Efficient path planning for UAV formation via comprehensively improved PSO

Radovan Tomala

Table of contents

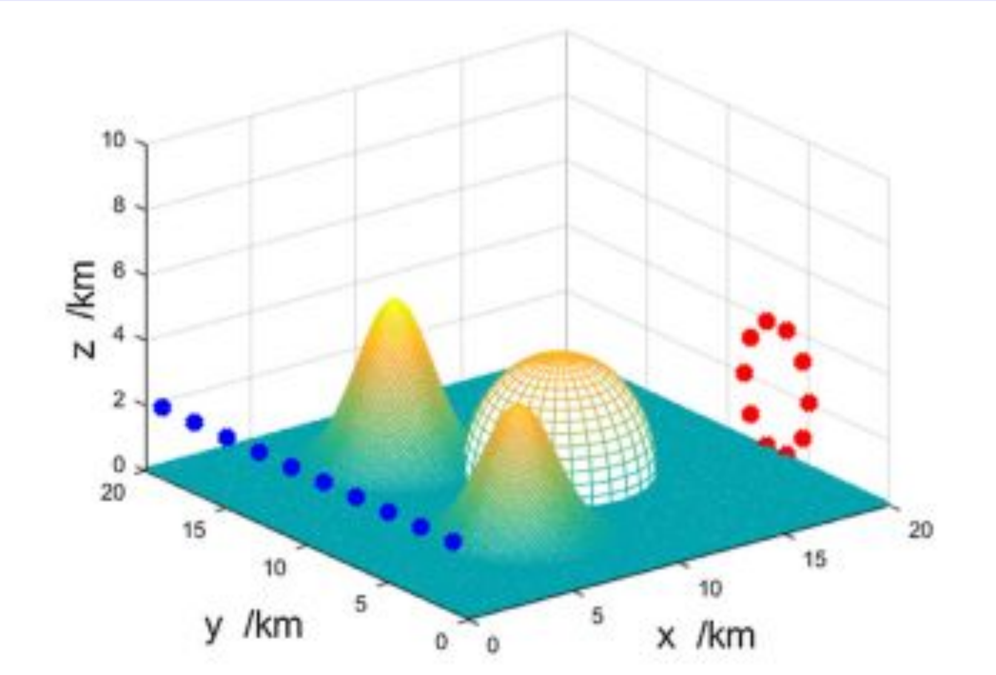
01.

Problem

UAV Formation

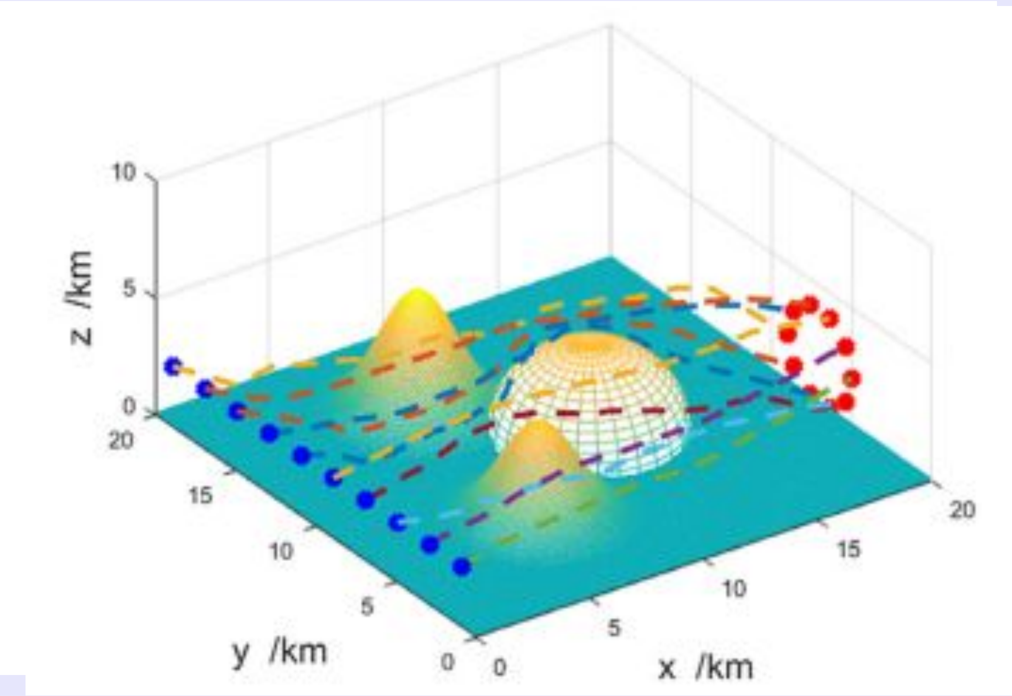
- Military purposes
- Entertainment purposes

Goal



- Plan routes for a formation of UAVs
- Blue dost - starting positions
- Red dots - final positions

Goal

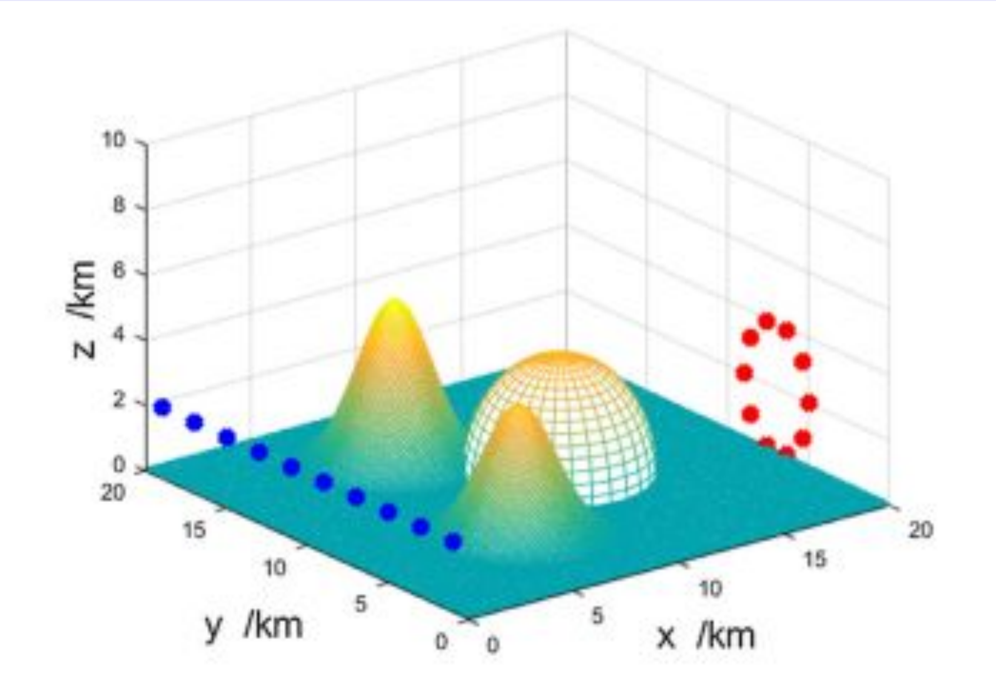


- Plan routes for a formation of UAVs
- Blue dost - starting positions
- Red dots - final positions

02.

Environment

Environment

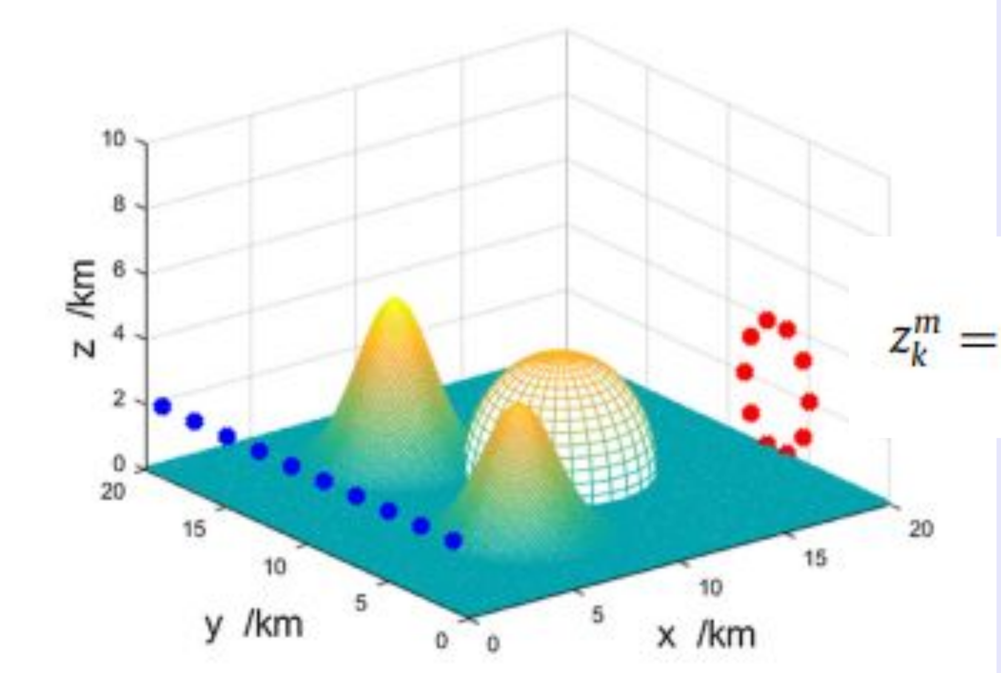


- 3D space divided to waypoints
- Threats:

  Mountains

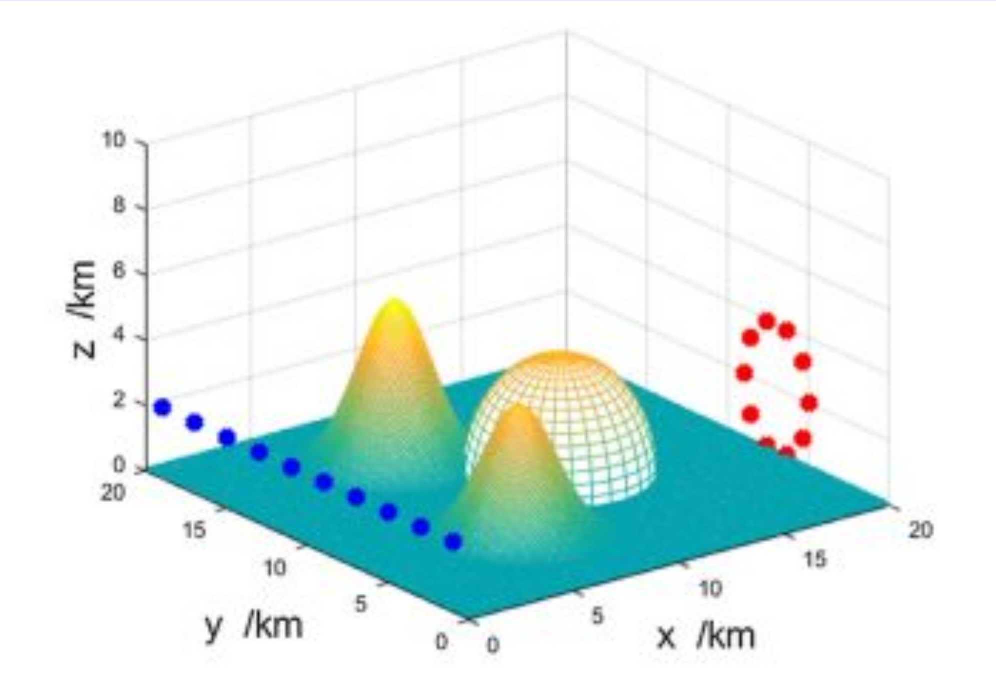
  Radars

Environment - Mountains



$$z_k^m = h_k * \exp\left(\frac{(x_k^m - x_k^{m0})^2}{x_k^t} + \frac{(y_k^m - y_k^{m0})^2}{y_k^t} \right)$$

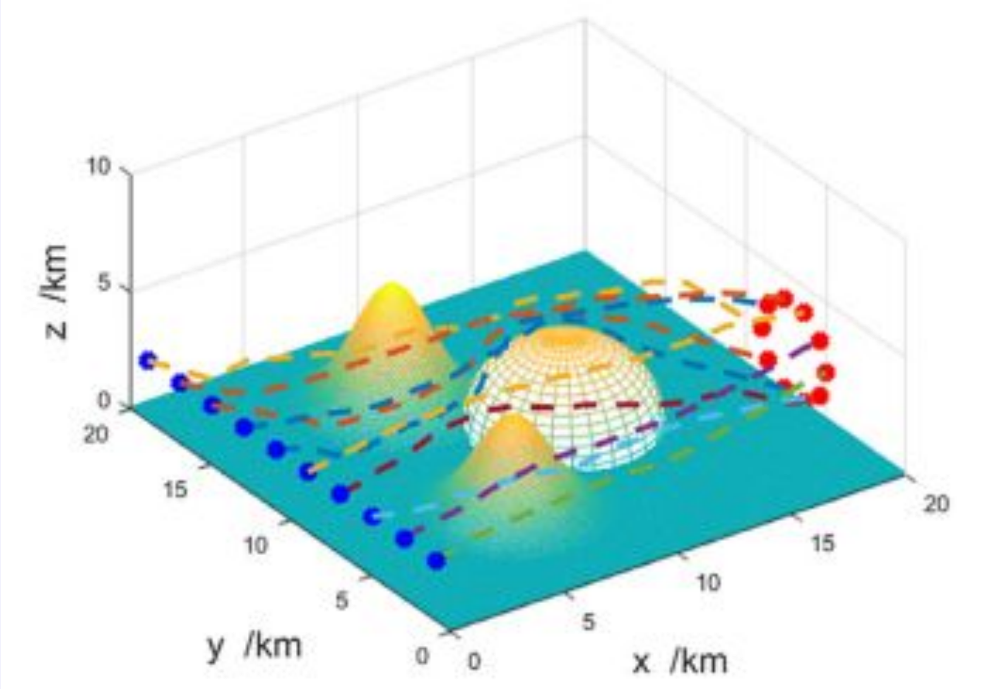
Environment - Radars



$$T_k = (x_k^r, y_k^r, z_k^r, R_k)$$

- intensity δ

Objective function



$$f = f_L + f_T + f_R + f_C$$

- path cost
- terrain cost
- radar cost
- collision cost

Objective function - Path cost

$$f_L = \frac{\sum_{i=2}^{N_w} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}}{\sqrt{\left(x_{N_w} - x_1\right)^2 + \left(y_{N_w} - y_1\right)^2 + \left(z_{N_w} - z_1\right)^2}}$$

Objective function - Terrain cost

$$f_T = \sum_{i=2}^{N_w} \sum_{j=1}^{n} A_{i,j} \quad \text{with} \quad A_{i,j} = \begin{cases} 1, & \text{if } z_{i,j} \leq z_{k,i,j}^{m} \\ 0, & \text{otherwise} \end{cases}$$

- segments between waypoints divided into n parts

Objective function - Radar cost

$$f_R = \sum_{i=2}^{N_w} \sum_{j=1}^{n} B_{i,j} \quad \text{with} \quad B_{i,j} = \begin{cases} (\delta / D_{i,j})^4, & \text{if } D_{i,j} \leq R_k \\ 0, & \text{otherwise} \end{cases}$$

$$D_{i,j} = \sqrt{(x_{i,j} - x_k^r)^2 + (y_{i,j} - y_k^r)^2 + (z_{i,j} - z_k^r)^2}$$

- segments between waypoints divided into n parts

Objective function - Collision cost

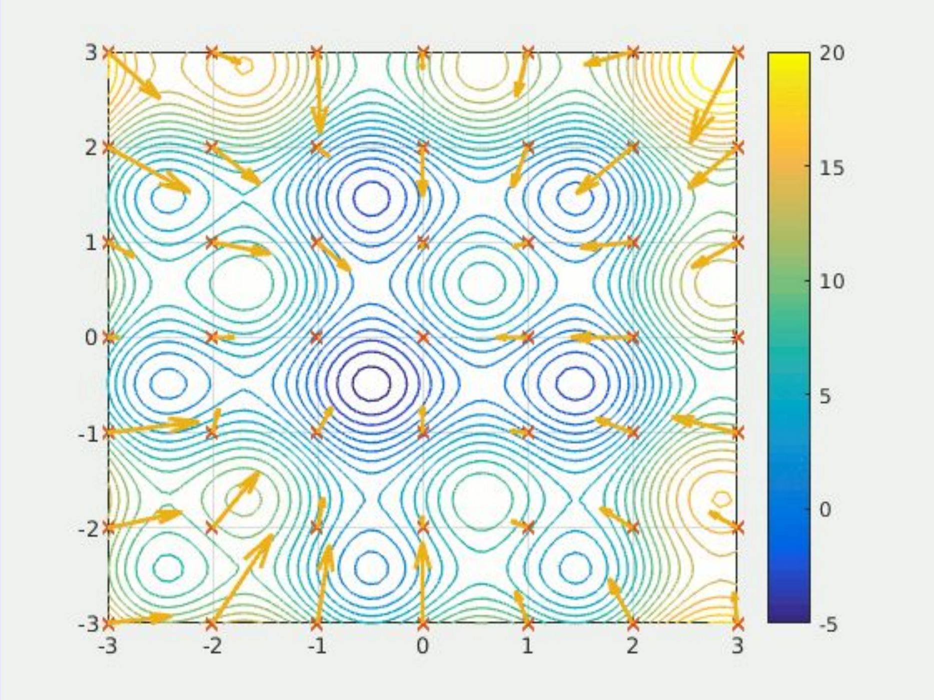
$$f_C = \sum_{i=2}^{N_w} \sum_{j=1}^{n} C_{i,j} \quad \text{with} \quad C_{i,j} = \begin{cases} 1, & \text{if } d_{i,j} \leq \bar{d} \\ 0, & \text{otherwise} \end{cases}$$

- segments between waypoints divided into n parts
- $d_{i,j}$ = shortest distance to other path points of different UAV
- $\bar{d}$ - safe distance between UAV

03.

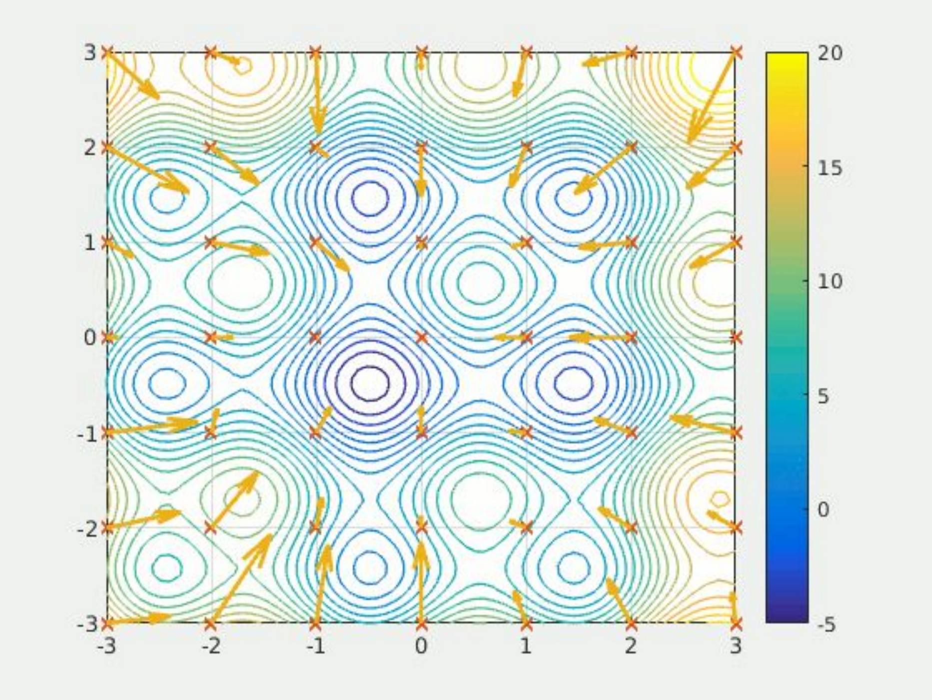
PSO

Particle Swarm Optimization



- population based
- swarm of candidate solutions
- each candidate has its velocity
- velocity = search direction

Particle Swarm Optimization



- personal best positions
- global best position

Particle Swarm Optimization Position and Velocity Updates

$$\begin{cases} \mathbf{v}_{i,j}(t+1) = w \cdot \mathbf{v}_{i,j}(t) + c_1 \cdot r_1 \cdot \left(\mathbf{p}_{i,j,best}(t) - \mathbf{p}_{i,j}(t)\right) \\ \quad +c_2 \cdot r_2 \cdot \left(\mathbf{g}_{j,best}(t) - \mathbf{p}_{i,j}(t)\right) \\ \mathbf{p}_{i,j}(t+1) = \mathbf{p}_{i,j}(t) + \mathbf{v}_{i,j}(t+1) \end{cases}$$

$$w = w_{\max} - \frac{(w_{\max} - w_{\min})\,t}{T}$$

- w - inertia weight
- c_1, c_2 - acceleration coefficients
- r_1, r_2 - random numbers

Particle Swarm Optimization in 3D pathfinding

- particle = one whole path from start to finish
- D - number of waypoints within particle

$$\mathbf{p}_i = \left(\mathbf{p}_{i,1}, \ldots, \mathbf{p}_{i,D}\right)^T = \left(\left(p^x_{i,1}, p^y_{i,1}, p^z_{i,1}\right), \ldots, \left(p^x_{i,D}, p^y_{i,D}, p^z_{i,D}\right)\right)^T$$

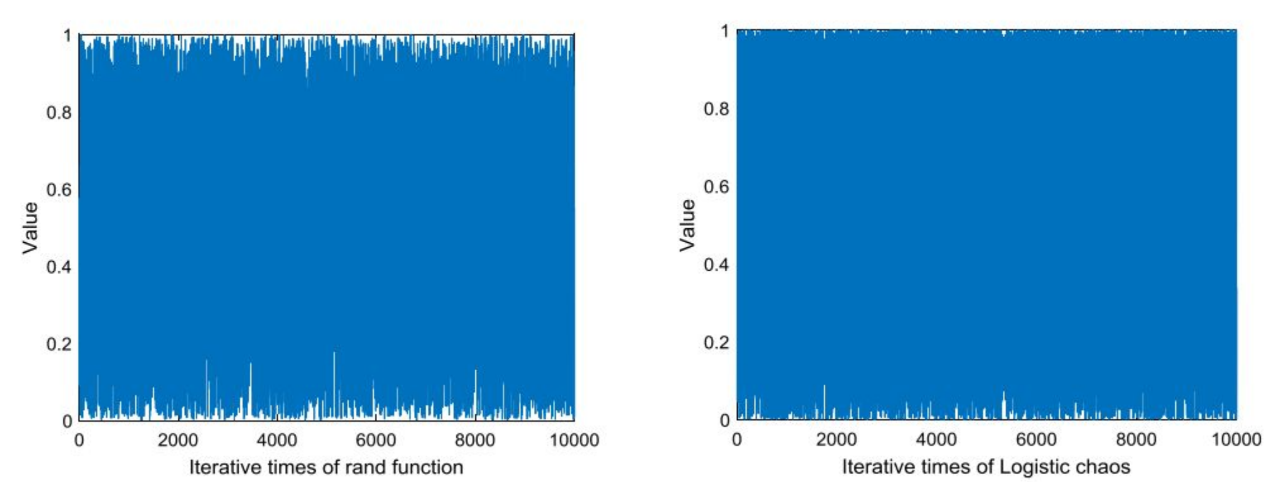
$$\mathbf{v}_i = \left(\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,D}\right)^T = \left(\left(v^x_{i,1}, v^y_{i,1}, v^z_{i,1}\right), \ldots, \left(v^x_{i,D}, v^y_{i,D}, v^z_{i,D}\right)\right)^T$$

04.

Improved PSO

Chaos-based particle initialization

- more uniform initial distribution is better
- Logistic map $x_{n+1} = \mu x_n (1 - x_n)$
- μ = 4 => system produces chaotic signal



Adaptive parameter adjustment

- c_1, c_2 - determine exploration/exploitation
- explore at start, exploit in the end

$$c_1 = c_{max} - \frac{(c_{max} - c_{min})\,t}{T}$$

$$c_2 = c_{min} + \frac{(c_{max} - c_{min})\,t}{T}$$

$$\begin{cases} \mathbf{v}_{i,j}(t+1) = w \cdot \mathbf{v}_{i,j}(t) + c_1 \cdot r_1 \cdot \left(\mathbf{p}_{i,j,best}(t) - \mathbf{p}_{i,j}(t)\right) \\ \quad + c_2 \cdot r_2 \cdot \left(\mathbf{g}_{j,best}(t) - \mathbf{p}_{i,j}(t)\right) \\ \mathbf{p}_{i,j}(t+1) = \mathbf{p}_{i,j}(t) + \mathbf{v}_{i,j}(t+1) \end{cases}$$

Maximum velocity design

- V_1, V_2 - start/end maximum velocities
- bigger in the beginning to allow for wider exploration
- smaller in the end to refine existing solution

$$V_{\max} = V_1 - \frac{(V_1 - V_2)\, t}{T}$$

Position updating strategy

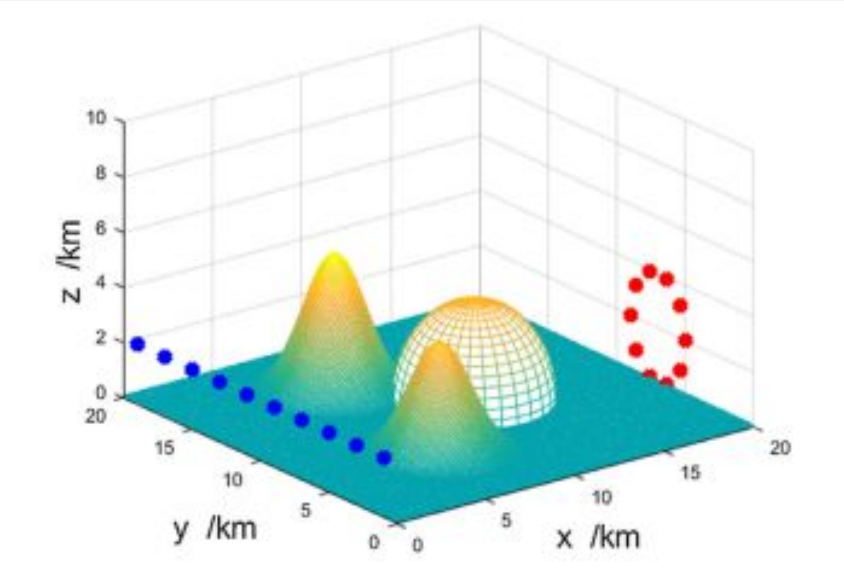
- sort particles by fitness
- divide sorted particles into groups p_{large} and p_{small}
- replace particles with large fitness by mutating particles with small fitness
- r3 - random number
- a - position offset constant

$$\begin{cases} \mathbf{p}_{large} = \mathbf{p}_{small} + a \cdot r_3 \\ \mathbf{v}_{large} = \mathbf{v}_{small} \end{cases}$$

05.

Algorithm

Algorithm - Environment Construction



1. Set flying space
2. Set mountains
3. Set radars
4. Set starting and destination positions

Algorithm - Main Loop

for i = 1 to UAVnumber

Initialize PSO for given UAV

for j = 1 to T

Update PSO parameters (inertia weight...)

Generate new position and velocity

Calculate **fitness**

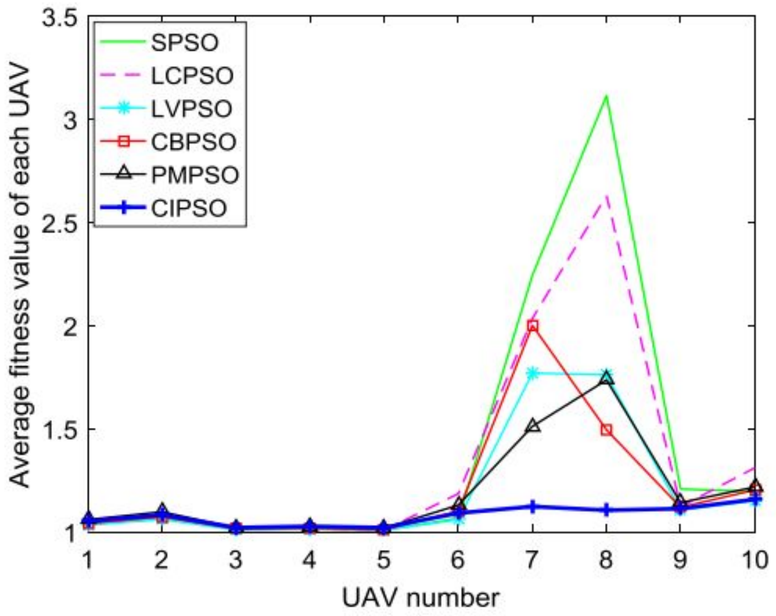
Select and remember **gbest and pbest**

Adopt **mutation strategy** and generate new position

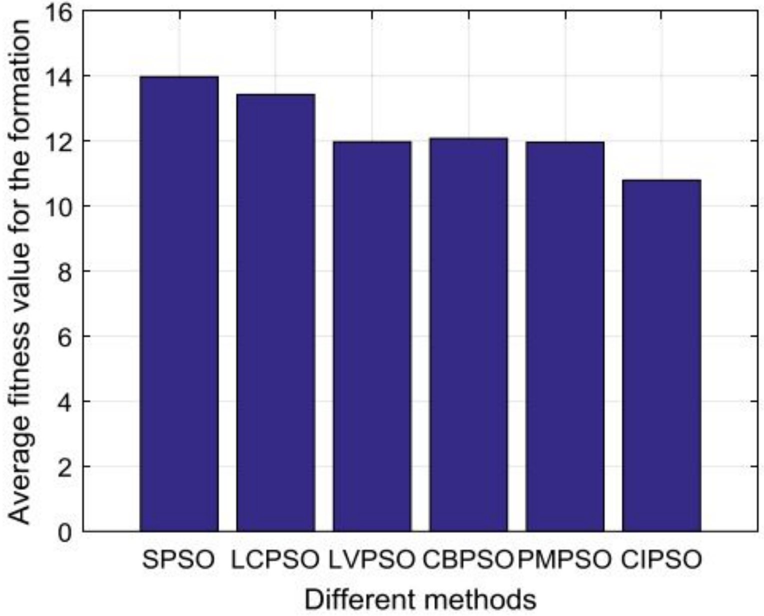
06.

Performance

Solution optimality

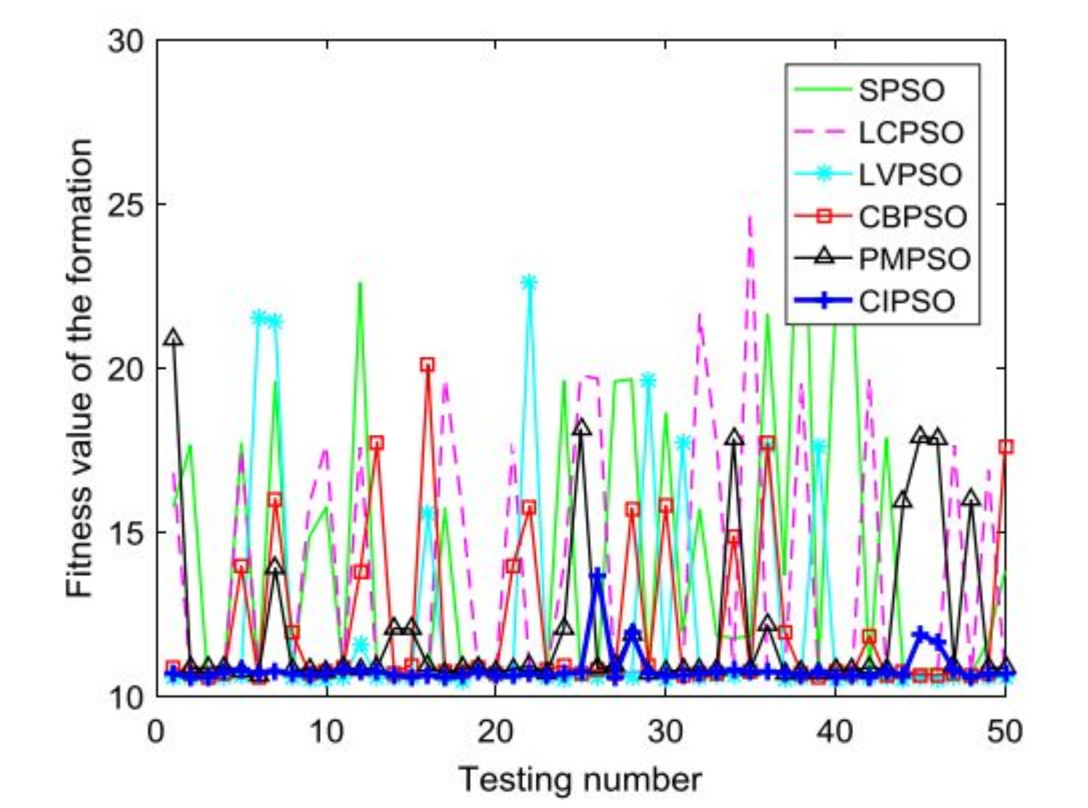


(a) AFV of each UAV

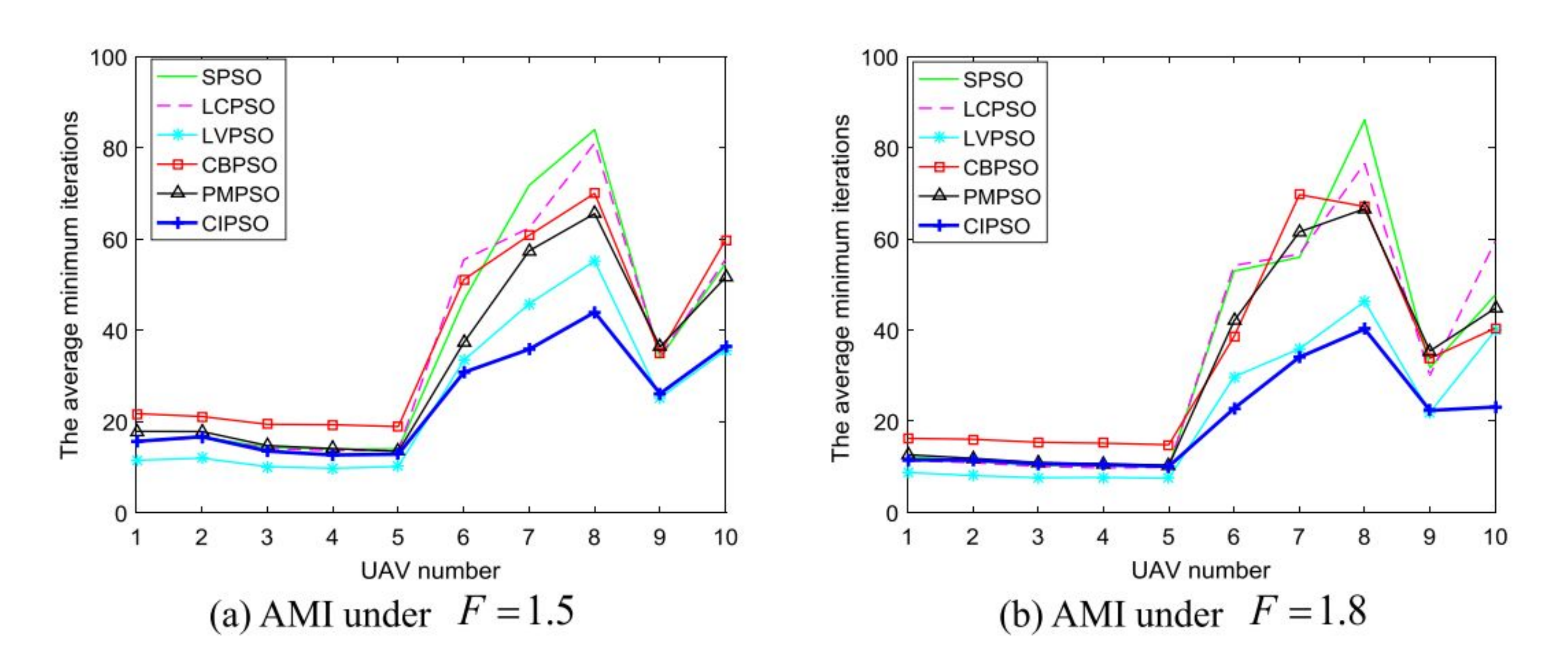


(b) AFV of the formation

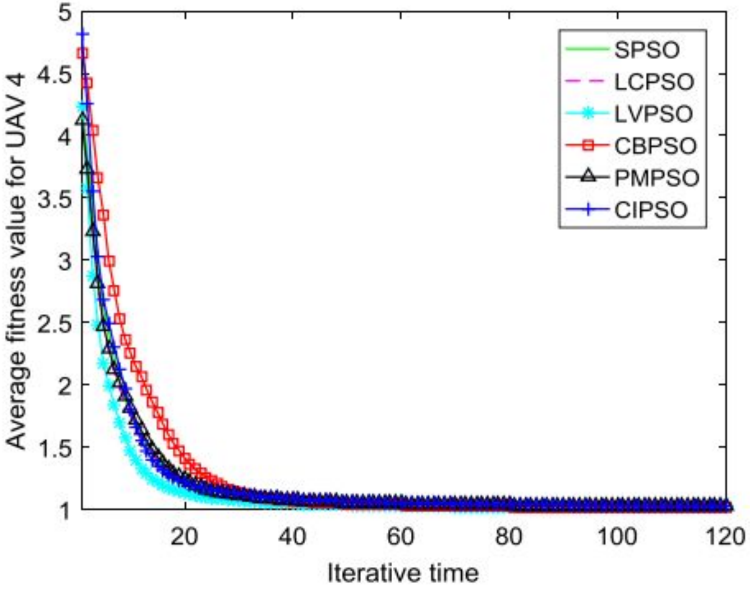
Solution optimality



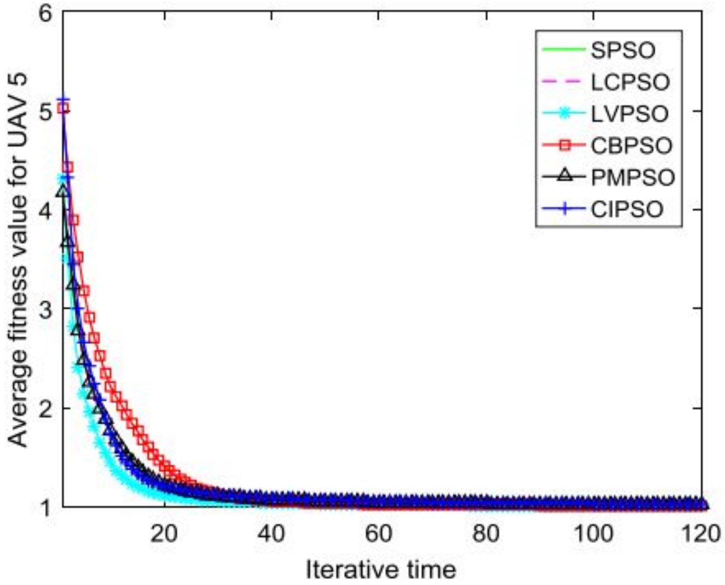
Convergence Speed



Convergence Speed

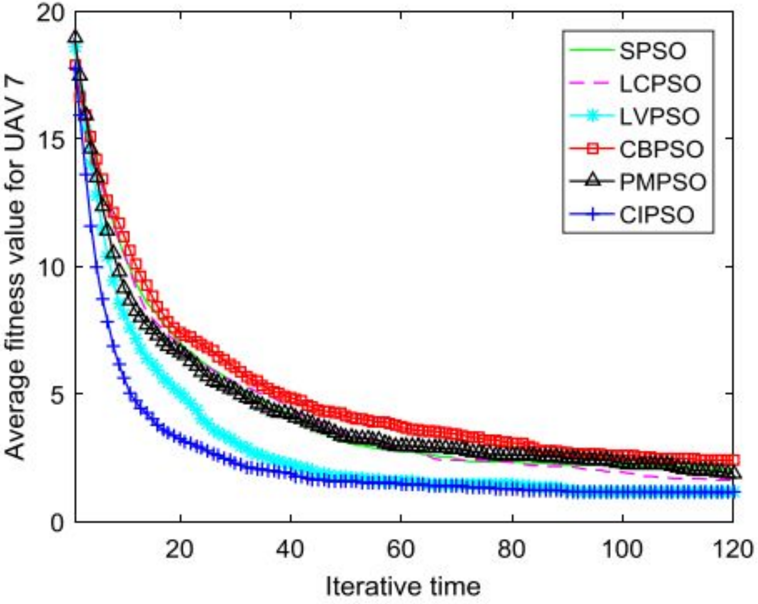


(a) AFV of UAV 4

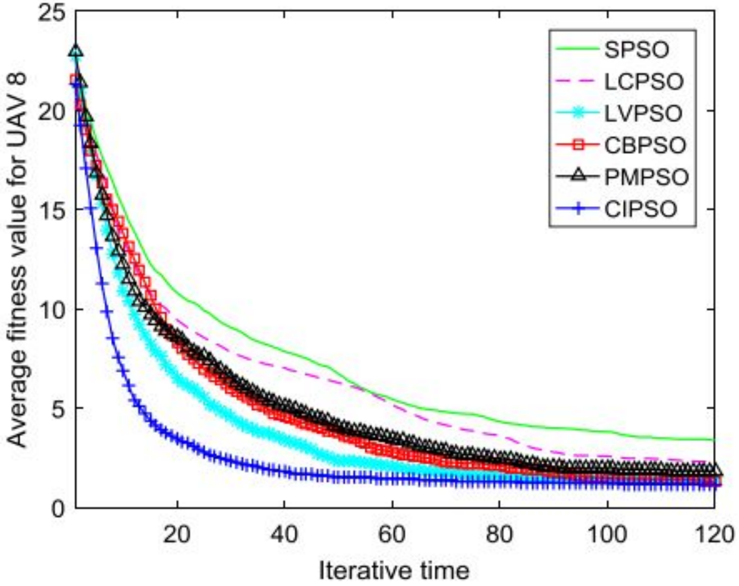


(b) AFV of UAV 5

Convergence Speed



(c) AFV of UAV 7



(d) AFV of UAV 8

Success Rate

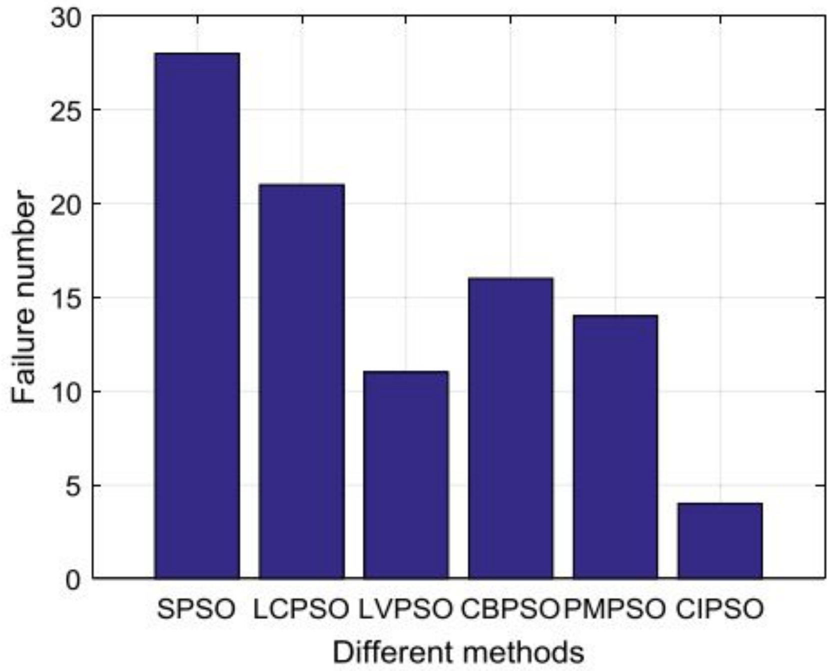


Fig. 9. Failure path number comparison.

Running Time

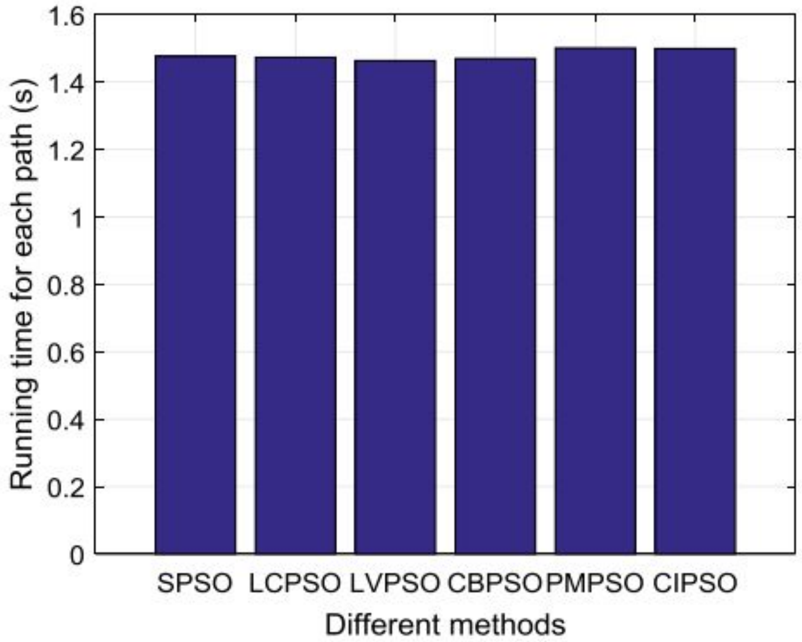
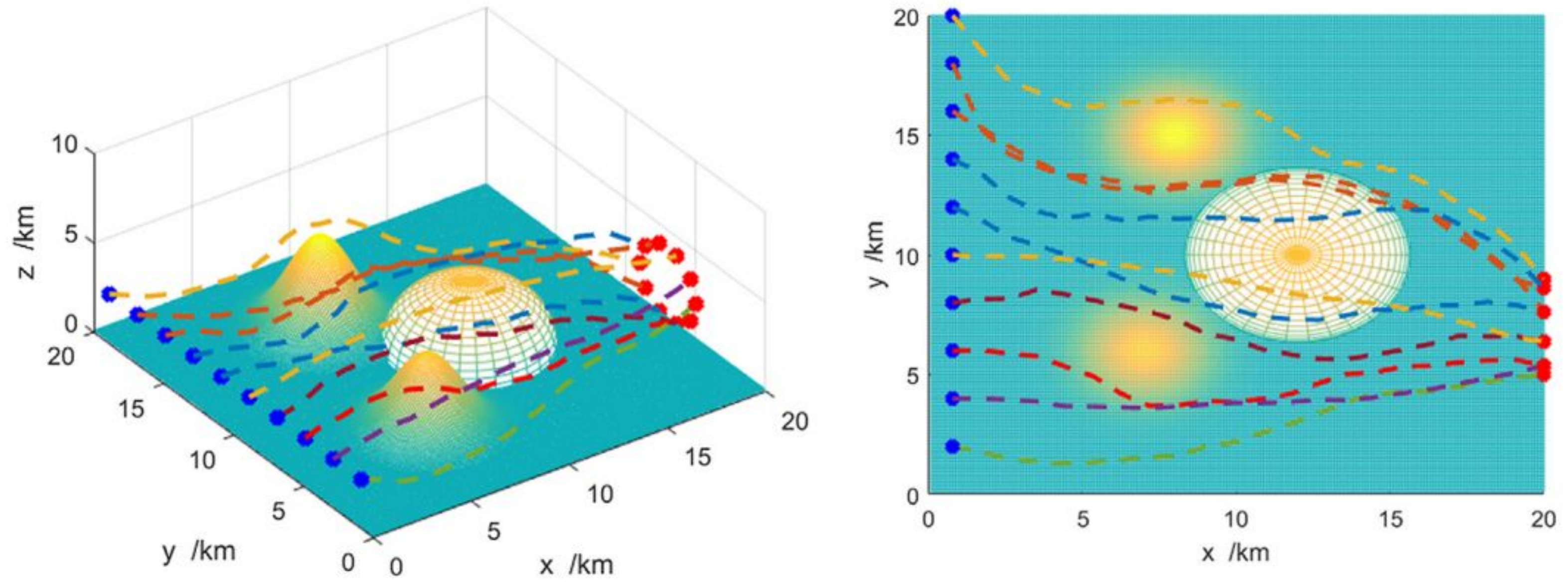


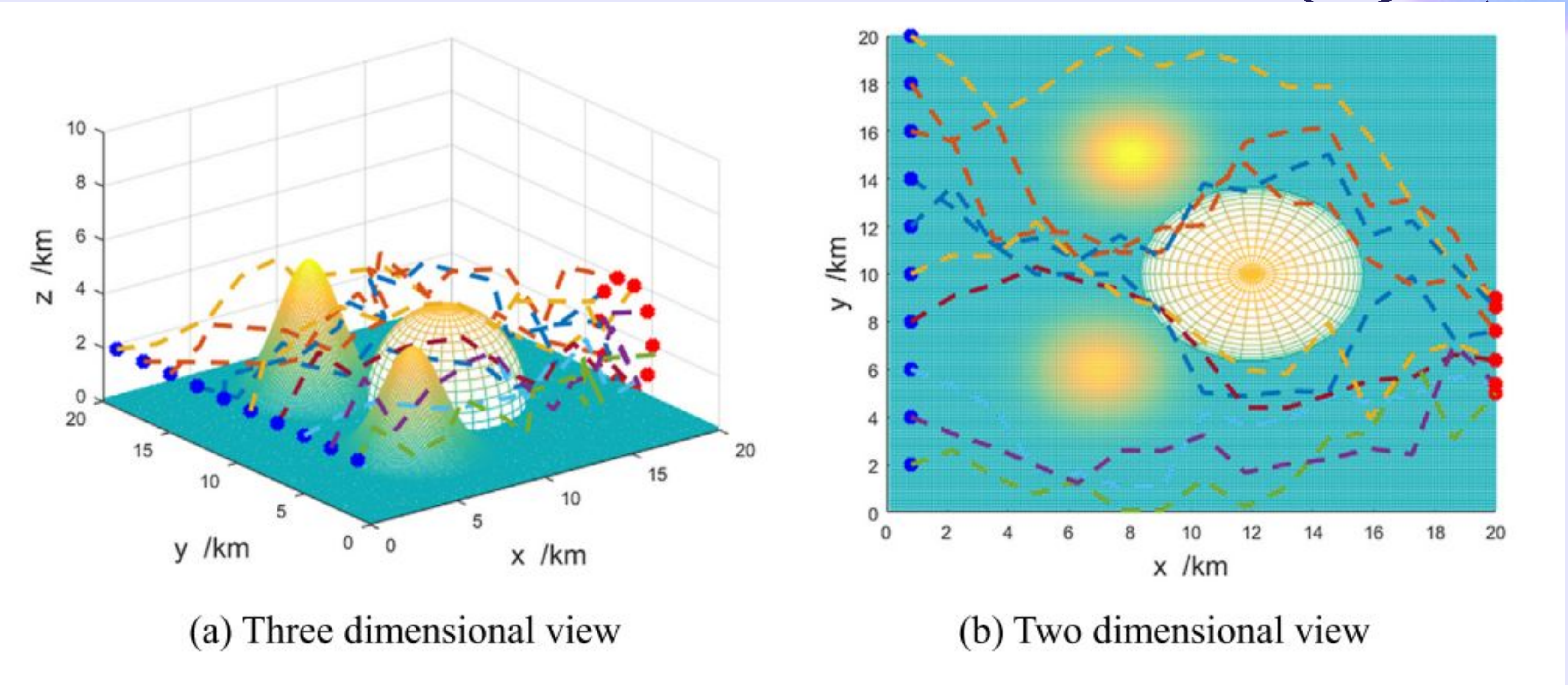
Fig. 10. Average running time comparison.

Comparison with MGA - CIPSO Paths

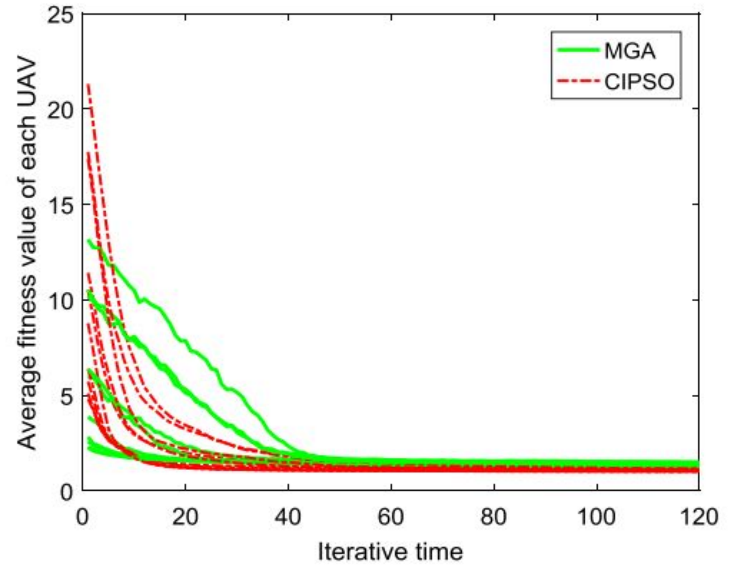


(a) Three dimensional view (b) Two dimensional view

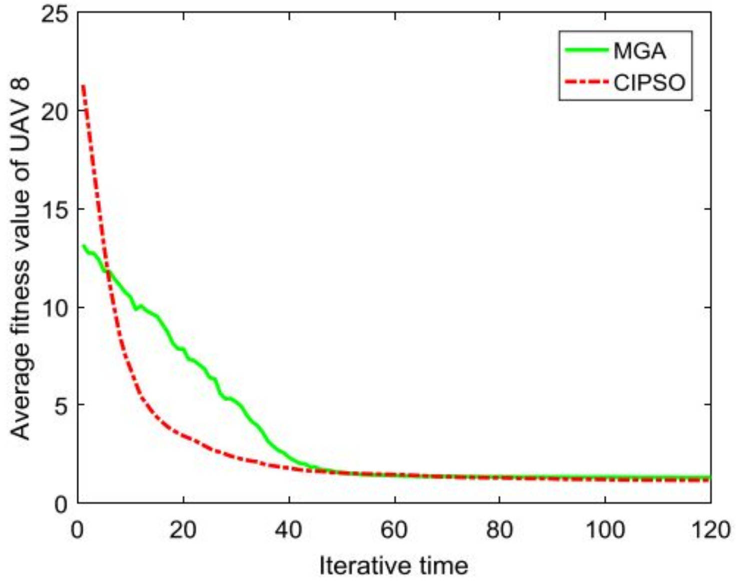
Comparison with MGA - MGA Paths



Comparison with MGA

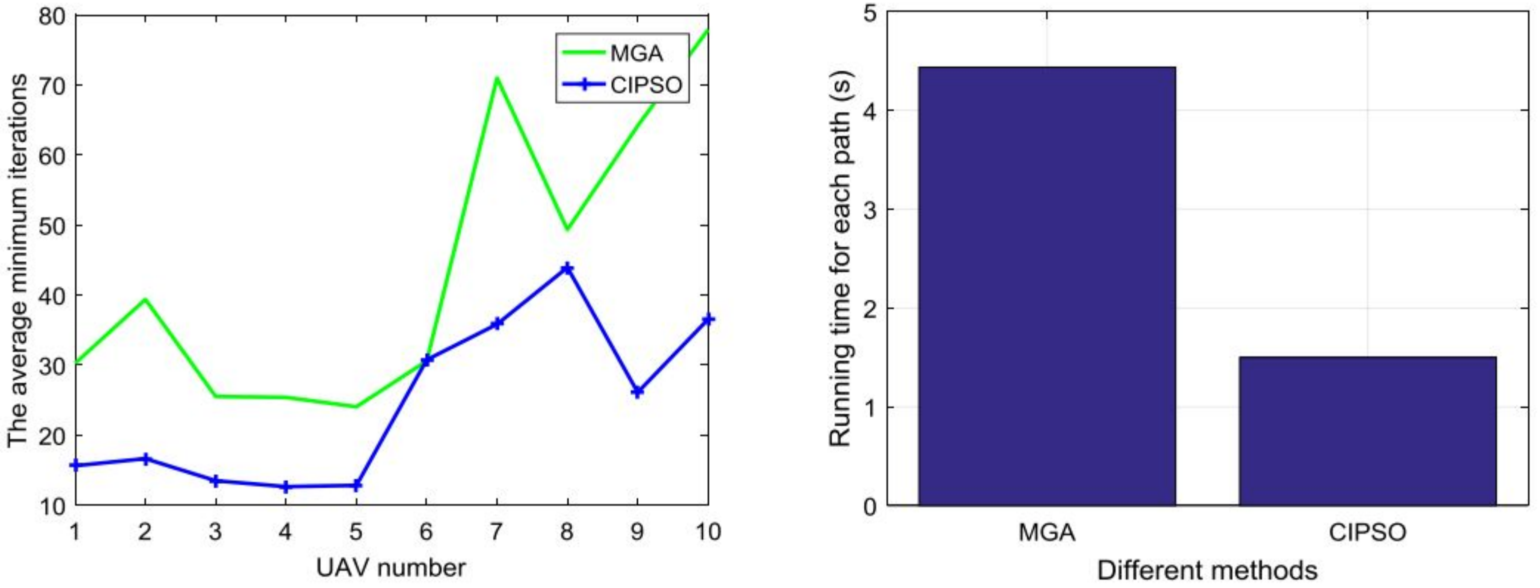


(b) AFV comparison between MGA and CIPSO



(b) AFV comparison of UAV 8

Comparison with MGA



(a) AMI comparison

(b) Running time comparison

Thank you

Do you have any questions?