tower model



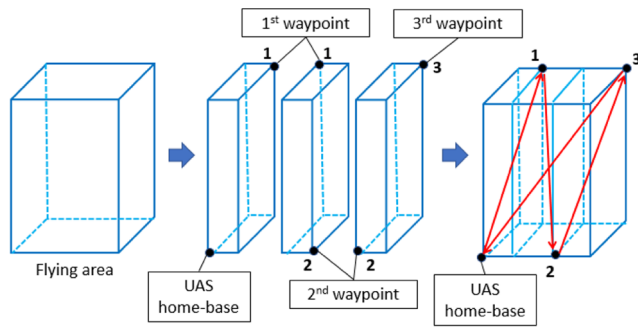


Fig. 3 Maximum flight distance for $N^{max} = 3$

that is parallel to the UAS position and contains the tower center-base (see Fig. 5). As the UAS moves closer to the tower, the GSD decreases and thus, the quality of the image will increase. The maximum GSD (G^{max}) is the GSD of an image taken at the farthestmost waypoint from the tower center-base constrained to the flying area. We define the image quality ratio of path i as the difference between G^{max} and the average image GSD of path i relative to G^{max} (see Eq. 9).

2.4 Tower Coverage Ratio

We define the tower coverage ratio of path i as the number of surface elements (n_i^s) covered by the images taken during path i relative to the total number of surface elements of the tower (N^s). To compute n_i^s , we use three different coordinate systems: (i) real coordinates, (ii) camera coordinates, and (iii) image coordinates. The origin of the real coordinates is the home-base of the UAS. The

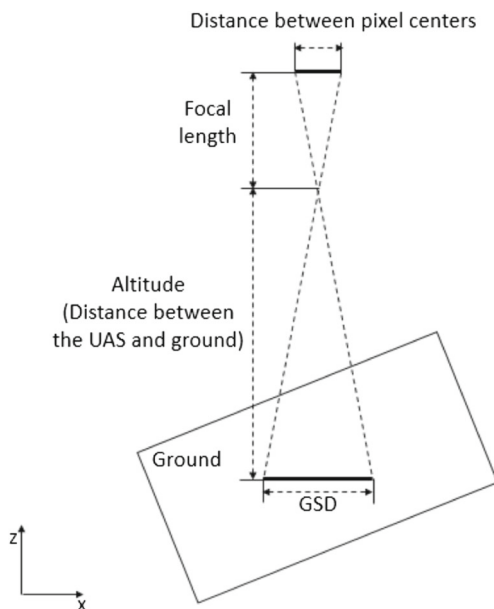


Fig. 4 Ground Sampling Distance (GSD)

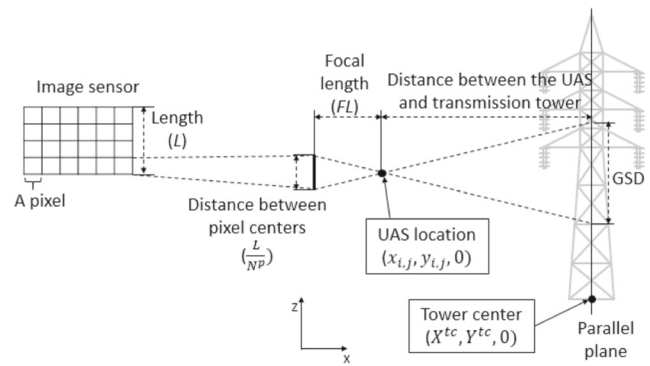


Fig. 5 GSD in inspecting a transmission tower

origin of the camera coordinates is the center of the camera lens. The origin of the image coordinate is the center of the image sensor (see Fig. 6).

2.4.1 Transforming Real Coordinates to Camera Coordinates

Suppose that S^{rc} is the point in the real coordinate system and that S^{cc} is the corresponding point in the camera coordinate system (see Fig. 7). Thus, S^{cc} is given by Eq. 1 where \mathbf{TR} is the rigid transformation matrix obtained by multiplying the rotation matrix (\mathbf{R}) and the translation matrix (\mathbf{T}) [27].

$$S^{cc} = \mathbf{TR} \cdot S^{rc} = \mathbf{R} \cdot \mathbf{T} \cdot S^{rc} \quad (1)$$

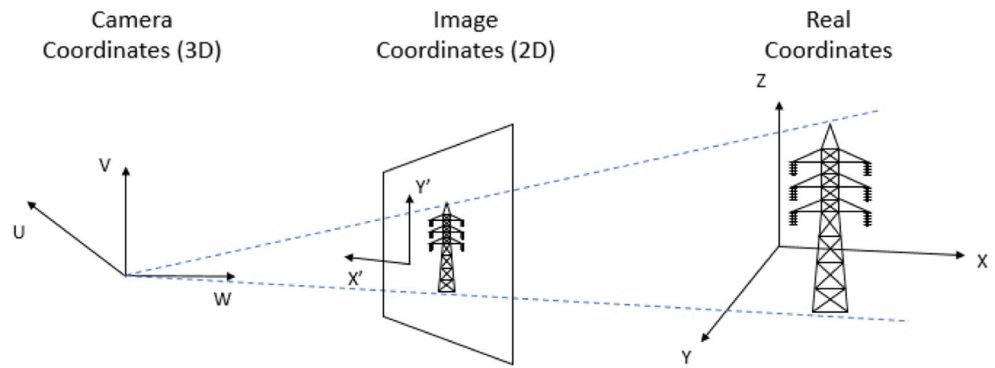
Assuming that the center of the camera lens is at the same location as the waypoint $(x_{i,j}, y_{i,j}, z_{i,j})$, the coordinate transformation is given by Eq. 2.

$$\begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_{i,j} \\ 0 & 1 & 0 & -y_{i,j} \\ 0 & 0 & 1 & -z_{i,j} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2)$$

To compute \mathbf{R} , we adopt the same technique described in [28]. \mathbf{R} is an orthogonal matrix that has the following properties: (i) the squares of the elements in any row or column sum to one, and (ii) the rows are the projection of a unit vector along the camera coordinate axes onto the real coordinates.

2.4.2 Transforming Camera Coordinates to Image Coordinates

To map surface elements from camera coordinates to 2D image coordinates, we use the perspective projection method (also known as the pinhole camera model) [28]. In Fig. 8, the point S^{cc} in camera coordinates is projected by

Fig. 6 Camera, image, and real coordinate systems

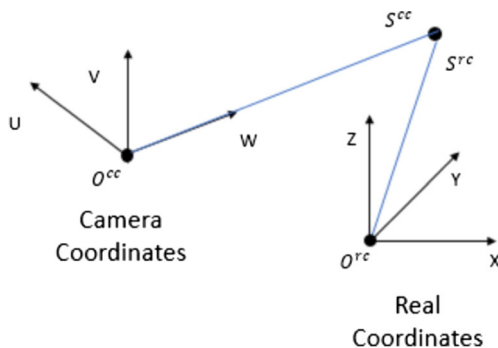
Eqs. 3 and 4 into the point S^{ic} in image coordinates. Notice that only surface elements in image coordinates located within the image size are retained.

$$x' = FL \frac{u}{w} \quad (3)$$

$$y' = FL \frac{v}{w} \quad (4)$$

2.4.3 Hidden Surface Elimination Problem

The hidden surface elimination problem deals with finding the surface elements that are visible in the image [29]. Surface elements in image coordinates within the image size may not be visible because they may be behind other surface elements (see Fig. 9). To solve this problem, we use the z-buffer method [30]. When multiple surface elements have the same image coordinates, the surface element with the shortest distance, measured in camera coordinates, to the center of the camera lens is retained. After gathering all visible surface elements in the images from all waypoints, duplicated surface elements are removed. The resulting number of visible unique surface elements for path i is denoted by n_i^s .

**Fig. 7** Transforming real coordinates to camera coordinates

2.5 Optimization Model

The objective of the optimization model is to maximize a total performance ratio (pr_T) which is the weighted average of flight time ratio, image quality ratio and tower coverage ratio (see Eq. 7). Equations 8 to 18 calculate the ratios. Equation 19 ensures that the UAS flies within the flying area. The decision variable is a flight path i having N waypoints and expressed by Eqs. 5 and 6.

$$\mathbf{fp}_i = (\mathbf{wp}_{i,1}, \mathbf{wp}_{i,2}, \dots, \mathbf{wp}_{i,N}) \quad (5)$$

where

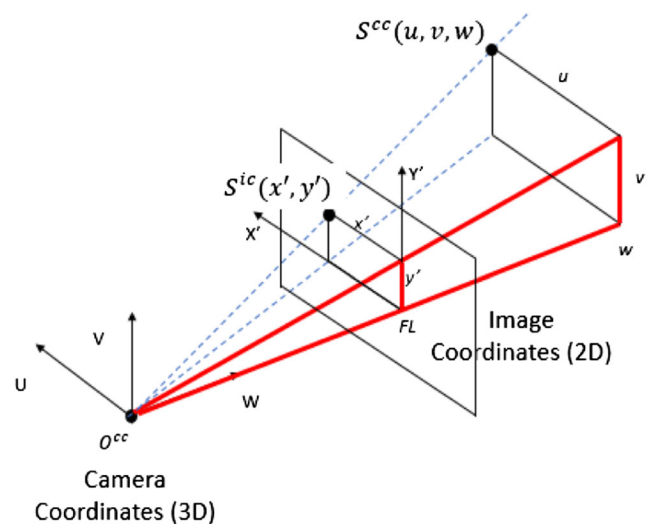
$$\mathbf{wp}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \quad (6)$$

The objective function and constraints are given by Eqs. 7–19.

$$\text{Max } pr_T = \omega_1 pr_1 + \omega_2 pr_2 + \omega_3 pr_3 \quad (7)$$

Subject to

$$pr_1 = 1 - \frac{ft_i}{FT^{max}} \quad (8)$$

**Fig. 8** Transforming camera to image coordinates

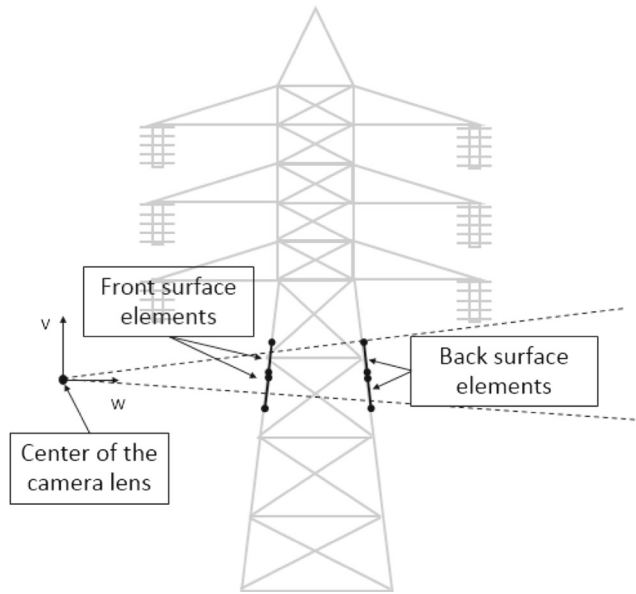


Fig. 9 Front and back surface elements

$$pr_2 = 1 - \left(\frac{1}{N} \sum_{j=1}^N g_{i,j} \right) \left(\frac{1}{G^{max}} \right) \quad (9)$$

$$pr_3 = \frac{n_i^s}{N^s} \quad (10)$$

$$FT^{max} = \frac{FD^{max}}{FS} + H \times N^{max} \quad (11)$$

$$fd_i = \frac{fd_i}{FS} + H \times N \quad (12)$$

$$fd_i = wd_i + \sqrt{x_{i,N}^2 + y_{i,N}^2 + z_{i,N}^2} \quad (13)$$

$$wd_i = \sum_{j=0}^{N-1} \sqrt{\Delta x_{i,j}^2 + \Delta y_{i,j}^2 + \Delta z_{i,j}^2} \quad (14)$$

$$\Delta x_{i,j} = x_{i,j+1} - x_{i,j} \quad (15)$$

$$\Delta y_{i,j} = y_{i,j+1} - y_{i,j} \quad (16)$$

$$\Delta z_{i,j} = z_{i,j+1} - z_{i,j} \quad (17)$$

$$g_{i,j} = \frac{L \times \sqrt{(x_{i,j} - X^{tc})^2 + (y_{i,j} - Y^{tc})^2}}{N^p \times FL}, \forall j \quad (18)$$

$$fp_i \in \Phi_C \quad (19)$$

```

1: For  $i = 1$  to  $M$  do
2:   Initialize  $P_i^1$  and  $V_i^1$ 
3:   Compute the fitness value of particle  $i$ 
4:   Determine  $PB_i^1$ 
5: End for
6: Determine  $GB^1$ 
7: Increase the time step  $t$  by 1
8: While (stop criterion is not met) do
9:   Update  $P_i^t$  and  $V_i^t$  using Eqs. (22) and (23)
10:  For  $i = 1$  to  $M$  do
11:    Compute the fitness value of particle  $i$ 
12:    Update  $PB_i^t$ 
13:  End for
14:  Update  $GB^t$ 
15:  Increase the time step  $t$  by 1
16: End while

```

Fig. 10 Pseudocode for the PSO

Notice that the objective function of this optimization problem is non-linear, non-differentiable and multimodal. The variables pr_1 and pr_2 in the objective function are calculated using Eqs. 13 and 18 which are non-linear equations. The variable pr_3 is calculated using Eq. 10 in which the variable n_i^s is an integer. The problem is multimodal because multiple local optima coexist.

3 Solution Algorithms

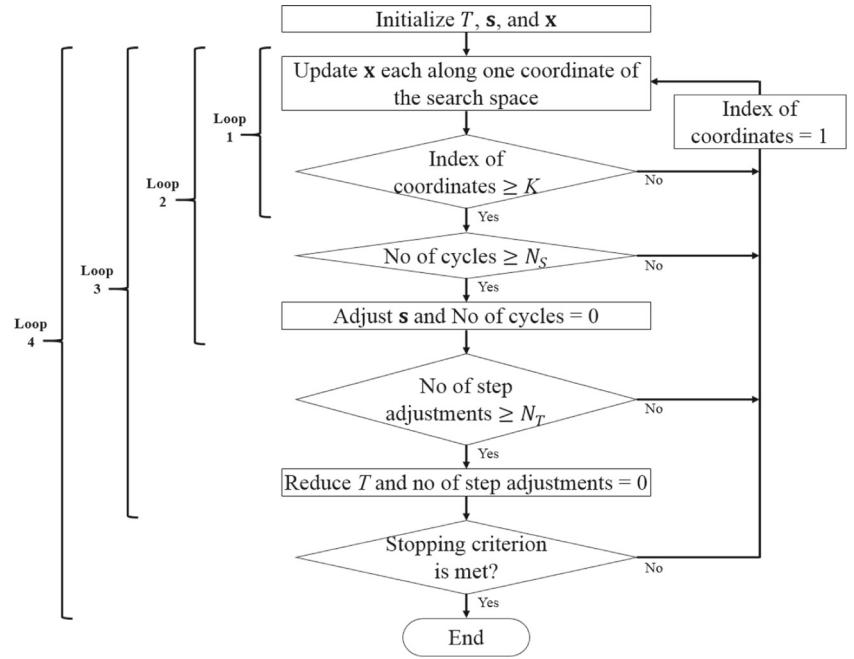
The optimization problem cannot be solved by using a gradient-based optimization method. Thus, we use two common gradient-free algorithms, namely, particle swarm optimization (PSO) based-algorithm and simulated annealing (SA) based-algorithm.

3.1 PSO-Based Solution Algorithm

The PSO is a global and evolutionary programming algorithm developed from simulating social behaviors such as bird flocking and fish schooling [31]. Specially, this algorithm is easy to implement by adjusting very few parameters and can obtain the optimal or near-optimal solutions within relatively short time [32].

In the standard PSO, a candidate solution, a set of candidate solutions, and the objective function are called particle, swarm, and fitness function, respectively. The algorithm starts by randomly initializing the position and velocity of the particles in the search space. Assuming a K -

Fig. 11 Flowchart for the SA



dimensional search space and M particles, the position and velocity are described by Eqs. 20 and 21.

$$\mathbf{P}_i = (p_{i,1}, p_{i,2}, \dots, p_{i,K}), \forall i \quad (20)$$

$$\mathbf{V}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,K}), \forall i \quad (21)$$

Particles move to the next position by adjusting their velocity. The velocity is determined by considering three

factors: current position, the best previous position of the particle (\mathbf{PB}_i^t) at the current time step, and the best previous position of the swarm (\mathbf{GB}^t) at the current time step. Equations 22 and 23 show updating \mathbf{P}_i and \mathbf{V}_i at each time step t .

$$\mathbf{P}_i^{t+1} = \mathbf{P}_i^t + \mathbf{V}_i^{t+1} \quad (22)$$

$$\mathbf{V}_i^{t+1} = W\mathbf{V}_i^t + C_1 r_1 (\mathbf{PB}_i^t - \mathbf{P}_i^t) + C_2 r_2 (\mathbf{GB}^t - \mathbf{P}_i^t) \quad (23)$$

where W is the inertial weight, C_1 and C_2 the learning factors, and r_1 and r_2 random numbers between 0 and 1. This updating continues until the stopping criterion is

```

1: For  $n_{cycles} = 0$  to  $N_S$  do
2:   If ( $n_{cycles} < N_S$ ) then
3:     For  $k = 1$  to  $K$  do
4:       Update  $\mathbf{x}$  using Eq. (24)
5:       While ( $\mathbf{x}'$  violates the boundary constraints) do
6:         Update  $\mathbf{x}$  using Eq. (24)
7:       End while
8:       If ( $f(\mathbf{x}') < f(\mathbf{x})$ ) then
9:         Accept  $\mathbf{x}'$ ,  $\mathbf{x}^* \leftarrow \mathbf{x}'$ ,  $m_k \leftarrow m_k + 1$ 
10:      Else
11:        Compute  $p$  using Eq. (25)
12:        If ( $\text{rand}() \leq p$ ) then
13:          Accept  $\mathbf{x}'$ ,  $m_k \leftarrow m_k + 1$ 
14:        End if
15:      End if
16:    End for
17:  Else
18:    For  $k = 1$  to  $K$  do
19:      If ( $m_k > 0.6 \times N_S$ ) then
20:        Adjust  $\mathbf{s}$  using Eq. (26)
21:      Else if ( $m_k < 0.4 \times N_S$ ) then
22:        Adjust  $\mathbf{s}$  using Eq. (27)
23:      End if
24:    End for
25:  End if
26: End for
  
```

Fig. 12 Pseudocode for the SA

Table 1 System parameters

UAS specifications	
Flying speed (FS)	5 m/s
Hovering time to take an image (H)	3 s
Maximum flight time (FT^{max})	491 s
Maximum flight distance (FD^{max})	1,856 m
Camera specifications	
Height of the image sensor (L)	4.6 mm
Focal length (FL)	3.6 mm
Vertical number of pixels in the image (N^p)	3,000
Transmission tower	
Total number of surface elements (N^s)	464
Tower center-base coordinates ($X^{tc}, Y^{tc}, 0$)	(13.72, 13.72, 0)
Flying area	
Maximum x , y , and z of the flying area (\mathbf{D})	(27.43, 27.43, 36.58)
Maximum number of waypoints (N^{max})	40
Maximum ground sampling distance (G^{max})	8.2 mm

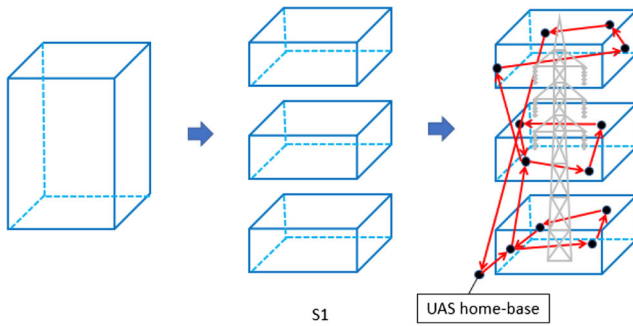


Fig. 13 Strategy S1 ($N = 12$)

satisfied. The stopping criterion is either the number of iterations or no further improvement in the fitness value between consecutive iterations. The pseudocode of the PSO algorithm is shown in Fig. 10.

In terms of the communication of the swarm, two models have been proposed: local topology and global topology. While the former allows a particle to exchange information with only proximate neighbors, in the latter all particles share information together. It is not possible to say which model has superior performance. A global topology shows a faster convergence rate than a local topology that escapes from a local optimum. However, a local topology finds a better fitness value than a global topology after some number of iterations for particularly multimodal problems [33].

In our problem, a particle is a candidate flight path i . For the communication of the swarm, we use a global topology to avoid obtaining a local optimum. It is possible that particles fly beyond the flying area Φ_C . To fix this issue, we limit the velocity to v^{max} and use a boundary condition that places the particle back into the area Φ_C . We use the absorbing approach described in [34] to relocate the particle at the boundary of the Φ_C with zero velocity.

3.2 SA-Based Solution Algorithm

The SA is a global programming algorithm derived from an analogy with annealing metals and firstly introduced

Fig. 14 Strategy S2 ($N = 12$)

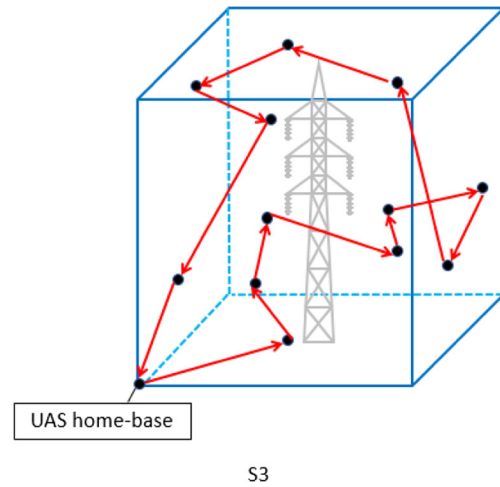
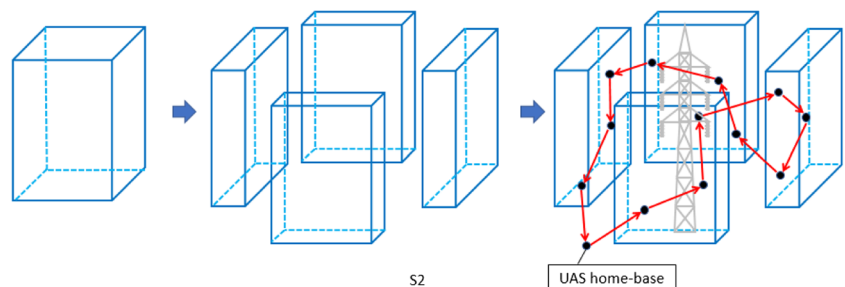


Fig. 15 Strategy S3 ($N = 12$)

to solve combinatorial optimization problem [35]. In thermodynamics, molecules are free to move at high temperatures, while they are restricted to move at low temperatures. If the temperature is lowered slowly in the cooling process, the molecules form a rigid structure that has the lowest energy state. Similarly, in the minimization problem, cooling gradually enables to find the solution having the lowest objective function value.

Our benchmark is the SA algorithm presented in [36] that is applicable for multimodal functions of continuous variables. As shown in Fig. 11, the algorithm starts by initializing the temperature (T), step vector (\mathbf{s}), and decision variable vector (\mathbf{x}). Then, four consecutive loops are executed until the stopping criterion is met.

Loop 1: Random move

Assuming a search space has K coordinates, \mathbf{x} is updated each along one coordinate according to Eq. 24.

$$\mathbf{x}' = \mathbf{x} + r \cdot \mathbf{s} \cdot \mathbf{e} \quad (24)$$

where r is the random number between 0 and 1, and \mathbf{e} is the unit vector of each coordinate. When \mathbf{x}' violates the boundary constraints, another \mathbf{x}' will be generated until it

Table 2 PSO parameter tuning results

W	C_1	C_2	pr_T	pr_1	pr_2	pr_3
0.3	1	1	0.83	0.81	0.69	0.98
	1	2	0.83	0.82	0.70	0.98
	2	1	0.83	0.82	0.70	0.99
	2	2	0.84	0.82	0.72	0.98
0.6	1	1	0.84	0.81	0.71	0.99
	1	2	0.84	0.82	0.72	0.98
	2	1	0.84	0.82	0.73	0.99
	2	2	0.83	0.82	0.70	0.98
0.9	1	1	0.83	0.81	0.70	0.98
	1	2	0.81	0.81	0.65	0.98
	2	1	0.82	0.81	0.67	0.98
	2	2	0.80	0.80	0.62	0.97
1.2	1	1	0.80	0.80	0.61	0.97
	1	2	0.79	0.80	0.59	0.98
	2	1	0.80	0.80	0.63	0.97
	2	2	0.79	0.80	0.59	0.97
1.5	1	1	0.79	0.80	0.60	0.96
	1	2	0.78	0.79	0.57	0.97
	2	1	0.79	0.80	0.60	0.97
	2	2	0.78	0.79	0.58	0.98

satisfies the constraints. If the new objective function value ($f(\mathbf{x}')$) is less than current one ($f(\mathbf{x}^*)$), we accept the \mathbf{x}' and record it as the optimum (\mathbf{x}^*). If not, we only accept or reject \mathbf{x}' with acceptance probability (p) computed by Eq. 25.

$$p = \exp\left(-\frac{f(\mathbf{x}') - f(\mathbf{x})}{T}\right) \quad (25)$$

This loop is repeated K times which is the number of coordinates.

Loop 2: Cycle

Table 3 SA parameter tuning results

T_0	r_T	pr_T	pr_1	pr_2	pr_3
0.01	0.50	0.74	0.79	0.48	0.94
	0.70	0.73	0.79	0.47	0.93
	0.85	0.73	0.79	0.48	0.93
0.02	0.50	0.73	0.77	0.47	0.95
	0.70	0.74	0.78	0.47	0.96
	0.85	0.73	0.78	0.46	0.94
0.03	0.50	0.74	0.78	0.48	0.97
	0.70	0.73	0.78	0.46	0.95
	0.85	0.72	0.77	0.44	0.95

This loop includes K random moves and one step vector adjustment which is done each along one coordinate according to Eq. 26 or 27.

$$\mathbf{s} = \mathbf{s} \times \left(1 + c \times \frac{m_k/N_S - 0.6}{0.4}\right) \quad \text{For } m_k > 0.6 \times N_S \quad (26)$$

$$\mathbf{s} = \frac{\mathbf{s}}{\left(1 + c \times \frac{0.4 - m_k/N_S}{0.4}\right)} \quad \text{For } m_k < 0.4 \times N_S \quad (27)$$

where m_k is the number of accepting \mathbf{x}' in each coordinate and N_S is the number of cycles. The pseudocode of the random move and cycle is illustrated in Fig. 12.

Loop 3: Step vector adjustment

This loop consists of K random moves and N_S cycles. It is repeated N_T times which is the number of step vector adjustments.

Loop 4: Temperature reduction

This loop consists of K random moves, N_S cycles, N_T step vector adjustments, and one temperature reduction which is done by Eq. 28.

$$T = r_T \times T \quad (28)$$

where r_T is the reduction coefficient.

The algorithm is terminated when either two conditions are satisfied: (i) all changes of the last accepted \mathbf{x}' over N_ε successive temperature reductions are less than ε , and the difference between the last accepted $f(\mathbf{x}')$ and current $f(\mathbf{x}^*)$ is less than ε , or (ii) the number of objective function evaluations is greater than N_{eval} .

In our problem, the decision variable vector is a flight path (\mathbf{fp}) determined in the search space having N coordinates. To apply the SA algorithm, we transform our problem to a minimization problem by changing the objective function (i.e. $\text{Min } 1 - pr_T$).

4 Numerical Experiments

All models for the numerical experiments are coded in MATLAB. Each experiment is run ten times on a computer with an Intel® Core™ i3-3110M (3M Cache, 2.40 GHz) processor with 8 GB of RAM memory. Table 1 shows the parameters of the UAS, camera, transmission tower, and flying area.

4.1 Inspection Strategies

We develop three inspection strategies (S1, S2, S3) for scanning the transmission tower shown in Fig. 2.

Table 4 Performance results over 3 strategies with different algorithms ($N = 12$)

	pr_T				pr_1		pr_2		pr_3		Time (min)
	Best	Avg.	Worst	CV(%)	Avg.	CV(%)	Avg.	CV(%)	Avg.	CV(%)	
S1-PSO	0.85	0.84	0.83	0.66%	0.85	0.47%	0.70	2.67%	0.98	0.63%	4.61
S2-PSO	0.77	0.77	0.75	0.68%	0.83	1.06%	0.48	2.30%	0.98	0.49%	15.87
S3-PSO	0.82	0.81	0.81	0.44%	0.83	1.39%	0.62	1.75%	0.98	0.50%	48.33
S1-SA	0.76	0.73	0.70	2.44%	0.84	0.85%	0.43	5.64%	0.92	4.96%	21.49
S2-SA	0.73	0.69	0.62	4.64%	0.84	1.03%	0.44	9.59%	0.79	14.19%	55.26
S3-SA	0.79	0.78	0.77	0.97%	0.83	0.82%	0.54	2.35%	0.96	1.39%	268.82

Strategy S1

The flying area is divided into horizontal cuboids. The UAS starts to scan from the bottom cuboid and flies through waypoints. Then, the UAS continues to scan in the cuboid above until it reaches the top cuboid (see Fig. 13).

Strategy S2

The flying area is divided into vertical cuboids. The UAS starts to scan from the home-base cuboid and continues to scan in the neighboring cuboid until it reaches the first cuboid (see Fig. 14).

Table 5 Performance results for different numbers of waypoints

	N	pr_T				pr_1		pr_2		pr_3	
		Best	Avg.	Worst	CV(%)	Avg.	CV(%)	Avg.	CV(%)	Avg.	CV(%)
S1-PSO	8	0.83	0.81	0.80	0.70%	0.88	0.65%	0.59	2.15%	0.97	1.44%
	12	0.85	0.84	0.83	0.66%	0.85	0.47%	0.70	2.67%	0.98	0.63%
	16	0.85	0.84	0.84	0.46%	0.82	0.63%	0.73	1.38%	0.99	0.44%
	20	0.84	0.84	0.83	0.35%	0.79	0.53%	0.74	0.93%	0.98	0.55%
	24	0.84	0.84	0.83	0.50%	0.76	0.88%	0.76	1.18%	0.99	0.27%
	28	0.83	0.83	0.82	0.18%	0.72	0.61%	0.77	0.63%	0.98	0.41%
	32	0.82	0.82	0.81	0.48%	0.69	0.52%	0.79	1.20%	0.98	0.52%
S2-PSO	8	0.76	0.73	0.72	1.70%	0.88	1.11%	0.43	8.86%	0.90	3.78%
	12	0.77	0.77	0.75	0.68%	0.83	1.06%	0.48	2.30%	0.98	0.49%
	16	0.78	0.77	0.75	1.05%	0.79	0.85%	0.53	4.30%	0.99	0.31%
	20	0.76	0.75	0.75	0.35%	0.74	0.96%	0.54	1.83%	0.99	0.19%
	24	0.76	0.75	0.73	1.06%	0.69	2.06%	0.56	3.07%	0.99	0.18%
	28	0.75	0.74	0.72	1.03%	0.66	2.94%	0.56	2.11%	0.99	0.19%
	32	0.73	0.73	0.72	0.74%	0.62	2.02%	0.58	1.82%	0.99	0.26%
S1-SA	8	0.74	0.72	0.68	2.29%	0.84	0.85%	0.43	5.64%	0.92	4.96%
	12	0.76	0.73	0.70	2.44%	0.84	0.85%	0.43	5.64%	0.92	4.96%
	16	0.76	0.74	0.71	1.90%	0.78	0.95%	0.48	7.47%	0.97	1.30%
	20	0.74	0.73	0.72	1.04%	0.73	1.16%	0.49	6.96%	0.96	2.04%
	24	0.72	0.71	0.68	1.78%	0.68	2.31%	0.47	8.61%	0.96	1.93%
	28	0.71	0.70	0.69	1.32%	0.64	2.74%	0.48	5.00%	0.97	3.07%
	32	0.70	0.69	0.67	1.25%	0.58	2.16%	0.50	4.94%	0.98	1.26%
S2-SA	8	0.68	0.66	0.63	2.18%	0.90	0.49%	0.44	13.27%	0.64	15.36%
	12	0.73	0.69	0.62	4.64%	0.84	1.03%	0.44	9.59%	0.79	14.19%
	16	0.72	0.69	0.66	2.87%	0.78	1.38%	0.87	8.15%	0.43	7.00%
	20	0.73	0.70	0.68	2.64%	0.74	1.81%	0.46	8.45%	0.90	7.21%
	24	0.72	0.68	0.65	3.42%	0.69	2.41%	0.47	7.41%	0.90	7.00%
	28	0.70	0.67	0.65	2.81%	0.63	2.43%	0.45	4.84%	0.93	6.68%
	32	0.67	0.66	0.64	1.48%	0.58	3.84%	0.46	6.40%	0.93	5.46%

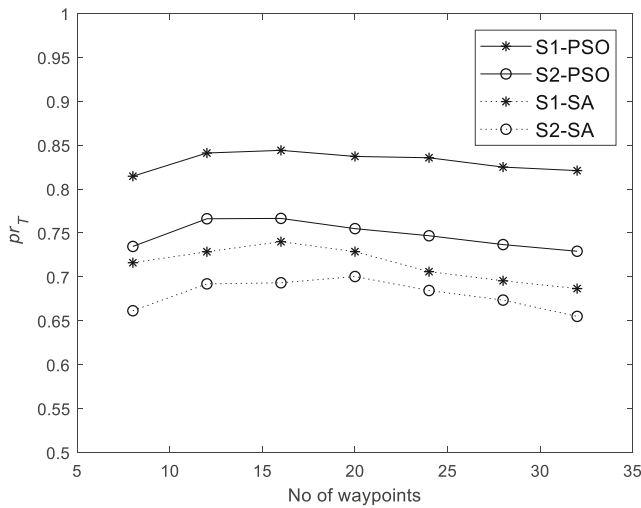


Fig. 16 Average total performance ratios for different numbers of waypoints

Strategy S3

The flying area is not divided (see Fig. 15).

4.2 Parameter Tuning and Stopping Criterion

In order to maximize the performance of the algorithms, it is necessary to tune the parameters. For the PSO-based algorithm, we run the inspection-strategy S1 with 16 waypoints for 20 different combinations of the parameters. The results are given in Table 2. The best performance (highest pr_T) is obtained with $W = 0.6$, $C_1 = 2$, and $C_2 = 1$. For the SA-based algorithm, we run the inspection-strategy S1 with 16 waypoints for 9 different combinations of the initial temperature T_0 and reduction coefficient r_T parameters, keeping other parameters ($N_S = 20$, $N_T = 20$, $N_\varepsilon = 4$, and $c = 2$) constant. The results are given in Table 3. The best performance (highest pr_T) is obtained with $T_0 = 0.03$ and $r_T = 0.5$. In terms of the stopping criterion, we use 50 number of iterations for the PSO-based algorithm, and $\varepsilon = 0.001$ and $N_{eval} = 10,000$ for the SA-based algorithm.

4.3 Experiment Results

We compare the performance ratios under the previously defined three inspection strategies for the PSO-based and SA-based algorithms. For all experiments, we obtain the best/average/worst performance ratios, coefficient of variation (CV), and the average computational time over ten replicates. We measure the computational time by using the “tic” and “toc” MATLAB functions.

4.3.1 Results Under Equal Weight Values

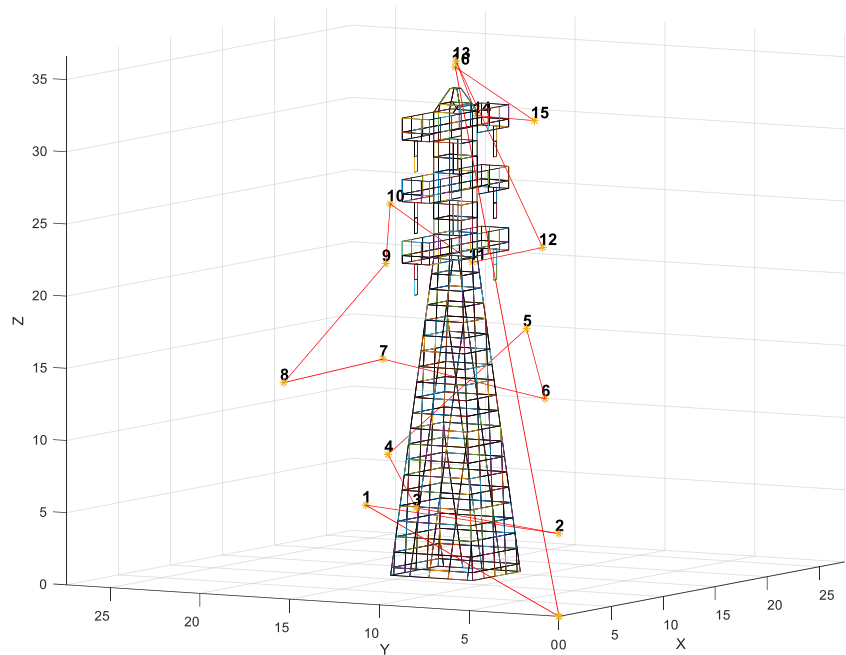
This section provides the results when we use equal weight values of the objective function ($\omega_1 = \omega_2 = \omega_3 = 1/3$). Using the same number of waypoints ($N = 12$), Table 4 shows that the strategies S1, S2, and S3 with the PSO-based algorithm surpass those with the SA-based algorithm in terms of the average total performance ratio. The image quality ratio is the main factor that causes this difference on the average total performance ratios. We find that the S1-PSO is the most efficient flight path because it produces the highest average total performance ratio and shortest computational time.

We test the strategies S1 and S2 by changing the number of waypoints N . Table 5 gives detailed results and Fig. 16 shows the average total performance ratio for different numbers of waypoints. As shown in this plot, the average total performance ratio improves as N increases, but decreases when $N > 16$ for the S1-PSO, S2-PSO and S1-SA, and $N > 28$ for the S2-SA. The reason is that a larger number of waypoints requires a longer hovering time, which in turn, deteriorates the flight time ratio. We also observe that a larger number of waypoints increases the image quality ratio when we use the PSO-based algorithm.

We study the effect of the swarm size on the performance ratios of the S1-PSO with 16 waypoints. In Table 6, we show the results for different swarm sizes. The results indicate that increasing the number of particles (particularly greater than 20 particles) does not affect significantly the performance ratios. However, having more particles requires a longer computational time. This observation supports the argument

Table 6 Performance results for different swarm sizes (S1-PSO with $N = 16$)

Swarm size	pr_T				pr_1		pr_2		pr_3		Time (min)
	Best	Avg.	Worst	CV(%)	Avg.	CV(%)	Avg.	CV(%)	Avg.	CV(%)	
10	0.842	0.836	0.827	0.59%	0.817	0.33%	0.704	1.55%	0.986	0.41%	1.81
20	0.851	0.844	0.840	0.46%	0.819	0.63%	0.726	1.38%	0.987	0.44%	3.43
30	0.850	0.845	0.838	0.52%	0.819	0.71%	0.734	1.59%	0.981	0.82%	5.35
40	0.854	0.845	0.836	0.66%	0.820	0.40%	0.732	2.36%	0.983	0.93%	7.01
50	0.851	0.848	0.843	0.33%	0.820	0.46%	0.737	0.78%	0.986	0.45%	8.54

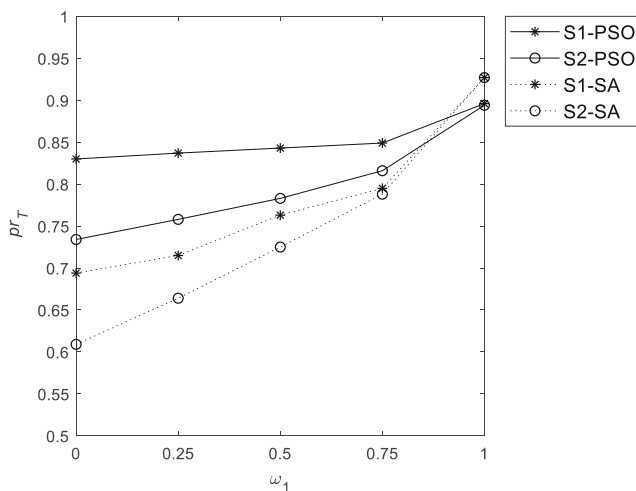
Fig. 17 Flight path for the best solution

given in [33] that states that a swarm size between 20 to 100 particles yields substantially no superior results.

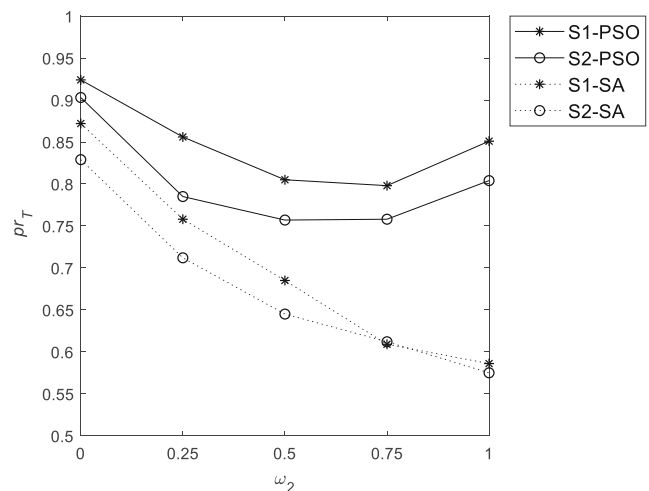
From Tables 5 and 6 we create Fig. 17 to illustrate the waypoints of the flight path for the best solution from S1-PSO with 16 waypoints obtained using a swarm size of 50. Notice that the best solution of the PSO-based algorithm (S1-PSO) improves the average total performance ratio of the best solution of the SA-based algorithm (S3-SA) by 9%.

4.3.2 Results Under Different Weight Values

We examine the average total performance ratios under different weights of the objective function by changing

**Fig. 18** Average total performance ratios for different weights of the fitness function by changing ω_1

one weight while keeping the other two weights at the same value. For example, when we change the first weight, the weight vector Ω will be $[0 \ 1/2 \ 1/2]$, $[1/4 \ 3/8 \ 3/8]$, $[1/2 \ 1/4 \ 1/4]$, $[3/4 \ 1/8 \ 1/8]$, and $[1 \ 0 \ 0]$. We solve the model for the S1-PSO, S2-PSO, S1-SA, and S2-SA with 12 waypoints. Figures 18, 19, and 20 illustrate the average total performance ratios as the ω_1 , ω_2 , and ω_3 are changed respectively. Overall, the PSO-based algorithm produces higher average total performance ratios than the SA-based algorithm except when $\Omega = [1 \ 0 \ 0]$. Also, the strategy 1 produces higher average total performance ratios than the strategy 2 except when $\Omega = [0 \ 1 \ 0]$ and $\Omega = [0 \ 1 \ 1]$ and we use the SA-based algorithm. The average total performance ratios are generally increased as we increase ω_1 and ω_3 ,

**Fig. 19** Average total performance ratios for different weights of the fitness function by changing ω_2

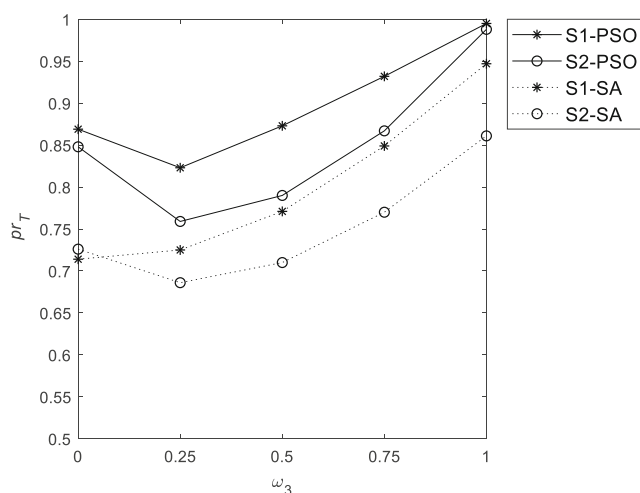


Fig. 20 Average total performance ratios for different weights of the fitness function by changing ω_3

but the ratios are decreased as we increase ω_2 . When only the tower coverage ratio is maximized (i.e. $\Omega = [0 \ 0 \ 1]$), we achieve almost 100% coverage of the tower. However, when only the flight time ratio or image quality ratio are maximized (i.e. $\Omega = [1 \ 0 \ 0]$ or $\Omega = [0 \ 1 \ 0]$), we do not achieve 100% on these ratios. The 100% flight time ratio corresponds to a zero flight-time. Similarly, a 100% image quality ratio corresponds to all waypoints located at the surface of the transmission tower creating a path collision risk.

5 Conclusion

We proposed a PSO-based algorithm to find an efficient UAS flight path for inspecting electric transmission towers. Our experimental results showed that the PSO-based algorithm outperformed the SA-based algorithm in terms of the average total performance ratio and computational time. When the flying area is divided horizontally and the waypoints are determined using the PSO-based algorithm, the average total performance ratio and computational time were superior. One advantage of the PSO-based algorithm is that it can provide multiple best solutions. Under equal weight values of the objective function and a swarm size of 20, we obtained two best solutions with the same total performance ratio ($pr_T = 0.84$), a solution A with 12 waypoints and performance ratios $pr_1 = 0.85$, $pr_2 = 0.70$, $pr_3 = 0.98$ and a solution B with 16 waypoints and performance ratios $pr_1 = 0.82$, $pr_2 = 0.73$, $pr_3 = 0.99$. The solution A would be preferable if fewer waypoints were required to reduce complexity of the navigation and if reducing flight time was more critical than obtaining images with higher quality. However, the solution B would

be preferable if higher tower coverage and higher image quality were more critical than navigation time. We also noticed that the swarm size did not affect significantly the quality of the solutions.

When we maximized the flight time ratio only, we obtained a flight path that provided 93% of flight time ratio. But, the image quality and tower coverage ratios were low, 20% and 0% respectively. Similarly, by maximizing the image quality ratio, we obtained a flight path that provided 85% of image quality ratio. But, the flight time and tower coverage ratios were 87% and 10%, respectively. By maximizing the tower coverage, we obtained a flight path that covered the tower 99%. But, the flight time and image quality ratios were 81% and 31%, respectively. By maximizing the three performance ratios simultaneously, we obtained a flight path that provided flight time, image quality, and tower coverage ratios of 85%, 70%, and 98%, respectively. These results show that the use of the proposed model can provide a flight path that comprises a good balance over the three performance ratios.

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

References

- Common, D.: Drones go commercial, take on tasks from industry to farming. CBC News. <https://www.cbc.ca/news/technology/drones-go-commercial-take-on-tasks-from-industry-to-arming-1.2657036?cmp=rss>. Accessed 10 Dec. 2017 (2014)
- Marshall, M.: Aerial infrared line inspection. In: Proceedings of the Rural Electric Power Conference, pp. A3/1–A3/2 (1999)
- U.S. Bureau of Labor Statistics: National census of fatal occupation injuries in 2015. Website. <https://www.bls.gov/news.release/pdf/cfoi.pdf> (2016). Accessed 10 Dec. 2017
- Grigsby, L.L.: Electric Power Generation, Transmission, and Distribution, 3rd edn, pp. 12.10–12.14. CRC Press, Boca Raton (2012)
- Montambault, S., Pouliot, N.: About the future of power line robotics. In: Proceedings of the International Conference on Applied Robotics for the Power Industry (2010). <https://doi.org/10.1109/CARPI.2010.5624466>
- Federal Aviation Administration: Unmanned Aircraft Systems. Website. <https://www.faa.gov/uas> (2017). Accessed 10 Dec. 2017
- Jones, D.: Power line inspection—a UAV concept. In: Proceedings of the IEE Forum on Autonomous Systems. <https://doi.org/10.1049/ic:20050472> (2005)
- Williams, M., Jones, D.I., Earp, G.K.: Obstacle avoidance during aerial inspection of power lines. *Aircr. Eng. Aerosp. Technol.* **73**(5), 472–479 (2001). <https://doi.org/10.1108/00022660110403023>
- Electric Power Research Institute: Future inspection of overhead transmission lines. Technical update report 1016921. <http://mydocs.epri.com/docs/corporatedocuments/sectorpages/pdu/sensorsrobots/Tline%20Roadmap.pdf> (2008). Accessed 10 Dec. 2017
- Electric Power Research Institute: Emerging and future inspection of overhead transmission lines. Technical update report 1021876 (2011)

11. Wang, U.: A new weapon for storm responders: send in the drones. *Elect. Power Res. Inst. J.* **2**, 7–9 (2012)
12. Zhou, N.: Study and application of power transmission line patrol system based on microwave communication. In: *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation*, pp. 1036–1039 (2015). <https://doi.org/10.1109/ICMTMA.2015.252>
13. Dong, G., Chen, X., Wang, B., Zhang, J., Liu, L., Wang, Q., Wei, C.: Inspecting transmission lines with an unmanned fixed-wings aircraft. In: *Proceedings of the International Conference on Applied Robotics for the Power Industry*, pp. 173–174 (2012). <https://doi.org/10.1109/CARPI.2012.6473355>
14. Luque-Vega, L.F., Castillo-Toledo, B., Loukianov, A., Gonzalez-Jimenez, L.E.: Power line inspection via an unmanned aerial system based on the quadrotor helicopter. In: *Proceedings of the IEEE Mediterranean Electrotechnical Conference*, pp. 393–397 (2014). <https://doi.org/10.1109/MELCON.2014.6820566>
15. Deng, C., Wang, S., Huang, Z., Tan, Z., Liu, J.: Unmanned aerial vehicles for power line inspection: a cooperative way in platforms and communications. *J. Commun.* **9**(9), 687–692 (2014). <https://doi.org/10.12720/jcm.9.9.687-692>
16. Zhang, J., Liu, L., Wang, B., Chen, X., Wang, Q., Zheng, T.: High speed automatic power line detection and tracking for a UAV-based inspection. In: *Proceedings of the International Conference on Industrial Control and Electronics Engineering*, pp. 266–269 (2012). <https://doi.org/10.1109/ICICEE.2012.77>
17. Larrauri, J.I., Sorrosal, G., Gonzalez, M.: Automatic system for overhead power line inspection using an unmanned aerial vehicle-RELIFO project. In: *Proceedings of the International Conference on Unmanned Aircraft Systems*, pp. 244–252 (2013). <https://doi.org/10.1109/ICUAS.2013.6564696>
18. Montambault, S., Beaudry, J., Toussaint, K., Pouliot, N.: On the application of VTOL UAVs to the inspection of power utility assets. In: *Proceedings of the International Conference on Applied Robotics for the Power Industry* (2010). <https://doi.org/10.1109/CARPI.2010.5624443>
19. Pagnano, A., Hopf, M., Teti, R.: A roadmap for automated power line inspection. Maintenance and repair. *Proc. CIRP* **12**, 234–239 (2013). <https://doi.org/10.1016/j.procir.2013.09.041>
20. Katrasnik, J., Pernus, F., Likar, B.: A survey of mobile robots for distribution power line inspection. *IEEE Trans. Power Delivery* **25**(1), 485–493 (2010). <https://doi.org/10.1109/TPWRD.2009.2035427>
21. Federal Aviation Administration: Summary of small unmanned aircraft rule (part 107). Website. https://www.faa.gov/uas/media/Part_107_Summary.pdf (2016). Accessed 10 Dec. 2017
22. Guerrero, J.A., Bestaoui, Y.: UAV path planning for structure inspection in windy environments. *J. Intell. Robot. Syst.* **69**, 297–311 (2013). <https://doi.org/10.1007/s10846-012-9778-2>
23. Franco, C.D., Buttazzo, G.: Coverage path planning for UAVs photogrammetry with energy and resolution constraints. *J. Intell. Robot. Syst.* **83**, 445–462 (2016). <https://doi.org/10.1007/s10846-016-0348-x>
24. Sujit, P.B., Hudzietz, B.P., Saripalli, S.: Route planning for angle constrained terrain mapping using an unmanned aerial vehicle. *J. Intell. Robot. Syst.* **69**, 273–283 (2013). <https://doi.org/10.1007/s10846-012-9729-y>
25. Belhadj, C.A., Dawoud, M.M., Maalej, N., Habiballah, I.O., Abdel-Galil, T.K.: Electric & magnetic field assessment for live-line workers next to A 132 KV transmission line conductor. In: *Proceedings of the IEEE/PES Transmission and Distribution Conference and Exposition: Latin America* (2008). <https://doi.org/10.1109/TDC-LA.2008.4641840>
26. Duveiller, G., Defourny, P.: A conceptual framework to define the spatial resolution requirements for agricultural monitoring using remote sensing. *Remote Sens. Environ.* **114**, 2637–2650 (2010). <https://doi.org/10.1016/j.rse.2010.06.001>
27. Zsedrovičs, T., Bauer, P., Hiba, A., Nemeth, M., Pencz, B.J.M., Zarandy, A., Vanek, B., Bokor, J.: Performance analysis of Camera rotation estimation algorithms in multi-sensor fusion for unmanned aircraft attitude estimation. *J. Intell. Robot. Syst.* **84**, 759–777 (2016). <https://doi.org/10.1007/s10846-016-0346-z>
28. Wang, D., Lim, K.B., Kee, W.L.: Geometrical approach for rectification of single-lens stereovision system with a triprism. *Mach. Vis. Appl.* **24**, 821–833 (2013). <https://doi.org/10.1007/s00138-012-0467-8>
29. Catmull, E.E.: A subdivision algorithm for computer display of curved surfaces. Ph.D. dissertation (1974)
30. Hughes, J.F., Dam, A.V., McGuire, M., Sklar, D.F., Foley, J.D., Feiner, S.K., Akeley, K.: *Computer Graphics—Principles and Practice*, 3rd edn. Addison-Wesley, Reading (2013)
31. Kennedy, J., Eberhart, R.: Particle swarm optimization, pp. 1942–1948 (1995). <https://doi.org/10.1109/ICNN.1995.488968>
32. Dong, Y., Tang, J., Xu, B., Wang, D.: An application of swarm optimization to nonlinear programming. *Comput. Math. Appl.* **49**, 1655–1668 (2005). <https://doi.org/10.1016/j.camwa.2005.02.006>
33. Bratton, D., Kennedy, J.: Defining a standard for particle swarm optimization. In: *Proceedings of the IEEE Swarm Intelligence System*, pp. 120–127 (2007). <https://doi.org/10.1109/SIS.2007.368035>
34. Xu, S., Rahmat-Samii, Y.: Boundary conditions in particle swarm optimization revisited. *IEEE Trans. Antennas Propag.* **55**(3), 760–765 (2007). <https://doi.org/10.1109/TAP.2007.891562>
35. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**, 671–680 (1983). <https://doi.org/10.1126/science.220.4598.671>
36. Corana, A., Marchesi, M., Martini, C., Ridella, S.: Minimizing multimodal functions of continuous variables with the “simulated annealing” algorithm. *ACM Trans. Math. Softw.* **13**(3), 262–280 (1987). <https://doi.org/10.1145/29380.29864>

Hyeoncheol Baik received the B.S. degree in Aerospace Information Engineering from Konkuk University, Seoul, South Korea, in 2009, and the M.S. degree in Aeronautical & Astronautical Engineering from Purdue University, IN, USA, in 2011. He is currently pursuing the Ph.D. degree in the Department of Industrial and Systems Engineering at Auburn University, Auburn, AL, USA. Previously, he worked in the global purchasing department at General Motors Korea. His research has included the optimization of UAS operation in the electric power industry and the military, and interdisciplinary decision making in air transportation. Mr. Baik is a member of INFORMS.

Jorge Valenzuela received the Ph.D. degree in industrial engineering at the University of Pittsburgh, Pittsburgh, PA, USA, in 2000. His research interests are applied and theoretical stochastic modeling and optimization. His recent research involves stochastic models for the economics of wind power, optimization of electric power generation, and cybersecurity. He is a Professor with the Department of Industrial and Systems Engineering, Auburn University, Auburn, AL, USA, where he teaches courses on stochastic operations research and information technology. Dr. Valenzuela is a member of INFORMS and IIE.