Neurocomputing 313 (2018) 229–246

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning

Daifeng Zhang, Haibin Duan*

State Key Laboratory of Virtual Reality Technology and Systems, School of Automation Science and Electrical Engineering, Beihang University (BUAA), Beijing 100083, PR China

ARTICLE INFO

Article history:
Received 27 January 2018
Revised 26 May 2018
Accepted 20 June 2018
Available online 30 June 2018

Communicated by Dr. Yuan Yuan

Keywords:
Multi-UAV cooperative path planning
Time stamp segmentation
Swarm intelligence optimization
Social-class strategy
Pigeon-inspired optimization

ABSTRACT

Path planning is a significant issue for the safe flight of unmanned aerial vehicles (UAVs). In the situation of multiple UAVs, the problem is even more challenging due to the tough manipulation of coordination and constrains. In this paper, a novel multi-UAV path planning model is developed which is based on the time stamp segmentation (TSS) technique. Other than the traditional methods, the TSS model utilizes the common time bases to simplify the handling of multi-UAV coordination cost. Then, a novel social-class pigeon-inspired optimization (SCPIO) algorithm is proposed as the solver of optimal search on the TSS model. The social-class strategy is utilized to enhance the convergence capabilities of the standard PIO. The efficiency of the proposed method is verified through the comparative experiments and the performance profiles (PP). Integrated experiment in a 3D environment demonstrates the reliability of the proposed system.

1. Introduction

Recently, the advantages of multiple unmanned aerial vehicles (UAVs) in defensive and civil fields have prompted the scholars' interests in the multi-UAV intelligence and autonomy. For example, the cooperation among different UAVs is an imperative issue for the mission success since the cooperative strategy could significantly promote the mission efficiency [1] and outstandingly decrease the risks in dynamical environments [2] through active interactions of the UAV swarms. On this premise, a reliable and secure multi-UAV path planner is a persistent demand. However, unlike the minor constrains of single UAV flight, the multi-UAV cooperative flight is much more challenging due to its multi-dimension, multi-constraint and complex coordination demands [3]. For example, how to design the desired paths to guide a swarm of UAVs to match up the arrival status consistently is a main dilemma [4,5]. Normally, a proper path planning model is inevitable for the problem simplification. Besides, an efficient algorithm is also necessary to implement the continuous search for the suitable paths. In this paper, the main purpose is to find some reliable solutions to the aforementioned aspects.

The development of UAV path planning has been prompted over the past few years and various practical approaches have been proposed. Traditional methods include the A* [6] algorithm, artificial potential field (APF) [7], Voronoi diagram [8], Dubins curve [9] and mesh grid models [10] etc. But most of them merely focus on the utility for single UAV and may not be suitable in the multi-UAV situations. Some reasons surely account for these cases. The approaches based on the mesh grids such as A* algorithm and Voronoi diagram have only concerned with the waypoint designations for a single UAV. Nevertheless, due to lack of common references used for multi-UAV coordination, it's hard to fuse the waypoint information from different UAVs to manipulate the coordination constraints of the specific multi-UAV tasks such that all the UAVs should arrive simultaneously [4–5]. The APF method is suitable to realize the collision-free and cohesive flight among UAVs. However, the trend to local minimum blocks its applications in complicated circumstances [11]. In [5], Wang et al. established a simplified UAV model and used a nonconvex optimal-control technology to solve the minimum-time multi-UAV cooperative trajectory planning. The sequential convex programming (SCP) was acknowledged as the main solver. More specifically, the involved concept that the UAVs' states could be utilized in the path planning is rational and enlightening, since it applied the additional reference sources to supplement the traditional models, which could simplify the implementation of multi-UAV cooperation. But the consumption of SCP is enormous and prevents its usage in practical situations. An intuitive way to overcome the aforementioned dilemma is to provide a common reference for the computation of multi-UAV coordination constraints and the evaluation to the candidate paths. Assume that all the UAV paths share the same num-

* Corresponding author.
E-mail address: hbduan@buaa.edu.cn (H. Duan).

https://doi.org/10.1016/j.neucom.2018.06.032
0925-2312/

Nomenclature

T_t, T_a	takeoff and arrival time
t_s	time stamp
m, n, i, k, t	index of UAV, time stamp, solution, social class and iteration
$S_{m,x}$, $S_{m,y}$	elements of takeoff point of mth UAV in 2D global frame
$\mathbf{P}_{m,n}$	point of mth UAV in 2D global frame at nth time stamp
$\mathbf{V}_{m,n}$	velocity of mth UAV in 2D global frame at nth time stamp
$X_gO_gY_g$	2D global frame
ψ_m	deflection angle between the global frame and local frame of mth UAV
$\bar{P}_{m,n,x}$, $\bar{P}_{m,n,y}$	point elements of mth UAV in 2D local frame at nth time stamp
$\bar{V}_{m,n,x}$, $\bar{V}_{m,n,y}$	speed elements of mth UAV in 2D local frame at nth time stamp
$\mathbf{P}_M$, $\mathbf{P}_R$, $\mathbf{P}_F$	points of mountain, radar and defensive force in 2D global frame
$F^M_{m,n}$, $F^R_{m,n}$, $F^F_{m,n}$	mountain, radar and defensive costs of mth UAV at nth time stamp
$\tilde{\mathbf{P}}_{m,n}$, $\tilde{\mathbf{P}}_F$	points of mth UAV at nth time stamp and fire force in 3D global frame
F_m^{Threat}, F_m^{Coor}, F_m^{Total}	threat, coordination and total costs of mth UAV
$L_{\max}$, $s_{\min}$	maximal voyage and minimal step length
$R_{\min}$	minimal turning radius
$T_{\max}$	maximal time of flight endurance
$\psi_{\max}$	maximum deflection angle
ω_1, ω_2	weights for threat sources and coordination limits
$\mathbf{x}_g^k(t)$, $\mathbf{x}_g(t)$	best individual of kth class and global optima at tth iteration
$\mathbf{x}_c(t)$	weighted population center at tth iteration
$S_P(t)$	set of superior pigeons at tth iteration
N_{cmax1}, N_{cmax2}	maximal iterations of two operators in PIO and SCPIO
N_k, N_{cmax}	population size of kth class and maximal iteration in PSO and DE
$\mathbf{H}_{\text{upper}}$, $\mathbf{H}_{\text{lower}}$	confidence upper bound and lower bound matrix
n_f, n_p, n_s	number of iterations, problems and algorithms
$t_{p,s}$	iteration needed for algorithm s on problem p within desired condition
γ, $\rho_s(\gamma)$	performance ratio and performance profile for algorithm s
ω, c_1, c_2	inertia weight and acceleration coefficients in PSO
D	number of waypoints/Dimension of optimizations
M	number of UAVs
$\mathbf{S}_m$, $\mathbf{E}_m$	takeoff and destination location of mth UAV in 2D global frame
$E_{m,x}$, $E_{m,y}$	elements of destination point of mth UAV in 2D global frame
$P_{m,n,x}$, $P_{m,n,y}$	point elements of mth UAV in 2D global frame at nth time stamp
$V_{m,n,x}$, $V_{m,n,y}$	speed elements of mth UAV in 2D global frame at nth time stamp
$X_mO_mY_m$	2D local frame of mth UAV
$\bar{\mathbf{P}}_{m,n}$	point of mth UAV in 2D local frame at nth time stamp
$\bar{\mathbf{V}}_{m,n}$	velocity of mth UAV in 2D local frame at nth time stamp
h, h_M	height of UAVs and height of mountain
R_{Ma}, R_{Mb}	bottom and top radius of mountain surfaces
R_R, R_F	detection radius and maximum fire range
λ, η	dangerous factors of radar and defensive force
L_c	length of current path
$V_{\min}$, $V_{\max}$	minimal and maximal airspeed
l_m	straight length between the takeoff point and destination of mth UAV
$t_{m,\min}$, $t_{m,\max}$	minimal and maximal time stamp
d_c, R_c	safety distance and maximum communication range
$\mathbf{v}_i(t)$, $\mathbf{x}_i(t)$	position and velocity of ith pigeon at tth iteration
N, L	population size and number of classes
f_{cost}, N_{cost}	cost function and its computational load
R, R_k	positive factors of the map and compass operator in PIO and kth class of SCPIO
$N_P(t)$	number of pigeons with superior values at tth iteration
$\mathbf{H}_{\text{means}}$, $\mathbf{H}_{\text{var}}$	means matrix, variance matrix
p, s	index of problem and solver
n_r	sample volume of duplicate test
f_L	best value obtained within all the algorithms on the benchmark problem
κ	confidence level of standard normal distribution
P_{cr}, F	crossover probability and mutation factor in DE

ber of waypoints and the same total arrival time, then it's possible to determine the arrival times of waypoints for each UAV. In this paper, we assume that the increment of waypoint arrival time or the time stamp is equal for each UAV. Then the time bases at which multi-UAV waypoints are updated could be exploited as the temporal references for the cooperative path planning. Hence, we can obtain the coordination costs through receiving the waypoints from other UAVs at the common time stamp and use a time stamp segmentation (TSS) [12] method to establish the basic model of multi-UAV path planning.

Swarm intelligence optimizations (SIOs) have been widely used to obtain the global optimal solutions in complicated networks and systems. The advantages of SIOs include the self-organization property to search for the global optima and the adaptation to various models. In terms of the multi-UAV path planning, the constant number of waypoints and the candidate paths could be regarded as a SIO's dimension and solutions, respectively. Then we can use the SIO algorithms to solve the path planning problem by arranging optimal waypoints and other relevant variables to the candidate paths. Compared with the aforementioned algorithms, SIOs could further exploit the common time bases to simplify the computation of multi-UAV coordination costs and update the multi-UAV solutions simultaneously. That is, the parallel optimization on different UAVs is available. Therefore, SIOs are powerful tools in the research field of multi-UAV path planning. Typical SIO algorithms include Particle Swarm Optimization (PSO) [13–15], Ant Colony Optimization (ACO) [4], Differential Evolution (DE) [16,17], Imperialist Competitive Algorithm (ICA) [18], Wolf Pack Search Algorithm (WPSA) [19], Central Force Optimization (CFO) [20] and

Artificial Bee Colony (ABC) [21]. Recently, a novel SIO namely Pigeon-inspired Optimization (PIO) [22] was developed and firstly applied for the path planning issue of air robots. Two kinds of operators were introduced to reinforce the PIO's searching abilities. Through comparative simulations, PIO surely outweighs some homogenous methods in the field of path planning. Moreover, PIO has stimulated its usage in various research fields such as image processing [23], gliding trajectory tracking for hypersonic vehicles [24] and objective recognition [25]. However, there are still some shortcomings about the standard PIO. For example, the balance between the convergence precision and computation cost are important for a SIO. A general method to guarantee the balance is expanding the cover plant of candidate solutions and searching for the global optima by means of the detected information [26]. In other words, the balance between the exploration and the exploitation should be cautiously considered. Normally, we keep the equilibrium state by improving the population diversity and the evolutionary strategies since the abundant diversity contributes to the extension of search span while a proper evolution scheme is conducive to the exploitation of an algorithm. According to Duan and Wang [23], the standard PIO suffers from less population search span and fast prematurity in the field of image processing, which illustrates the possible failure of standard PIO. In [27], Zhang and Duan modified the standard PIO using the predator–prey concept to optimize 3D flight paths for uninhabited combat aerial vehicles (UCAVs). The predator–prey competition schemes were embedded into the PIO's evolution strategy to obtain the global optima. However, the programming model was still based on single UAV. As a matter of fact, practical observations and theoretical analysis have indicated that the pigeon flocks may have leader-follower hierarchies during the in-flight process [28–32]. Inspired by this phenomenon and associating with the scheme of PIO, we believe that the multi-class interaction strategy could be exploited to improve the population diversity and thus promote the searching ability for global optima.

The main contributions in this paper involve the following aspects. Firstly, a novel TSS model is developed for the evaluation of multi-UAV path planning. Other than the traditional waypoint models, the TSS model could additionally provide the desired velocity and motion time bases as the references for multi-UAV coordination. Secondly, a novel social-class PIO (SCPIO) algorithm is proposed to search for the feasible paths on the TSS model. The social-class strategy of in-flight pigeons is utilized to improve the searching abilities of the standard PIO. Stability and parameter instructions of SCPIO are both given. Comparative experiments demonstrate the advantages of the proposed framework.

The remainder of this paper is arranged as follows. Section 2 will introduce the TSS model which includes the cooperation concept, time stamp utility, threat and constraint modeling. Section 3 will first review the standard PIO and then present the principles of SCPIO. The analysis on the stability and complexity of SCPIO are also provided. Section 4 will demonstrate the efficiency of the proposed methods through comparative experiments and the performance profile (PP) technique. The parameter instructions of SCPIO are also involved. Conclusions and future works will be discussed in the final section.

2. Problem formulation

2.1. Concept of multi-UAV cooperative path planning

Multi-UAV cooperative path planning is to generate a few trajectories which satisfy the safety, short range and cooperation requirements for each UAV. The safety illustrates the demand to avoid the obstacles and threat areas in complicated environments. The short range means a less fuel and energy cost. The cooperation is determined by the mission requirements of UAV swarms. For example, the assault mission requests UAVs to take off from the base and arrive at the designated location simultaneously [4]. Meanwhile, in order to keep fluent communications, the distances between UAVs should be limited within a certain range during the flight episodes. Note that the filtered trajectories may not be optimal for single UAV, but with regard to the mission requirements, they should be optimal [4].

A typical scenario of the multi-UAV missions is shown in Fig. 1 where 3 UAVs collaboratively execute the attack task against the specific target. The threat sources involve the terrain obstacles, the enemy's radars and defensive forces. Aircrafts should bypass the threat areas, avoid internal collisions and arrive at the destinations simultaneously to execute the assault. Meanwhile, they should keep inside the desired communication range to maintain the active interactions with neighbors. As shown in Fig. 1, the paths of different UAVs are not necessarily dispersed. The overlap and cross points among different UAV paths illustrate that several adjacent UAVs will fly through the same waypoints at different time or formulate a leader–follower formation when they pass through the narrow passages.

2.2. Multi-UAV path planning with time stamp segmentation

The TSS model evenly divides the common flight time into several time intervals which serve as the common time bases for M aircrafts to assign their trajectories. Assume that all the UAVs share the same takeoff time T_t and arrival time T_a, the desired flight time is $T_a - T_t$ (The way to determine takeoff and arrival time will be discussed later). Given the number of waypoints D, we have the time stamp $t_s = (T_a - T_t)/D$. In the TSS model, two components namely the velocity component $\mathbf{V}_{m,n} = [V_{m,n,x}, V_{m,n,y}]^T$ and the position component $\mathbf{P}_{m,n} = [P_{m,n,x}, P_{m,n,y}]^T$ are designated, where $m = 1 \cdots M$ denotes the index of aircraft and $n = 1 \ldots D$ represents the index of time stamp. In this paper, the height h of UAVs keeps constant and no vertical velocities take effect. The relationship between $\mathbf{V}_{m,n}$ and $\mathbf{P}_{m,n}$ is represented as

$$\mathbf{P}_{m,n} = \mathbf{P}_{m,n-1} + \mathbf{V}_{m,n} \cdot t_s \tag{1}$$

When the time stamp is confirmed, the original waypoints or the position components are determined by the velocity components. Therefore, the velocity components are the subjects to be solved in the TSS model. However, when it comes to the threat and communication costs, the position components should be utilized. As shown in Fig. 2, in order to increase the search probability for the optimal paths, the aforementioned components of mth UAV are defined in the local frame $X_m O_m Y_m$ where the origin takes the takeoff point of mth UAV and the axis X_m points to the direction from the takeoff point $\mathbf{S}_m = [S_{m,x}, S_{m,y}]^T$ to the destination $\mathbf{E}_m = [E_{m,x}, E_{m,y}]^T$ in the 2D global frame $X_g O_g Y_g$. The conversion from the global frame to the local frame can be given as

$$\begin{bmatrix} \tilde{P}_{m,n,x} \\ \tilde{P}_{m,n,y} \end{bmatrix} = \begin{bmatrix} \cos\psi_m & \sin\psi_m \\ -\sin\psi_m & \cos\psi_m \end{bmatrix} \begin{bmatrix} P_{m,n,x} \\ P_{m,n,y} \end{bmatrix} \tag{2}$$

where ψ_m denotes the deflection angle between the two frames and

$$\psi_m = \arctan \frac{E_{m,y} - S_{m,y}}{E_{m,x} - S_{m,x}} \tag{3}$$

The goal of path planning is to find out a best configuration of the velocity and waypoint components for each UAV and to output the best routes when encountering the flight constrains, threat sources and cooperative requirements.

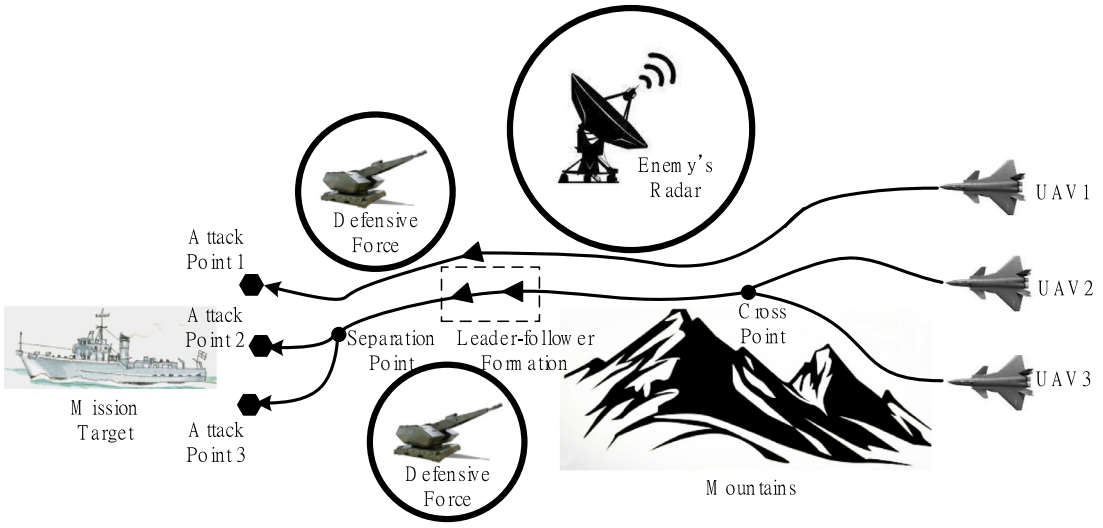


Fig. 1. Path scenario of multi-UAV cooperative assault in complex environment.

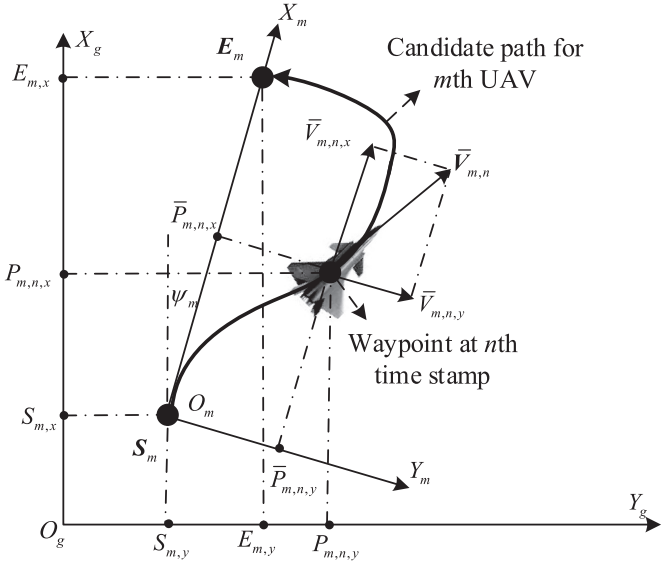


Fig. 2. Coordinates in the TSS path planning model.

2.3. Threat sources and flight constraints

2.3.1. Threat modeling

The threats in this paper mainly include the three sources: mountains, enemy's radars and defensive forces. The dangerous level of each threat could be evaluated by the cost functions [4]. Mountains can be approximately simplified as the trapezoidal cylinders identified by the radius and heights. Given the mth UAV's location $\mathbf{P}_{m,n}$ at nth time stamp in the 2D global frame, the mountain cost of mth UAV at nth time stamp is given as

$$F_{m,n}^{M}=\begin{cases}\dfrac{10^{3}}{\|\mathbf{P}_{m,n}-\mathbf{P}_{M}\|}, & h\le h_{M}\text{ and }\|\mathbf{P}_{m,n}-\mathbf{P}_{M}\|\le\left(1-\dfrac{h}{h_{M}}\right)\cdot R_{Ma}+\dfrac{h}{h_{M}}\cdot R_{Mb}\\ 0, & \text{else}\end{cases}\tag{4}$$

where $\mathbf{P}_M$ denotes the 2D global position of the mountain's surface center and h_M stands for the mountain's height. $R_{Ma}, R_{Mb}\,(R_{Ma}>R_{Mb})$ represent the bottom and top radius of the mountain's surfaces.

The enemy's radar and defensive force share a similar geometric configuration which could be described by a hemisphere with specific surface center and radius whereas the respective dangerous evaluations are different. Given a radar's location $\mathbf{P}_R$ and detection radius R_R, the radar cost of mth UAV at nth time stamp is

$$F_{m,n}^{R}=\begin{cases}\left(\dfrac{\lambda}{\|\mathbf{P}_{m,n}-\mathbf{P}_{R}\|}\right)^{2}, & h\le R_{R}\text{ and }\|\mathbf{P}_{m,n}-\mathbf{P}_{R}\|\le\sqrt{R_{R}^{2}-h^{2}}\\ 0, & \text{else}\end{cases}\tag{5}$$

where λ denotes a dangerous factor indicating the radar's detection level.

The threat cost of defensive forces is described as the following piecewise model in the 3D global frame.

$$F_{m,n}^{F}=\begin{cases}\infty, & \left\|\tilde{\mathbf{P}}_{m,n}-\tilde{\mathbf{P}}_{F}\right\|\le\dfrac{R_{F}}{3}\\ \eta\left(1-\dfrac{\left\|\tilde{\mathbf{P}}_{m,n}-\tilde{\mathbf{P}}_{F}\right\|}{R_{F}}\right), & \dfrac{R_{F}}{3}<\left\|\tilde{\mathbf{P}}_{m,n}-\tilde{\mathbf{P}}_{F}\right\|\le R_{F}\\ 0, & \left\|\tilde{\mathbf{P}}_{m,n}-\tilde{\mathbf{P}}_{F}\right\|>R_{F}\end{cases}\tag{6}$$

where $\tilde{\mathbf{P}}_{m,n}=[\mathbf{P}_{m,n}^{T},h]^{T}$ are the coordinates of mth UAV at nth time stamp in the 3D global frame while $\tilde{\mathbf{P}}_{F}=[\mathbf{P}_{F}^{T},0]^{T}$ stands for 3D location of defensive forces. R_F indicates the maximum firing range and η stands for the defensive coefficient. Then the threat cost of a candidate path of mth UAV could be summarized as

$$F_{m}^{Threat}=L_{c}+\sum_{n=1}^{D}\left(F_{m,n}^{M}+F_{m,n}^{R}+F_{m,n}^{F}\right)\tag{7}$$

where L_c denotes the length of the candidate path. Note that, in order to decline the probability to touch the threat areas, at least three waypoints of a candidate path should be evaluated by the cost function at each time stamp, which include the waypoint at the current time stamp, the waypoint at last time stamp and the midpoint between them.

2.3.2. Aircraft in-flight constraints

In the TSS model, the ranges of the velocity components and the time stamp are determined by the in-flight constraints of aircrafts such as the maximal voyage $L_{\max}$, the minimal step length $s_{\min}$, the range of airspeed $[V_{\min}, V_{\max}]$, and the minimal turning radius $R_{\min}$. Suppose the straight length between the takeoff point

and the destination of mth UAV is l_m, the range of time stamp can be calculated by

$$t_s \in \bigcap_{m=1}^{M} [t_{m,\min}, t_{m,\max}]$$
$$t_{m,\min} = \max\left\{\frac{s_{\min}}{V_{\max}}, \frac{l_m}{V_{\max}D}\right\}$$
$$t_{m,\max} = \min\left\{\frac{L_{\max}}{V_{\min}D}, \frac{l_m}{V_{\min}D}, \frac{T_{\max}}{D}\right\} \tag{8}$$

where $T_{\max}$ stands for the longest endurance time and $\bigcap$ denotes the intersection. Accordingly, we can easily obtain the selection rule for the common arrival time

$$T_a \in D \cdot \bigcap_{m=1}^{M} [t_{m,\min}, t_{m,\max}] + T_t \tag{9}$$

and the maximum allowed yaw angle

$$\psi_{\max} = \arcsin\frac{V_{\min}t_s}{2R_{\min}} \tag{10}$$

According to Ref. [27], the unidirectional optimization of candidate solutions could simplify the solving complexity. Similar to this concept, we arrange the constant lateral elements of velocity component in local frame at any time stamp, that is $\bar{V}_{m,n,x} = l_m/(D \cdot t_s)$. Then use the SIOs to search for the optimal configurations of the longitudinal elements $\bar{V}_{m,n,y}$ to obtain the desired paths. Considering Eq. (10), we can get the range of the longitudinal elements, that is $\bar{V}_{m,n,y} \in [-\bar{V}_{m,n,x}\tan\psi_{\max}, \bar{V}_{m,n,x}\tan\psi_{\max}]$, $m = 1\dots M$, $n = 1\dots D$.

2.3.3. Coordination costs

As illustrated in Section 2.1, the cooperative requirements consist of the common endurance time, collision avoidance and communication limitations. The endurance time is spontaneously fulfilled in the TSS model. Define the safety distance d_c and the maximum communication range R_c among multiple UAVs. Suppose another aircraft $m_1 \in \{1\dots M,\ m_1 \neq m\}$ within the neighborhood of mth UAV, the cooperative requirement is that, at nth time stamp, the space between the two corresponding UAVs should accord with $d_c \le \|\mathbf{P}_{m,n} - \mathbf{P}_{m_1,n}\| \le R_c$. Accordingly, the coordination cost can be described as

$$F_m^{Coor} = \begin{cases} 0, & d_c \le \|\mathbf{P}_{m,n} - \mathbf{P}_{m_1,n}\| \le R_c \\ \infty, & \text{otherwise} \end{cases}, \quad n = 1\dots D \tag{11}$$

Hence, the overall cost of a candidate path of mth UAV could be summarized as $F_m^{Total} = \omega_1 F_m^{Threat} + \omega_2 F_m^{Coor}$, where ω_1, ω_2 represent the impact weights for the threat sources and coordination limits respectively. The rest in this paper is to find out the optimal configuration of the longitudinal elements $\bar{V}_{m,n,y}$ for each UAV that minimize the overall cost.

3. Methods

3.1. Review to the standard PIO

PIO is a novel SIO that has been successfully applied on many scientific and engineering problems. Two operators inspired by the pigeons' navigation tools formulate the framework of standard PIO [22]. One is the map and compass operator which is adjusted during the first part of optimization; the other is the landmark operator which is referenced when the population approaches to the global optima. In both operators, the candidate solutions for the benchmark problems are considered as the pigeons' positions and the best position indicate the optimal solution accordingly. During the episode of the map and compass operator, the solutions are evolved by

$$\mathbf{v}_i(t) = \mathbf{v}_i(t-1)e^{-Rt} + rand \cdot (\mathbf{x}_g(t-1) - \mathbf{x}_i(t-1))$$
$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \mathbf{v}_i(t),\ i = 1\cdots N \tag{12}$$

where $\mathbf{x}_i(t)$ represents the ith solution at tth iteration which also is called the position of ith pigeon in PIO. $\mathbf{v}_i(t)$ is the so-called velocity of ith pigeon which reflects the evolution slope of ith solution. N is the population size and $\mathbf{x}_g(t)$ denotes the global best solution or the global leader at tth iteration in PIO. *rand* is a random number subject to the uniform distribution between 0 and 1. R is a positive factor that reveals the influence of the map and compass utility.

At the stage of landmark operator, half pigeons with inferior costs are assumed to follow the superior flocks. That is to say, the swarm size of superiors will be reduced by half per iteration $N_P(t) = N_P(t-1)/2$ with the initial size N until an individual is reserved. The landmark operator utilizes the following evolution strategy to update each pigeon

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + rand \cdot (\mathbf{x}_c(t-1) - \mathbf{x}_i(t-1))$$
$$\mathbf{x}_c(t) = \frac{\sum_{j\in S_P(t)} \mathbf{x}_j(t) f\left(\mathbf{x}_j(t)\right)}{N_P(t) \sum_{j\in S_P(t)} f\left(\mathbf{x}_j(t)\right)} \tag{13}$$

where $\mathbf{x}_c(t)$ is the weighted population center which serves as the searching reference during the landmark phase. $S_P(t)$ stands for the set of superiors and $f(\bullet)$ represents a quality criteria that evaluates the solutions' importance. Consider the cost function of benchmark problem $f_{\text{cost}}(\bullet)$, we have

$$f(x) = \begin{cases} \dfrac{1}{f_{\text{cost}}(x) + \varepsilon}, & \text{for minimum optimization} \\ f_{\text{cost}}(x), & \text{for maximum optimization} \end{cases} \tag{14}$$

where ε is a near-zero constant to ensure the validation of Eq. (14). Remark that, in the end of both operators, a greedy strategy in Eq. (15) should be performed to improve the solutions' quality.

$$\mathbf{x}_i(t) = \begin{cases} \mathbf{x}_i(t), & f(\mathbf{x}_i(t)) > f(\mathbf{x}_i(t-1)) \\ \mathbf{x}_i(t-1), & f(\mathbf{x}_i(t)) \le f(\mathbf{x}_i(t-1)) \end{cases} \tag{15}$$

3.2. Social-class PIO algorithm

Although the standard PIO has been proved superior in some practices, there are still some shortcomings such as lack of diversity and prematurity [23]. In this work, inspired by the inherent social-class properties of in-flight pigeons, a novel SCPIO is developed to address the two aforementioned problems and improve the performances of standard PIO.

The hierarchies during the pigeons' flight have aroused much attention on the investigation of collective motion [28–30] and engineering applications [31,32]. Nagy et al. [28,29] discovered an implicit hierarchical network in pigeon flocks through the statistical analysis on the actually gathered positions. Zhang et al. [30] extended this work and put forward that a switching mechanism may occur between the social hierarchies and the egalitarian modes during the flight of pigeons. Enlightened by these discoveries, Luo and Duan [31] developed a distributed control framework for multi-UAV flight. In this architecture, several interactive functions between different UAVs were designed to realize the motion consensus. Qiu and Duan [32] also investigated the flight control of the multi-UAV close formations by abstracting and utilizing the communication principles of pigeons' social hierarchies. A schematic diagram of in-flight hierarchy in a pigeon flock is shown in Fig. 3. This social network is composed of four layers of individuals. The arrows indicate the connections or communications between the two individuals in different layers. The pigeon in the

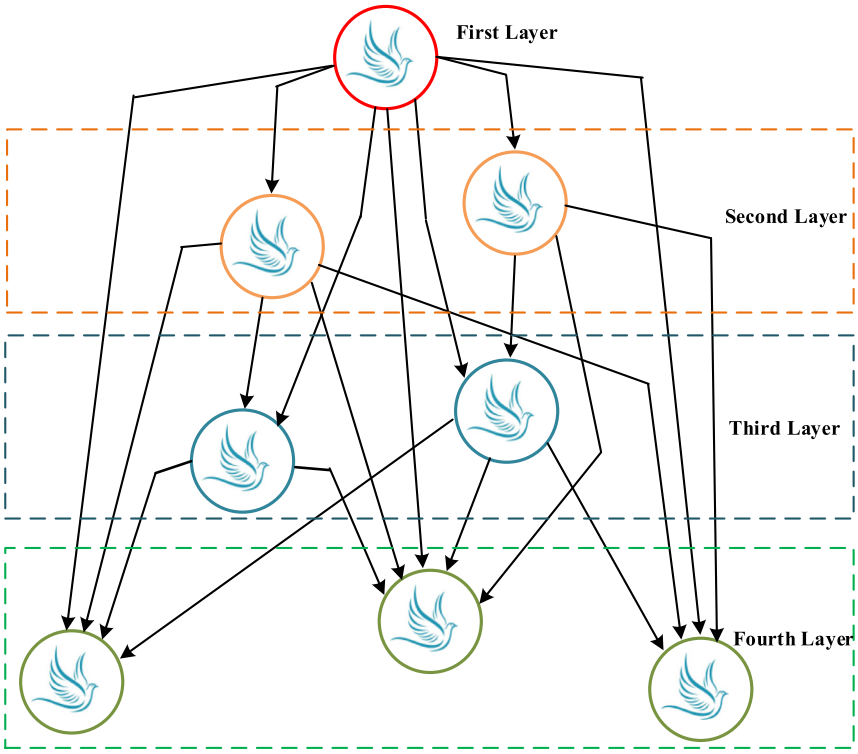


Fig. 3. Schematics of the social hierarchy in pigeon flocks.

top layer is considered as the leader of the whole flock since its information is disseminated across the network. Note that the information flow is unidirectional. That is, the pigeons in lower classes only receive information form higher layers but can't transmit their status.

Based on this concept, we allocate the whole pigeons into $L(L \geq 2)$ social layers or classes in SCPIO and prescribe that the first class is the top-layer group where the group members are available to share their information across the population. Accordingly, the Lth class serves as the bottom hierarchy where the members can't transmit their information but only receive the positions of higher class groups. Similarly, the information flow in SCPIO is also unidirectional. The solutions with better cost values shall enter the higher level classes in sequence. For example, if the swarm size of first class is N_1, we allocate the best N_1 solutions into the top layer during the current iteration, and others likewise. The number of individuals in each social class is optional. Let $\mathbf{x}_g^k (k = 1, \cdots L)$ be the best solution of the kth social class and assume that the ith solution belongs to the kth social class, then the revised map and compass operator in SCPIO is presented as

$$\begin{aligned}\mathbf{v}_i(t) &= \mathbf{v}_i(t-1)e^{-R_k} + rand \cdot (\mathbf{x}_g(t-1) - \mathbf{x}_i(t-1)) \\ &\quad + \sum_{j=1}^{k} rand \cdot \left(\mathbf{x}_g^j(t-1) - \mathbf{x}_i(t-1)\right) \\ \mathbf{x}_i(t) &= \mathbf{x}_i(t-1) + \mathbf{v}_i(t), \ i = 1 \cdots N \end{aligned} \tag{16}$$

where R_k represents the map and compass factor of the kth class. From the Eq. (16), it's apparent that the individuals from lower classes could receive and exploit the superior information from higher classes. This principle could disseminate the superiors' perfect experiences and in turn enhances the explorations of the whole population. Note that, compared with the standard PIO, the velocity weight in SCPIO is turned into a constant e^{-R_k} to keep the solutions' dependence on the individual experiences to support the maintenance of population diversity when the index of iteration t is large enough.

The half reduction of superior set in the original landmark operator is essential to the convergence behavior since it stimulates the population to explore around the estimated near-best positions. This mechanism increases the probability to approximate to the global optima. In SCPIO, in order to keep the avoidance of premature, the size of superiors decreases faster, that is $N_P(t) = N_P(t-1)/L$. Besides, a new evolutionary scheme is used in the revised landmark operator such that

$$\mathbf{x}_i(t) = \begin{cases} \mathbf{x}_i(t-1) + rand \cdot (\mathbf{x}_c(t-1) - \mathbf{x}_i(t-1)), & i \in S_P(t) \\ \mathbf{x}_c(t-1) + rand \cdot (\mathbf{x}_c(t-1) - \mathbf{x}_i(t-1)), & i \notin S_P(t) \end{cases} \tag{17}$$

$$\mathbf{x}_c(t) = \frac{\sum_{j \in S_P(t)} \mathbf{x}_j(t) f\left(\mathbf{x}_j(t)\right)}{N_P(t) \sum_{j \in S_P(t)} f\left(\mathbf{x}_j(t)\right)} \tag{17}$$

That is to say, the inferiors will shift into the neighborhood of a near-best center to encourage the explicit search for global optima. Just like the process of the standard PIO, a greedy strategy is also required in the end of SCPIO. As a result, the detail procedure of SCPIO is summarized by the following steps.

Step 1: Initialize the necessary parameters of SCPIO, such as the search space dimension D, iteration thresholds for two operators $N_{\text{cmax1}}, N_{\text{cmax2}} (N_{\text{cmax1}} < N_{\text{cmax2}})$, the population size N and the number of social classes L. Besides, the number of individuals N_k and the positive factor R_k of kth class should also be designated.

Step 2: Randomly initialize the pigeons' positions and velocities by the uniform distribution.

Step 3: Sort the population by comparing their cost values and rank them into the corresponding classes in sequence. Obtain the global leader $\mathbf{x}_g$ and the best member $\mathbf{x}_g^k$ of each social class.

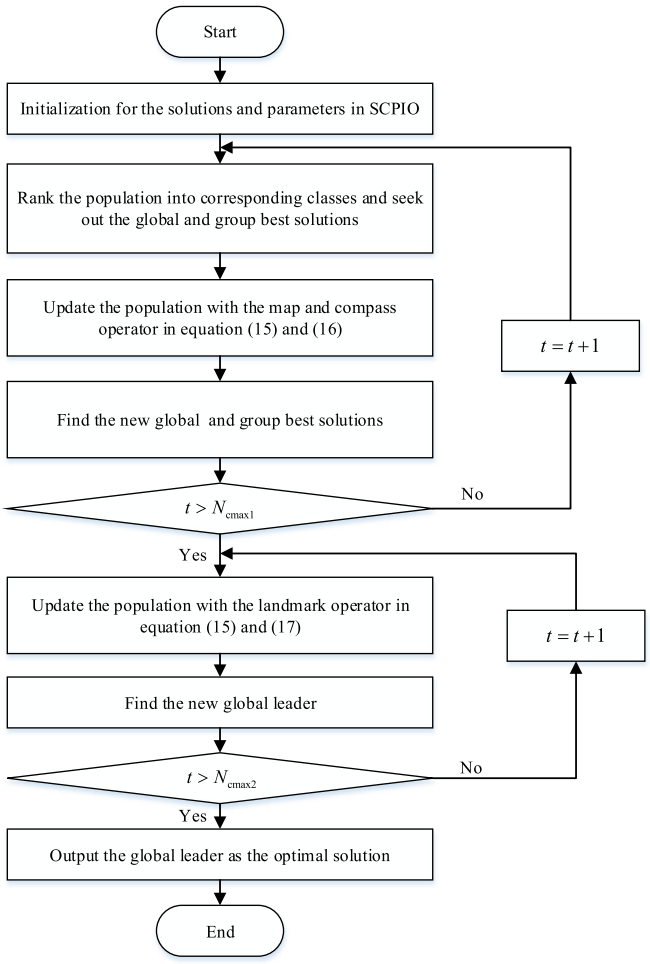


Fig. 4. Implementation flowchart of SCPIO.

Step 4: Implement the map and compass operator. The populations are evolved by Eq. (16) and the greedy strategy in Eq. (15). Then compare the cost values and screen out the new leader as well as the best members of each class.

Step 5: If the index of iteration $t > N_{cmax1}$, forward to the landmark operator. Otherwise, go back to Step 3.

Step 6: Implement the landmark operator. Evaluate the cost values and arrange the $N_P(t)$ superior individuals into the superior set $S_P(t)$. The size of $S_P(t)$ satisfies $N_P(t) = N_P(t-1)/L$. Then update the populations by Eq. (17) and the greedy strategy in Eq. (15). Screen out the new leader and the best members of each social class.

Step 7: If the index of iteration $t > N_{cmax2}$, output the global leader as the optimal solution. Otherwise, go back to Step 6.

A flow chart is shown in Fig. 4 to give an intelligible overview to the aforementioned steps.

3.3. Stability analysis on SCPIO

The convergence proof of SCPIO is a challenging issue due to its stochastic structure [33]. Leboucher et al. [34] presented that two main conditions account for the local convergence of a stochastic optimizer, namely *algorithm condition* and *convergence condition*. The *algorithm condition* ensures the reachability to the optimal set of the candidate algorithm while the *convergence condition* checks whether the algorithm is stable within the optimal region. Using the properties of Markov chain, Zhang and Duan [27] have proved that the solutions of the standard PIO could enter into the global optimal set in probability 1 (see Theorem 1 in [27]). This illustrates the justification of the *algorithm condition*. The *algorithm condition* of SCPIO could be deduced via the similar approaches due to its straight derivation from the standard PIO. However the *convergence conditions* or stabilities of SCPIO or PIO are seldom investigated. Fortunately, the stability of PSO has been widely investigated, which is helpful for the homogenous methods. For example, in order to manipulate the random variables in PSO, the following 1-order stability definition [35] is proposed.

Definition 1. A random SIO is 1-order stable if the Eq. (18) is satisfied and the corresponding parameter region is 1-order stable region.

$$\lim_{t\to\infty} E(\mathbf{x}(t)) = \mathbf{y} \tag{18}$$

where $E(\mathbf{x})$ is the expectation of the candidate solution $\mathbf{x}$ and $\mathbf{y}$ denotes a constant vector.

Indeed the 1-order definition is not available to fully describe the dynamic behaviors of the algorithm as it can't provide the variation trend of candidate solutions. Hence the following 2-order definition [36] is presented next, which is based on the convergence of the variances.

Definition 2. A random SIO is 2-order stable if the Eq. (19) is satisfied and the corresponding parameter region is 2-order stable region.

$$\lim_{t\to\infty} D(\mathbf{x}(t)) = 0 \tag{19}$$

where $D(\mathbf{x})$ is the variance of the candidate solution $\mathbf{x}$.

Theorem 1. *If a SIO satisfies the 2-order stability, it also satisfies the 1-order stability.*

Proof. Since $D(\mathbf{x}(t)) = E(\mathbf{x}(t) - E(\mathbf{x}(t)))^2$, the Eq. (19) implies that the expectation exists for almost all $t > 0$. On the other hand, according to [37], the solutions will converge to a constant vector $\mathbf{y}$ with probability 1 such that

$$P\left\{\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{y}\right\} = 1$$

By computing the limit of the expectation, simply we have the form of Eq. (18), that is, the 1-order stability is established.

Note that the candidate solution in both Eqs. (18) and (19) stands for the entire population. That is, we demand all the candidate solutions remain stagnation in the local convergence status. However, this constraint is too strict and not realistic for SIO because the randomness of evolution process cannot guarantee the uniformity of the population [36]. Recently, a new 2-order stability definition which is based on the weak stagnation condition has been proposed by Liu [33]. It assumes that the global leader remains constant during the convergence state while other non-dominant solutions are allowed to improve. The weak stagnation 2-order stability is denoted as follows.

Definition 3. If the population's leader stays at the dominant position $\mathbf{x}_g(T_c)$ during the iterations between $t = T_c$ and $t = T_c + T (T \geq 3)$, then the optimizer is said to be stable if

$$\lim_{T\to\infty} D(\mathbf{x}_g(T_c + T)) = 0 \tag{20}$$

where T_c denotes the index of iteration at which the global leader first enters the dominant position.

Remark 1. The weak stagnation 2-order stability implies that the *convergence condition* of a random SIO is fully dependent on the population's leader in spite of the algorithm's space topology [33]. Therefore, the stagnation constraint is weak and realistic compared

with Definition 2 in most cases. In this paper, the weak stagnation 2-order stability will be adopted to obtain the *convergence condition* of SCPIO.

Corollary 1. *If a SIO satisfies the weak stagnation 2-order stability, it also satisfies the 1-order stability, that is*

$$\lim_{T\to\infty} E(\mathbf{x}_g(T_c+T)) = \mathbf{x}_g(T_c) \tag{21}$$

In terms of SCPIO, the weak stagnation 2-order stability is also fully determined by the global leader. From the implementation flow of SCPIO, it's reasonable that SCPIO is stable as long as the two involved operators are both stable. First take the map and compass operator into consideration. From Eq. (16), we have

$$\begin{aligned}\mathbf{x}_i(t+1) &= \alpha_1(t)\cdot\mathbf{x}_i(t) + \alpha_2(t)\cdot\mathbf{x}_i(t-1) + \beta_0(t)\cdot\mathbf{x}_g(t)\\ &\quad + \sum_{j=1}^{k}\beta_k(t)\cdot\mathbf{x}_g^j(t)\end{aligned} \tag{22}$$

where the coefficients are presented as

$$\begin{aligned}\alpha_1(t) &= 1 + e^{-R_k} - \sum_{j=0}^{k} q_j(t)\\ \alpha_2(t) &= -e^{-R_k}\\ \beta_j(t) &= q_j(t), \quad j = 0,\cdots,k\end{aligned}$$

and $q_j(t)$ stands for a random number subject to the uniform distribution between 0 and 1 at the tth iteration. Suppose that the pigeons conducting the map and compass operator is in the weak stagnation state during $t\in[T_c, T_c+T]$. Because the global leader surely belongs to the first class, it's obvious that $\mathbf{x}_g(t) = \mathbf{x}_g^1(t) = \mathbf{x}_g(T_c),\ t\in[T_c, T_c+T]$. Thus for the global leader, the Eq. (22) could be simplified into the following stochastic difference equation (SDE).

$$\mathbf{x}_g(t+1) = \alpha_1(t)\cdot\mathbf{x}_g(t) + \alpha_2(t)\cdot\mathbf{x}_g(t-1) + \alpha_3(t)\cdot\mathbf{x}_g(T_c) \tag{23}$$

where $\alpha_3(t) = \beta_0(t) + \beta_1(t)$ and $\alpha_1(t) + \alpha_2(t) + \alpha_3(t) = 1$. To solve the Eq. (23), the recursion approach is utilized.

Theorem 2. *Suppose that the population is in the weak stagnation state of the map and compass operator during $t\in[T_c, T_c+T]$, the global leader stays at the dominant position $\mathbf{x}_g(T_c)$, then the solutions of SDE (23) submit to the following hypersurface*

$$\mathbf{x}_g(T_c+t) = \mathbf{x}_g(T_c) + G(t)\cdot(\mathbf{x}_g(T_c+1) - \mathbf{x}_g(T_c)),\quad t\in[0,T] \tag{24}$$

where $G(t)$ satisfies the recursion

$$G(t+1) = \alpha_1(T_c+t)G(t) - e^{-R_1}G(t-1),\ G(0) = 0,\ G(1) = 1 \tag{25}$$

Proof. For $t=0$ and $t=1$, the result is self-evident. Using the method of induction, assume Theorem 2 holds for $t=k-1$ and $t=k$. From the Eq. (24), it yields

$$\begin{aligned}\mathbf{x}_g(T_c+k-1) &= \mathbf{x}_g(T_c) + G(k-1)\cdot(\mathbf{x}_g(T_c+1) - \mathbf{x}_g(T_c))\\ \mathbf{x}_g(T_c+k) &= \mathbf{x}_g(T_c) + G(k)\cdot(\mathbf{x}_g(T_c+1) - \mathbf{x}_g(T_c))\end{aligned} \tag{26}$$

Considering the Eq. (23) at $t = T_c+k$, we have

$$\begin{aligned}\mathbf{x}_g(T_c+k+1) &= \alpha_1(T_c+k)\cdot\mathbf{x}_g(T_c+k) - e^{-R_1}\mathbf{x}_g(T_c+k-1)\\ &\quad + \left(1 + e^{-R_1} - \alpha_1(T_c+k)\right)\cdot\mathbf{x}_g(T_c)\end{aligned} \tag{27}$$

Combining the Eqs. (26) and (27), it yields

$$\begin{aligned}\mathbf{x}_g(T_c+k+1) &= \mathbf{x}_g(T_c) + G(k+1)\cdot(\mathbf{x}_g(T_c+1) - \mathbf{x}_g(T_c))\\ G(k+1) &= \alpha_1(T_c+k)G(k) - e^{-R_1}G(k-1)\end{aligned} \tag{28}$$

Hence Theorem 2 establishes when $t=k+1$, and the proof is finished.

Remark 2. From the SDE (24), the global leader's dynamics is mainly dominated by $G(t)$, while $G(t)$ is mainly affected by the random variable $\alpha_1(t)$. In other words, the solution trajectories of Eq. (24) will surround the dominant position $\mathbf{x}_g(T_c)$ and fluctuate around it with a magnitude relative to $\alpha_1(t)$. Besides, from Definition 3 and Corollary 1, it is apparent that if the map and compass operator converges to stability, then we have $\lim_{t\to\infty} E(G(t)) = 0$.

Theorem 3. *If the global leader stays at the dominant position $\mathbf{x}_g(T_c)$ during the weak stagnation state $t\in[T_c,+\infty]$, then the map and compass operator is stable if and only if*

$$\lim_{t\to\infty} E\left(G^2(t)\right) = 0 \tag{29}$$

Proof. According to Theorem 2, during the weak stagnation period, the global leader's dynamics submits to the Eq. (24). Due to $\mathbf{x}_g(T_c)$ and $\mathbf{x}_g(T_c+1)$ are both constant, the variance of the dominant position has the form of

$$D(\mathbf{x}_g(T_c+t)) = D(G(t))[\mathbf{x}_g(T_c+1) - \mathbf{x}_g(T_c)]^2 \tag{30}$$

According to Definition 3, the stability condition is equivalent to $\lim_{t\to\infty} D(G(t)) = 0$. Moreover, since $D(G(t)) = E(G^2(t)) - E(G(t))^2$, we have $\lim_{t\to\infty} E(G^2(t)) = 0$.

Theorem 4. *The stability of the map and compass operator is determined by the following sufficient and necessary condition, that is*

$$R_1 > 0.1203 \tag{31}$$

The proof of this theorem is given in the Appendix via solving a difference equation about the expectation $E(G^2(t))$. As for the landmark operator, assume that the global leader stays at another dominant position $\mathbf{x}_g(\tilde{T}_c)$ forever, where $\tilde{T}_c$ denotes the initial index of iteration in the weak stagnation stage of the landmark operator. Due to the fast attenuation of $N_P(t)$, it's sufficient to believe that $N_P(t) = 1$ when the landmark operator falls into the weak stagnation stage. This also indicates that the exclusive member in $S_P(t)$ is the global leader and $\mathbf{x}_c(t) = \mathbf{x}_g(t)$ $(t\in[\tilde{T}_c,+\infty])$. Then according to Eq. (17), the dominant position of the global leader will be unchangeable during the weak stagnation stage. Thus the landmark operator is stable spontaneously. In summary, the *convergence condition* of the SCPIO is established if and only if Eq. (31) is satisfied.

3.4. Complexity analysis of SCPIO

The computational complexity of SCPIO is easily obtained from its mathematical formulas. Given the population size N, the iteration thresholds for two respective operators N_{cmax1},N_{cmax2}, the number of social classes L and the computational load of the cost function N_{cost}. For the map and compass operator, from Eq. (16), the evolution cost at each iteration is $O(NL)$. Furthermore, the ranking process for each social class is also required before individual update. If the polling strategy is utilized, the ranking cost per iteration is $O(NL + NN_{\text{cost}})$. Therefore, the complexity of the map and compass operator is $O(N_{\text{cmax1}}\cdot(NL + NN_{\text{cost}}))$.

On the other hand, in the landmark operator, the size of superior set will be reduced by L per iteration. According to Eq. (17), the evolution cost at each iteration is $O(N + \log_L N)$. If the polling method is considered during the ranking the superior cluster $S_P(t)$, the ranking cost at each iteration is $O(N\log_L N + NN_{\text{cost}})$. Then the computational cost of the landmark operator is obtained $O(N_{\text{cmax2}}(N\log_L N + NN_{\text{cost}}))$. Thus the total complexity of the SCPIO is summarized as

$$O(N_{\text{cmax2}}N(\log_L N + N_{\text{cost}}) + N_{\text{cmax1}}N(L + N_{\text{cost}}))$$

Normally, since the number of social classes L is far less than N, the complexity of the SCPIO is approximately equal to the standard PIO [27]. Thus it's reasonable that the SCPIO is not worse than

the standard PIO in the respect of computation costs as it doesn't increase the computational complexity substantially.

4. Numerical experiments and analysis

4.1. Performance profile and comparative tests on benchmark functions

As an efficient benchmarking approach for large-scale deterministic optimization algorithms, the performance profile (PP) technique [38,39] has been widely used in the research of mathematical programming. It adopts the cumulative distribution function (CDF) to distinguish the capabilities of candidate algorithms. Recently, PP has been extended to support the stochastic optimization algorithms by taking the confidence intervals of test samples as the raw data of PP [39], which also provides the dynamic implications of test samples by comparisons of confidence intervals.

According to the PP technique, a comprehensiveness matrix $\boldsymbol{H}$ with size $n_f \times n_p \times n_s$ should be given first, where n_f, n_p, n_s denote the number of iterations used in each algorithm, the number of benchmark functions and the number of candidate algorithms, respectively. Define $t_{p,s}$ the index of iteration needed for algorithm s on the benchmark function p to arrive at the following condition:

$$f_{\text{cost}}(x_0) - f_{\text{cost}}(x) \geq (1-\tau)(f_{\text{cost}}(x_0) - f_L) \tag{32}$$

where x_0 is the start position, τ is a small positive factor that represents the convergence tolerance of the situation above and f_L denotes the best value obtained within all the algorithms during the current test. The smaller the tolerance factor, the higher the requirement to convergence accuracy. If we cannot find any position satisfying the condition above, $t_{p,s} = \infty$. Then we give the definition of performance profiles for algorithm s using the following CDF:

$$\rho_s(\gamma) = \frac{1}{n_p}\text{Int}\left(\left\{p \in \text{P} \middle| \frac{t_{p,s}}{\min\{t_{p,s}, s \in \text{S}\}} \leq \gamma\right\}\right) \tag{33}$$

where P, S denote the problem set and the algorithm set, respectively. The function $\text{Int}(\cdot)$ returns the volume of a set and $\gamma \geq 1$ denotes the performance ratio which reveals the coverage area of solvable problems of the algorithm s. In particular, $\rho_s(1)$ represents the proportion of problems that s can solve with best performances in S. Apparently, a higher performance profile ρ_s illustrates that the candidate algorithm s could solve a larger scope of problems within the convergence limitation in Eq. (32) and thus behaves superiorly. To evaluate the stochastic optimizers using PP technique, the duplicate test and statistical tools such as confidence intervals are both considered. Given a sample of benchmarking test $\{X_1, \cdots X_{n_r}\}$ with capacity of n_r, the means and variance of the test samples are

$$\bar{X} = \frac{1}{n_r}\sum_{j=1}^{n_r} X_j, \quad S^2 = \frac{1}{n_r - 1}\sum_{j=1}^{n_r}\left(X_j - \bar{X}\right)^2 \tag{34}$$

Hence we have the corresponding confidence interval $[\bar{X} - \kappa S/\sqrt{n_r}, \bar{X} + \kappa S/\sqrt{n_r}]$, where κ stands for the confidence level of a standard normal distribution. In this paper, we set $\kappa = 2$ to obtain a 95% level [39] during the benchmarking process and $n_r = 50$ times of duplicate tests are implemented for each algorithm on each problem. Obtain the sample means and variance by Eq. (34) and compose the means and variance matrix $\mathbf{H}_{\text{means}}$, $\mathbf{H}_{\text{var}}$. Firstly, arrange the PP technique to evaluate the means of candidate algorithms. In this paper, we mainly focus on the minimum problems. Therefore, choose several algorithms with lowest means into the elitist group. Then, combine the means and variance matrix to formulate the confidence upper and lower bound matrix $\mathbf{H}_{\text{upper}}$, $\mathbf{H}_{\text{lower}}$. Secondly, arrange the PP technique to do the comparison between the upper bounds of the elites and the lower bounds of the other algorithms. If the elites' confidence upper bounds still surpass the inferiors' lower bounds, then the elite algorithms surely have stronger behaviors both in the convergence accuracy and in the dynamic characteristics. Otherwise, it's suggested that no significant difference is found between the elites and the inferiors.

Table 1
Configurations for candidate algorithms in PP evaluation.

Standard PIO and SCPIO parameters							
N_{cmax1}	40	N_{cmax2}	60	R	0.3	L	4
$R_{1,2,3,4}$	3	$N_{1,2,3,4}$	10,5,5,10				
PSO Parameters							
N_{cmax}	60	ω	0.42	c_1	1.55	c_2	1.55
DE Parameters							
N_{cmax}	60	P_{cr}	0.7	F	0.5		

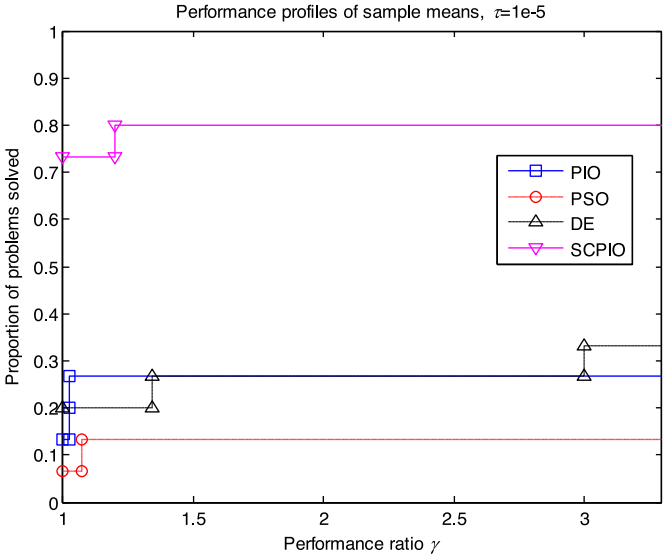


Fig. 5. Performance profiles of sample means for candidate algorithms.

In this section, we use the PP technique to evaluate the capabilities of SCPIO. Three homogenous optimizers are selected, including the standard PIO, PSO and DE algorithm. The initial configurations of the candidate optimizers are shown in Table 1. To keep fairness, we set the population size $N = 30$ within all the candidate algorithms and $n_p = 15$ classical benchmark functions are executed simultaneously with the same dimension $D = 10$. The detail information of the selected benchmark functions are listed in Table 2. Conduct the duplicate tests and collect the sample. Then, calculate the means and variance matrix by Eq. (34) for each algorithm. Set $\tau = 10^{-5}$, and compute $t_{p,s}$ on the condition of Eq. (32). Collect the performance profiles for each algorithm with different performance ratios by Eq. (33). The PPs of sample means of all the algorithms are shown in Fig. 5.

Obviously the means profiles of SCPIO are better than other optimizers from Fig. 5. Specifically, SCPIO is available to solve almost 80% benchmark functions subject to the convergence situation in Eq. (32) while the best of rest optimizers only occupies less than half. Subsequently, we compare the confidence upper bound matrix of SCPIO against the lower bound matrices of the other algorithms to implement a further evaluation by PP technique (shown in Fig. 6). As a result, the upper bound of SCPIO still surpasses the lower bounds of others. In other words, although some random effects or fluctuations surely exist in SIOs, the worst scores of SCPIO are still competitive. Hence, it's believed that SCPIO outperforms

Table 2
Benchmark functions in PP evaluation.

Function name	Variable bounds	Optimal solutions	Optimal value
Ackley	$[-32.768, 32.768]$	$\mathbf{x}^* = (0, \ldots, 0)$	$f(\mathbf{x}^*) = 0$
Dixon–Price	$[-10, 10]$	$\mathbf{x}^* = \left\{x_i, i = 1, \ldots, D \middle\vert x_i = 2^{\wedge}\left(-\frac{2^i-2}{2^i}\right)\right\}$	$f(\mathbf{x}^*) = 0$
Griewank	$[-600, 600]$	$\mathbf{x}^* = (0, \ldots, 0)$	$f(\mathbf{x}^*) = 0$
Levy	$[-10, 10]$	$\mathbf{x}^* = (1, \ldots, 1)$	$f(\mathbf{x}^*) = 0$
Michalewicz	$[0, \pi]$	$\mathbf{x}^* = (2.20, 1.57, \ldots)$	$f(\mathbf{x}^*) = -9.66015$
Perm($\beta = 10$)	$[-D, D]$	$\mathbf{x}^* = (1, \frac{1}{2}, \ldots, \frac{1}{D})$	$f(\mathbf{x}^*) = 0$
Rastrigin	$[-5.12, 5.12]$	$\mathbf{x}^* = (0, \ldots, 0)$	$f(\mathbf{x}^*) = 0$
Rosenbrock	$[-10, 10]$	$\mathbf{x}^* = (1, \ldots, 1)$	$f(\mathbf{x}^*) = 0$
Hyper-ellipsoid	$[-65, 65]$	$\mathbf{x}^* = (0, \ldots, 0)$	$f(\mathbf{x}^*) = 0$
Schwefel	$[-500, 500]$	$\mathbf{x}^* = (420.9687, \ldots, 420.9687)$	$f(\mathbf{x}^*) = 0$
Sphere	$[-5.12, 5.12]$	$\mathbf{x}^* = (0, \ldots, 0)$	$f(\mathbf{x}^*) = 0$
Styblinski–Tang	$[-5, 5]$	$\mathbf{x}^* = (-2.903534, \ldots, -2.903534)$	$f(\mathbf{x}^*) = -39.16599D$
Sum of Powers	$[-1, 1]$	$\mathbf{x}^* = (0, \ldots, 0)$	$f(\mathbf{x}^*) = 0$
Sum Squares	$[-10, 10]$	$\mathbf{x}^* = (0, \ldots, 0)$	$f(\mathbf{x}^*) = 0$
Zakharov	$[-10, 10]$	$\mathbf{x}^* = (0, \ldots, 0)$	$f(\mathbf{x}^*) = 0$

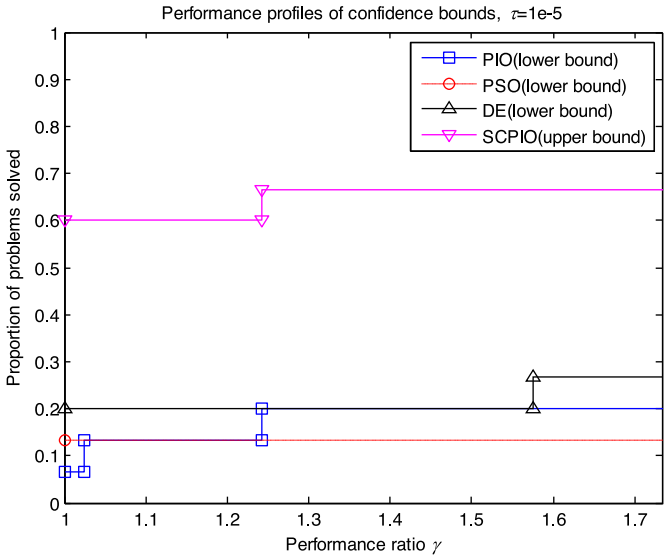


Fig. 6. Performance profiles of confidence bounds for candidate algorithms.

Table 3
Common parameter setting in PP evaluation.

Common parameters							
N	30	D	10	N_{cmax1}	40	N_{cmax2}	60
L	3	N_1	10	N_2	10	N_3	10
n_f	60	n_p	15	n_s	8	n_r	50

the other three optimizers in the respect of both accuracy and dynamic behaviors.

4.2. Parameter instructions of SCPIO based on performance profiles

4.2.1. Investigation on the map and compass factor

In order to configure the parameters of SCPIO via PP technique, we also conduct $n_r = 50$ times duplicate tests with the same group of benchmark functions in Section 4.1 for each specified SCPIO. In this section, $n_s = 8$ SCPIOs with different R_k (listed in Fig. 7) are tested where the factors in different social classes should be equal. Other common parameters are given in Table 3. Through PP computations, the means profiles for all algorithms are shown in Fig. 7.

From the results, it seems that $R_k = 3$ may have an advantage over other parameters. However, by comparison of confidence bounds (shown in Fig. 8), there seems no clear disparity. Through comprehensive analysis on the results in both figures, it's supposed

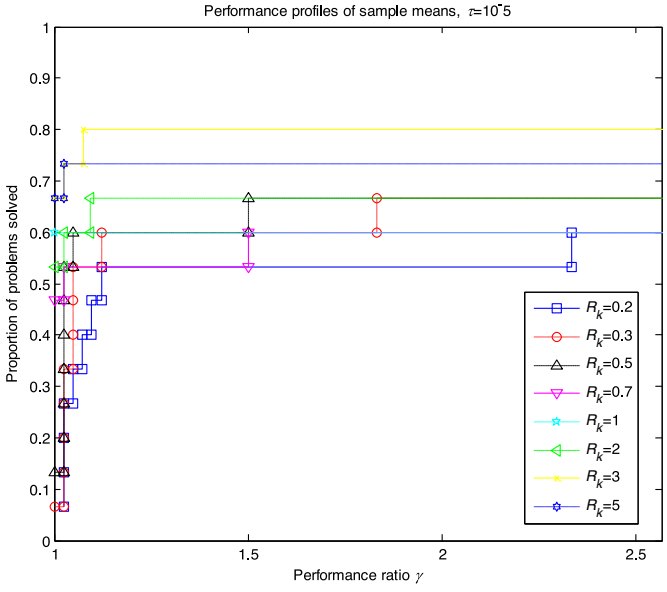


Fig. 7. Means profiles for SCPIOs with different R_k.

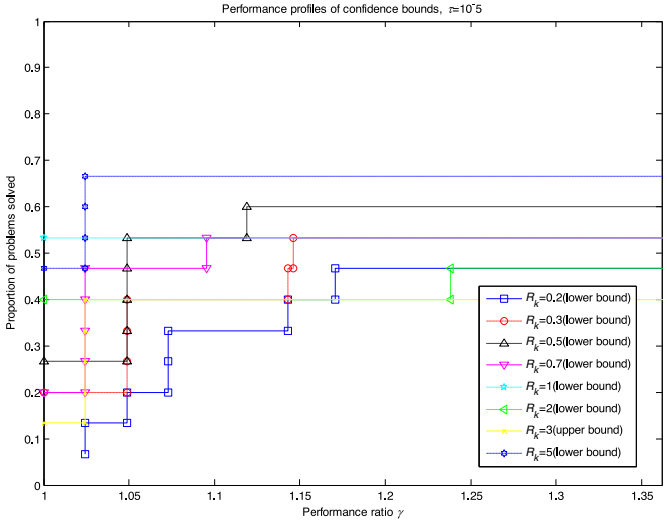
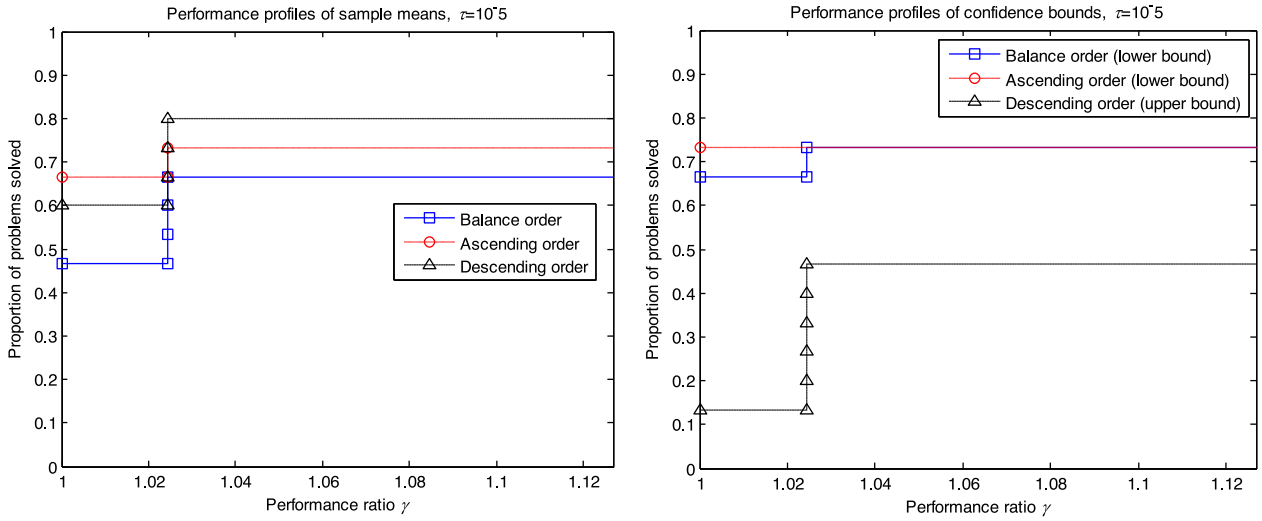
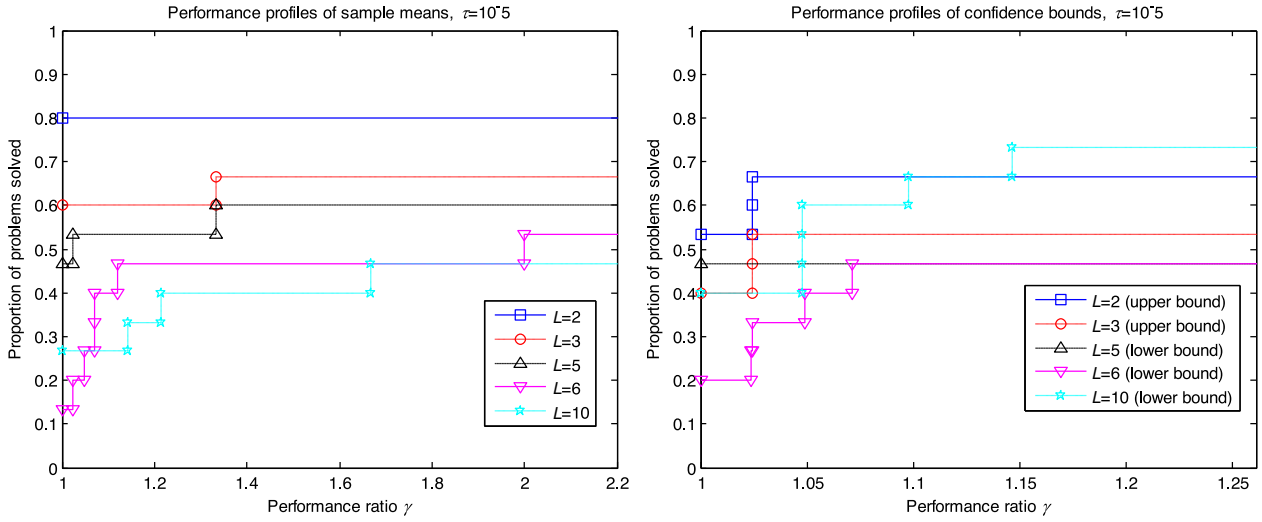


Fig. 8. Confidence bound profiles for SCPIOs with different R_k.



(a) Profiles of sample means (b) Profiles of confidence bounds

Fig. 9. Performance profiles for SCPIOs with different R_k sequences.



(a) Profiles of sample means (b) Profiles of confidence bounds

Fig. 10. Performance profiles for SCPIOs with different number of social classes.

that $R_k > 1$ may be a better choice if high-precision requirements are expected.

On the other hand, the sequences of R_k among classes may also influence the efficiency of SCPIO. Hence, another experiment is carried out where three sequences of R_k are involved, including the balance order ($R_{1,2,3} = 2, 2, 2$), the ascending order ($R_{1,2,3} = 1, 2, 3$) and the descending order ($R_{1,2,3} = 3, 2, 1$). The comparable profiles are shown in Fig. 9. However, from the results, no significant diversities could be found among different sequences. Thus it's supposed that any form of sequence is available for SCPIO.

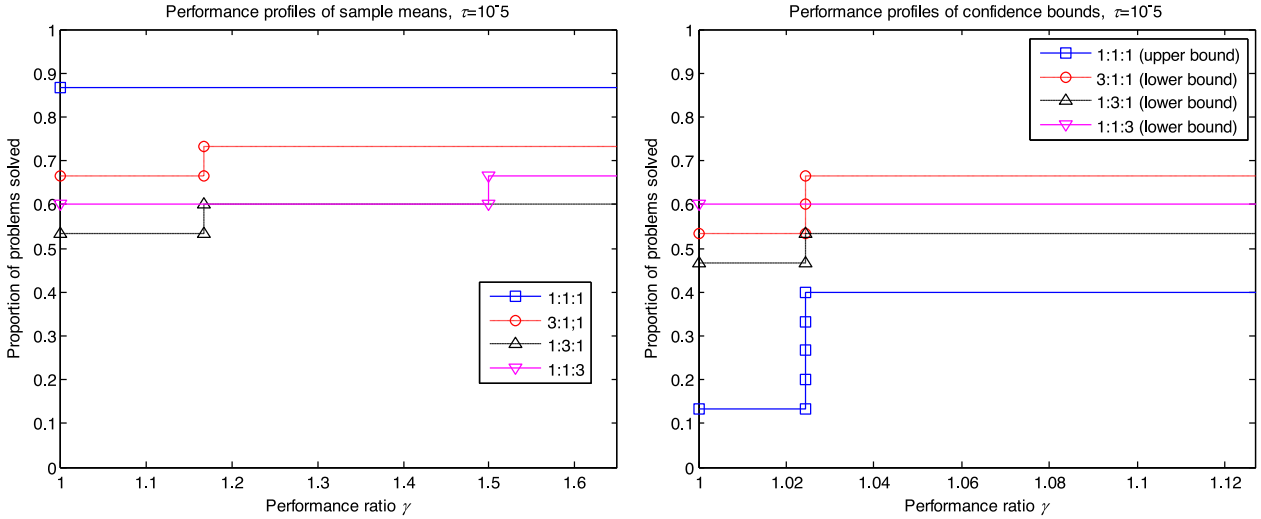
4.2.2. Investigation on social classes

The number of social classes is an essential parameter for SCPIO because it stands for the specific population topology in search spaces. In this section, $n_s = 5$ SCPIOs with different number of classes ($L = 2, 3, 5, 6, 10$) are investigated by the PP technique. Similar with the former evaluations, the common parameters in Table 3 are adopted except for the number of social classes L. For each algorithm, set $R_k = 2$ and evenly arrange the population into the classes in accordance with the individuals' costs. The obtained comparative profiles are shown in Fig. 10. According to the results, large number of social classes has no contributions to the improvement of SCPIO, and perhaps $L = 2 \sim 4$ may be enough for most situations.

Additionally, we check the impact of class sizes on the structural optimization of SCPIO and another experiment with different proportions of class members is implemented. Here, we also take $R_k = 2$ and $L = 3$ for each candidate algorithm with the rest configurations consistent with Table 3. $n_s = 4$ SCPIOs with different class scale proportions (shown in Fig. 11) are tested with the PP technique. The consequences in Fig. 11 illustrate that, although the proportion 1:1:1 performs competitively at the means level, it can't promise stability during the duplicate tests. However, if our goal is to tentatively obtain a high-precision solution, the balance proportion may be preferable.

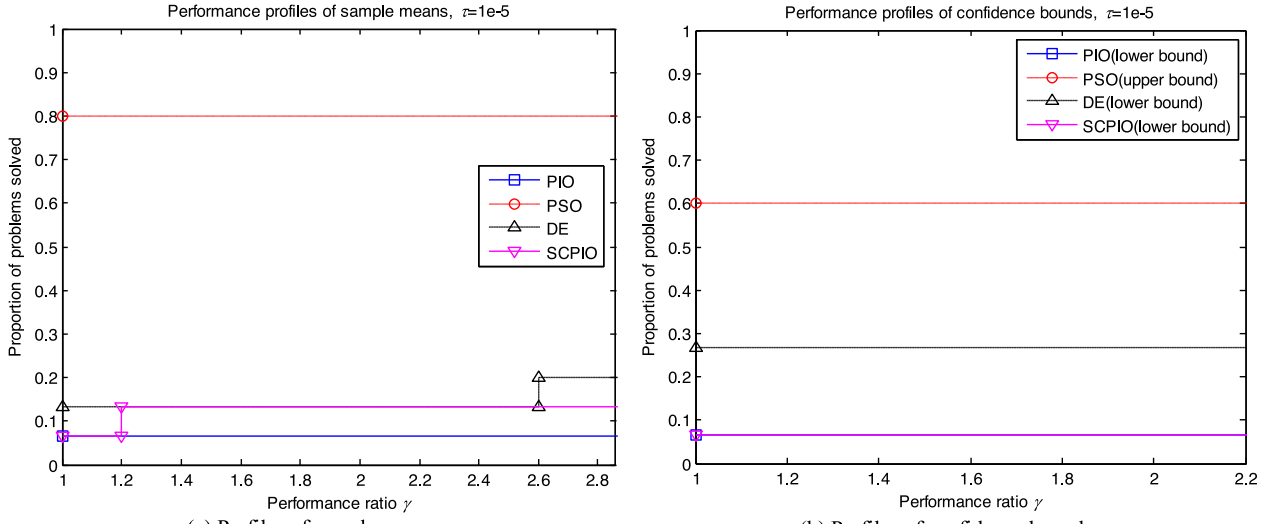
4.2.3. Investigation on the operators' behaviors

Durations of both operators are important cues for the performances of SCPIO. Unfortunately, through multiple tests with PP technique, no special pattern was discovered in this field. However, the investigation on single operator implies some new con-



(a) Profiles of sample means (b) Profiles of confidence bounds

Fig. 11. Performance profiles for SCPIOs with different class scale proportions.



(a) Profiles of sample means (b) Profiles of confidence bounds

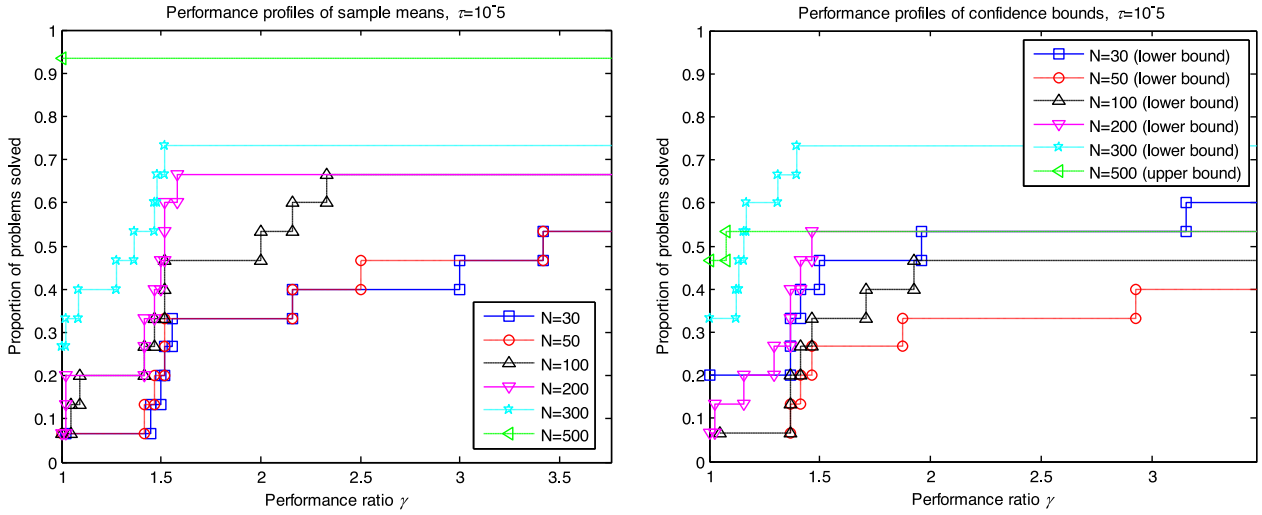
Fig. 12. Performance profiles for algorithms during the map and compass duration.

cepts. We extract the sample in Section 4.1 within the first N_{cmax1} iterations and obtain the corresponding performance profiles shown in Fig. 12.

According to the results, during the map and compass operator, PSO is competitive than PIO and SCPIO. This illustrates that the population topology of PSO is better than the map and compass operator. Nevertheless, during the landmark operator, both PIO and SCPIO improve remarkably. This phenomenon implies that, the map and compass operator pursues the crude exploration in the beginning, while the landmark operator keeps the avoidance of premature in both PIO and SCPIO and provides the satisfied solutions at the final stage. The possible reasons may be that, after the crude search of the map and compass operator in the whole search span, the search area will be evolved and compressed to several possible neighborhoods around the global optima. Then, during the landmark operator, the explicit search around the near-best neighborhoods will successively accelerate the search for the global optima. This principle will remarkably improve the solutions' accuracy without stagnation into the local solutions. Thus in the end of iteration, the behaviors of PIO and SCPIO will be enhanced significantly. The advantages of SCPIO over the standard PIO lies in the fact that the constant factor and the social-class framework of the revised map and compass operator could increase the population diversity and extend the search span through dependencies on self-best experiences and multi-point guidance. Besides, the revised landmark operator further supports the avoidance of premature at the later search period. However, since the coarse search in the first period, the convergence speed of PIO and SCPIO is not better than PSO and DE. That is, the map and compass operator still needs growth to improve the search efficiency.

4.2.4. Investigation on the population sizes

For the evolutionary algorithms, the population size is a key parameter that determines the convergence level. Intuitively, a larger population size should generate more accurate solutions. But this also significantly increases the complexity of the algorithm. In [40], the authors indicated that the best population size of evolutionary algorithms depends on the prototypes of benchmark problems. In this section, we test 6 SCPIOs ($N=30$, 50, 100, 200, 300, 500) and use the PP technique to check the relationships between the population size and the convergence level. Similarly, we set $R_k = 2$ and use the common parameters in Table 3 to conduct the dupli-



(a) Profiles of sample means (b) Profiles of confidence bounds

Fig. 13. Performance profiles for SCPIOs with different population sizes.

cate tests. The corresponding results are shown in Fig. 13. It can be seen that the computation accuracy surely promotes with the increase of the population size. Nevertheless, since the computation cost also increases remarkably, we always expect a moderate population size in most situations. That is, the selection of population size of SCPIO or other evolutionary algorithms still depends on the user's demands and the search scope of benchmark problems. Consider the results in Fig. 13(a), the algorithms with moderate sizes $N = 100 \sim 300$ could still solve almost 70% problems within the precision requirement. Hence, in general, a moderate population size (such as $N = 100$) is always expected for SCPIO since it can keep balance between the precision and the computation cost within the problem limitations.

4.3. Simulations on SCPIO and TSS model

In this section, the path planning system based on the TSS model and SCPIO algorithm will be introduced. In this framework, SCPIO serves as the solver of the TSS model for a single UAV. As is mentioned in Section 2.2, an individual solution in SCPIO of mth UAV is composed of the velocity and position components $\mathbf{V}_{m,n}$, $\mathbf{P}_{m,n}$ $(n = 1 \ldots D)$. Since the path planning task is to search for the optimal elements $\bar{V}_{m,n,y}$ for each UAV that minimize the overall cost, the dimension of SCPIO should be consistent with the number of time intervals or waypoints D. The time stamp is determined by the flight constraints in Section 2.3.2 and the cost function of mth UAV employs the total threat and coordination cost F_m^{Total} in Section 2.3.3. Note that, in order to implement the computation of multi-UAV coordination cost, the reliable communications should be guaranteed and the information from other UAVs should be updated per iteration. Then, with the evolution of $\mathbf{V}_{m,n}$ and $\mathbf{P}_{m,n}$ by SCPIO, it's easy to find the optimal paths satisfying the safety and cooperation requirements for each UAV. The procedure of the integrated system can be summarized by the steps below.

Step 1: Initialize the environmental settings of multi-UAV cooperative path planning, such as the number of UAVs, the takeoff and destination points, threat settings and locations, flight constraints of UAVs etc.

Table 4
Environmental settings for the integrated simulation.

Layout of the mission points					
	UAV 1	UAV 2	UAV 3	UAV 4	
Takeoff point (km)	(5,5,5)	(5,10,5)	(10,5,5)	(10,10,5)	
Destination (km)	(85,85,5)	(85,90,5)	(90,85,5)	(90,90,5)	
Layout of mountain threats					
	Mountain 1	Mountain 2	Mountain 3		
$\mathbf{P}_M$(km)	(45,25)	(67,80)	(30,70)		
h_M(km)	7	8	8		
R_{Ma}, R_{Mb}(km)	18, 6	15, 5	17, 4		
Layout of defensive forces and radars					
	Force 1	Force 2	Force 3	Force 4	Radar
$\mathbf{P}_F$, $\mathbf{P}_R$(km)	(20,30)	(72,42)	(75,58)	(25,45)	(56,54)
R_F, R_R(km)	7	8	8	7	12
Flight and coordination constraints					
$s_{\min}$(km)	2	$L_{\max}$(km)	250	$V_{\max}$(km/h)	600
$V_{\min}$(km/h)	300	$R_{\min}$(km)	2	$T_{\max}$(min)	60
T_t, T_a(min)	0,18	d_c(km)	2	R_c(km)	20
Obtained common variables for each UAV					
t_s(min)	0.45	$\psi_{\max}$(rad)	0.557		
Obtained specific variables for each UAV					
	UAV 1	UAV 2	UAV 3	UAV 4	
$\bar{V}_{m,n,x}$(km/h)	400	400	400	400	
$\bar{V}_{m,n,y}$(km/h)	[−250, 250]	[−250, 250]	[−250, 250]	[−250, 250]	

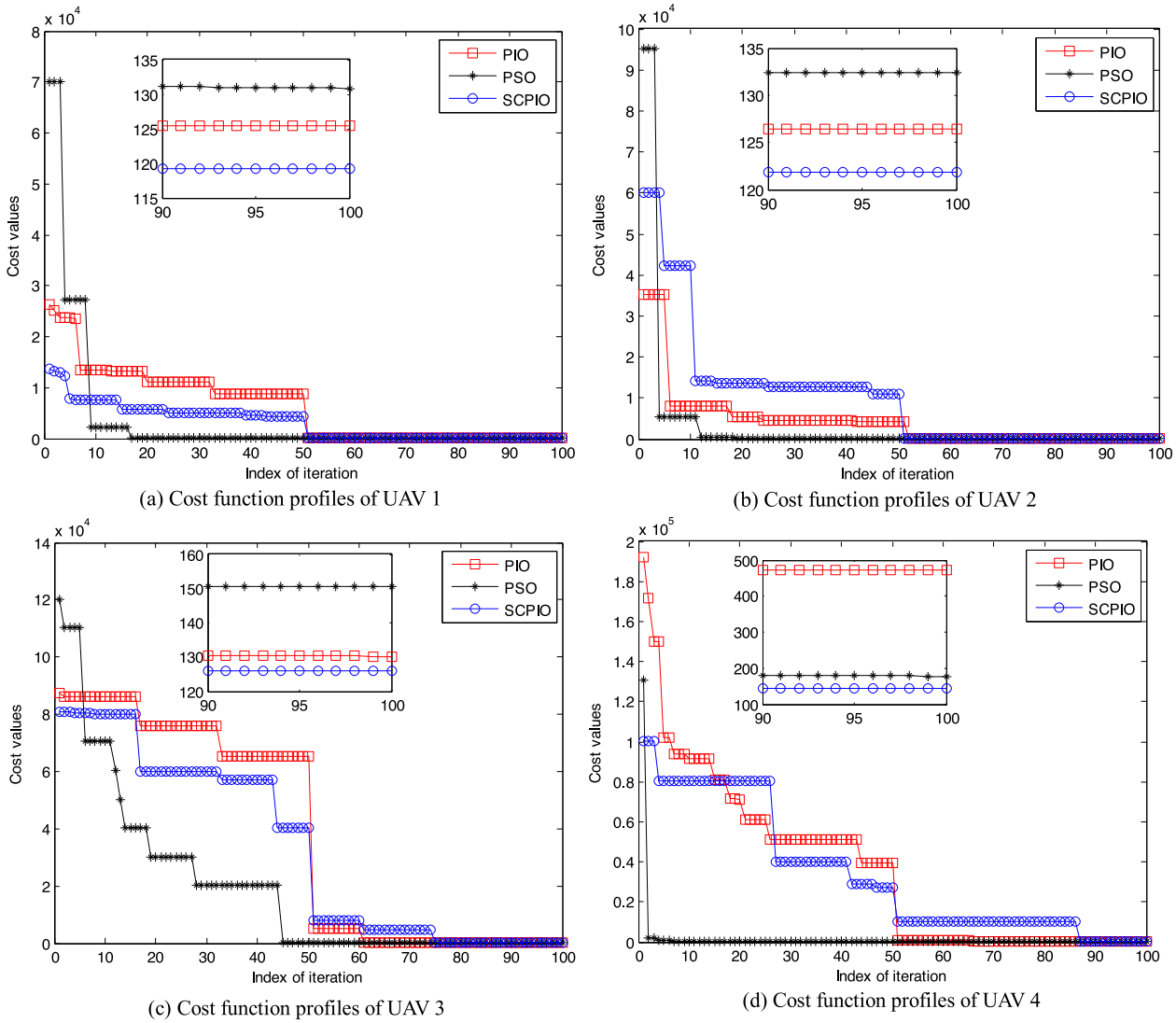


Fig. 14. Comparative SIO results of path planning for respective UAV.

Table 5
Parameter configurations of SIOs (SCPIO, PIO and PSO).

PIO and SCPIO parameters							
N	100	D	40	N_{cmax1}	50	N_{cmax2}	100
R	0.3	L	3	$R_{1,2,3}$	3	$N_{1,2,3}$	30, 40, 30
PSO Parameters							
N	100	D	40	N_{cmax}	100	ω	0.42
c_1	1.55	c_2	1.55				

Table 6
Best SIO costs of path planning for respective UAV.

	UAV 1	UAV 2	UAV 3	UAV 4
SCPIO	119.171	121.871	126.148	144.970
PIO	125.386	126.378	130.182	470.763
PSO	130.833	132.411	150.686	173.981

Step 2: Given the takeoff time T_t and the dimension D, according to the constraints in Eqs. (8) and (9), select the arrival time T_a and the time stamp t_s. Build the local frame $X_mO_mY_m$ for mth UAV and calculate the constant lateral element $\bar{V}_{m,n,x}$ and the variation range of the longitudinal element $\bar{V}_{m,n,y}$.

Step 3: For mth UAV, initialize the configurations of SCPIO. The individual solutions are composed of the velocity and position components $\mathbf{V}_{m,n}$, $\mathbf{P}_{m,n}$ $(n = 1 \ldots D)$. The elements to be optimized are $\bar{V}_{m,n,y} (n = 1 \ldots D)$ while others could be determined through Eq. (1).

Step 4: For mth UAV per iteration, select a candidate solution and calculate the threat cost F_m^{Threat} by Eqs. (4)–(7). Simultaneously, receive the best waypoints at the current iteration from other UAVs and calculate the coordination cost F_m^{Coor} by Eq. (11). Then sum up the costs and obtain the total cost F_m^{Total} for this solution.

Step 5: For mth UAV per iteration, implement the SCPIO to update the candidate solution. If all the candidate solutions have been evolved and the index of iteration is maximal, finish the searching process and output the optimal components. Otherwise, go back to Step 4.

The position component is the optimal path for each UAV, while the velocity component serves as the auxiliary information for the UAV's navigation. Since we can update the population within an iteration of SCPIO for a single UAV, it's easy to conduct the optimal search for other UAVs within the same iteration. That is, the aircrafts could implement the respective SCPIO in parallel, which of course declines the computation cost. Another benefit of the

current system is the additional navigation sources. The time stamps account for the motion time bases of multi-UAV cooperation. As is mentioned in Introduction, the motion time bases are the common references for multi-UAV coordination which could simplify the computation of coordination cost and provide the time bases for the navigation of aircrafts. Meanwhile, the velocity components generated by the optimal solutions represent the desired velocities during the episode $(t_0, t_0 + t_s)$ where t_0 stands for the start time of the episode. The desired velocities are significant sources for flight control because they provide the explicit control references (such as the velocity references at each time stamp) during the corresponding episode $(t_0, t_0 + t_s)$. However, the limitation of dimension causes the crudity of the output paths. Therefore, after the optimal search of SCPIO, the path smoothing technique should be utilized in addition to eliminate the sharp corners and polish the paths.

To verify the feasibility of the integrated system, a comparative experiment has been conducted, where three kinds of SIOs (SCPIO,

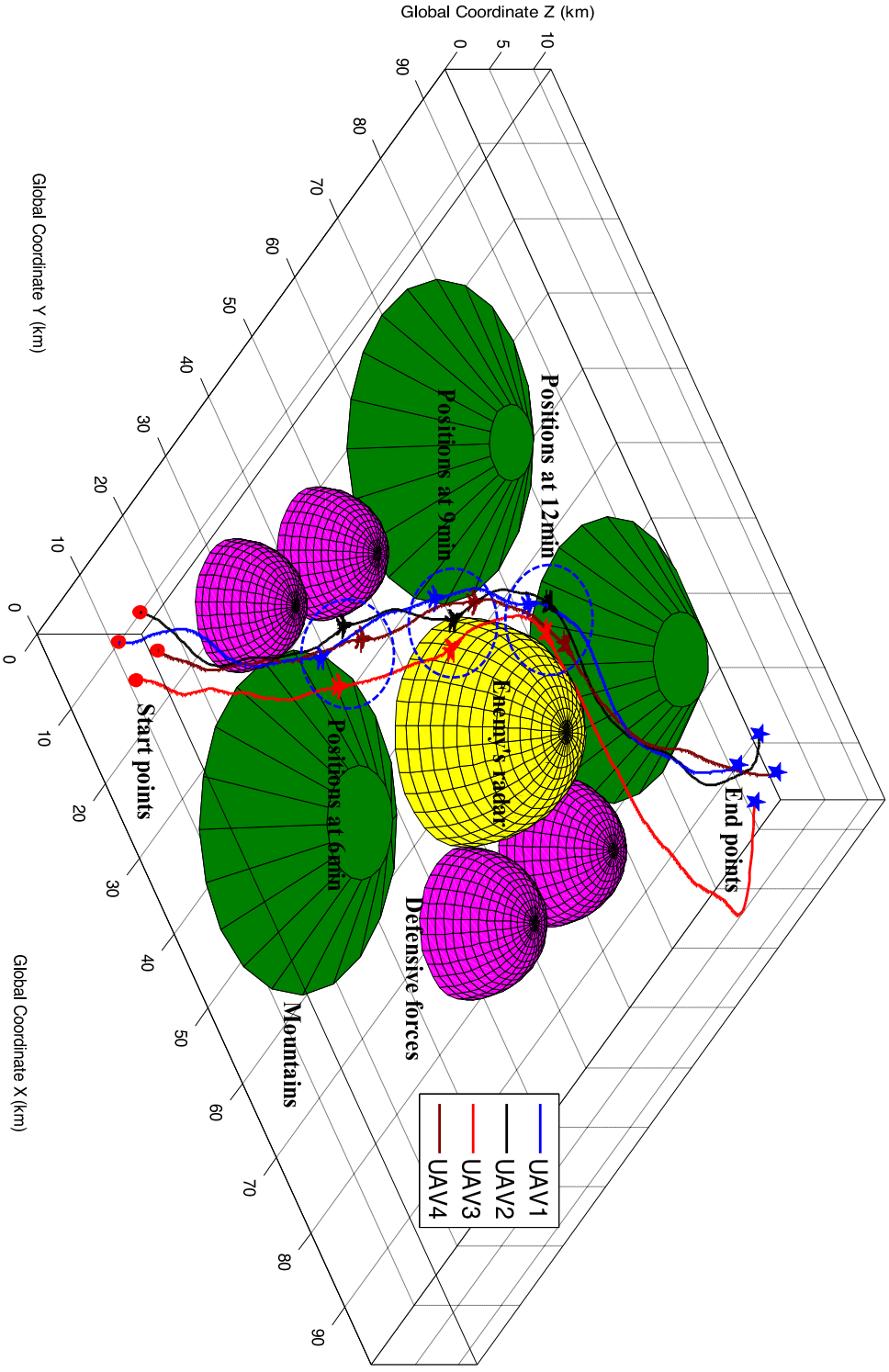


Fig. 15. Optimal trajectories for UAVs by SCPIO within virtual battleground.

PIO and PSO) are taken as the optimization approaches for the TSS model. Four UAVs are considered and the basic environmental settings are given in Table 4. The parameter configurations of SIOs are shown in Table 5. Note that the infinite scores of the threat and coordination costs in Eqs. (6) and (11) are substituted by 10^4 in this simulation. The comparative results of three SIOs are presented in Fig. 14 and Table 6, and the optimal paths of SCPIO within 3D battleground are displayed in Fig. 15.

According to Table 6, it is apparent that SCPIO converges to the lowest costs for all the candidate UAVs. However, from the profiles in Fig. 14, the convergence speed of PIO and SCPIO is not better than PSO in the first evolution period (map and compass operator). The main reason is that the map and compass operator is responsible for the crude exploration and can't provide an accurate estimation of the global optima. As is mentioned in Section 4.2.3, the population topology of PSO surpasses the map and compass operators of PIO and SCPIO and thus supports the fast convergence accordingly. But at the final phase, the solutions of SCPIO are competitive. According to Fig. 14, there exist a clear descending trend in both profiles of PIO and SCPIO after the map and compass operator. This consequence strongly supports the standpoints in Section 4.2.3 that the landmark operator is responsible for the explicit search and the avoidance of premature. Furthermore, in most cases of Fig. 14, the revised map and compass operator of SCPIO outperforms that of the standard PIO, which also demonstrates the consequences in Section 4.2.3 that the constant factor and social-class framework could improve the search ability of SCPIO by increasing the population diversity.

Finally, in view of Fig. 15, there surely exist a few overlaps and cross points among different UAVs' paths on the TSS model. However, this doesn't mean the collisions among UAVs. As shown in Fig. 15, we extract the UAV positions at three key moments within the total flight time 18 min (simulation time at 6 min, 9 min and 12 min). At each moment, four UAVs could keep dense formations and pass by the terrains and threats without any collision. This phenomenon supports the scenario in Section 2.1 and extends the results in [4] which suppose that the dispersal among multi-UAV paths is necessary. In fact, if the trajectories are excessively dispersed, the growth of length costs is obviously inevitable. In addition, it's also difficult to satisfy the coordination constraints if the communication distances among different UAVs are too far away. Therefore, in many cases, the crossovers of paths are important, especially for the TSS model.

5. Conclusions

This paper addresses the challenging issue of multi-UAV cooperative path planning. Two essential techniques have been proposed: First is the TSS model and second is the SCPIO algorithm. The TSS model provides the time stamps as the common time references for waypoint programming, which not only simplifies the computation of multi-UAV coordination cost, but also generates the desired velocities as the auxiliary information for navigation and control. SCPIO is developed as the solver for the TSS model which utilizes a social-class strategy to improve the original operators of the standard PIO. Two outstanding characteristics of SCPIO are discovered. On the one hand, the map and compass operator is enhanced by a predefined multi-class structure which accordingly raises the population diversity. On the other hand, the landmark operator is responsible for the explicit search and premature avoidance at the later period. Both features contribute to the convergence accuracy of SCPIO. Through duplicate tests and PP technique, the performances and parameter instructions of SCPIO have been demonstrated. Finally, an integrated experiment based on TSS and SCPIO has been conducted within a 3D complex environment to verify the proposed methods.

From the experimental results, a new concept about the multi-UAV cooperative path planning has been verified that the overlaps and cross points don't mean the path contradictions of different UAVs. In most cases, the path overlaps even account for the decrease of path costs and the warranty of reliable communications. Through analysis on performance profiles, it's supposed that SCPIO may occupy promising behaviors in some other fields. Thus, in the future, it's sincerely suggested that SCPIO could be utilized to solve more challenging issues in scientific areas and modification approaches are welcomed to refine the original disadvantages.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under grants #61333004, #61425008 and #91648205, Aeronautical Foundation of China under grant #2015ZA51013 and Academic Excellence Foundation of BUAA for Ph.D. Students in 2018.

Appendix

Proof of Theorem 4: According to Theorem 3, in order to manifest the stability of the map and compass operator, the expectation $E(G^2(t))$ should be calculated first. Denote the following parameters

$$\mu = E(\alpha_1), \quad \sigma^2 = D(\alpha_1) \tag{a1}$$

For the global leader during the weak stagnation state, consider the independence between q_0 and q_1, we have

$$\mu = e^{-R_1}, \quad \sigma^2 = \frac{1}{6} \tag{a2}$$

Denote $u = e^{-R_1}$, from Eq. (25), it yields

$$\begin{aligned} G^2(t+1) &= \alpha_1^2(T_c+t)G^2(t) + u^2G^2(t-1) \\ &\quad -2u\alpha_1(T_c+t)G(t)G(t-1) \\ G(t+1)G(t) &= \alpha_1(T_c+t)G^2(t) - uG(t)G(t-1) \end{aligned} \tag{a3}$$

Then, according to the independence, we have

$$\begin{aligned} E\left(G^2(t+1)\right) &= \left(\mu^2+\sigma^2\right)E\left(G^2(t)\right) + u^2E\left(G^2(t-1)\right) \\ &\quad -2u\mu E(G(t)G(t-1)) \\ E(G(t+1)G(t)) &= \mu E\left(G^2(t)\right) - uE(G(t)G(t-1)) \end{aligned} \tag{a4}$$

Using the elimination, the following difference equation is obtained

$$\begin{aligned} &E\left(G^2(t+2)\right) + \left(u-\mu^2-\sigma^2\right)E\left(G^2(t+1)\right) \\ &+\left(u\mu^2-u\sigma^2-u^2\right)E\left(G^2(t)\right) - u^3E\left(G^2(t-1)\right) = 0 \end{aligned} \tag{a5}$$

which satisfies $E(G^2(0)) = 0$, $E(G^2(1)) = 1$, $E(G^2(2)) = \mu^2+\sigma^2$. Hence the characteristic equation of Eq. (a5) is

$$\Phi(r) = r^3 + \left(u-\mu^2-\sigma^2\right)r^2 + \left(u\mu^2-u\sigma^2-u^2\right)r - u^3 = 0 \tag{a6}$$

The stability of the map and compass operator requires the sufficient and necessary condition $\lim_{t\to\infty} E(G^2(t)) = 0$ which also implies that the norm of all the three solutions of Eq. (a6) should be less than 1. Since $R_1 > 0$, $u \in (0, 1)$, then $\Phi(u) = -2u^2\sigma^2 < 0$, $\Phi(+\infty) = +\infty$. The stability demands that, there at least exists one solution at the real axis (0, 1), which also implies $\Phi(1) > 0$. Accordingly, we have

$$(1-u)\mu^2 + (1+u)\sigma^2 < (1+u)^2(1-u) \tag{a7}$$

On the other hand, the norm of two other solutions should also be less than 1. Assume that the real root of Eq. (a6) is $r_1 \in (u, 1)$,

the characteristic Eq. (a6) can be rewritten as

$$\Phi(r) = (r - r_1)\left[r^2 + \left(u - \mu^2 - \sigma^2 + r_1\right)r + \frac{u^3}{r_1}\right] = 0 \tag{a8}$$

By solving the equation above, the other two roots are obtained

$$r_{2,3} = \frac{\left(\mu^2 + \sigma^2 - u - r_1\right) \pm \sqrt{\Delta}}{2} \tag{a9}$$

where $\Delta = (\mu^2 + \sigma^2 - u - r_1)^2 - 4u^3/r_1$. Therefore the norm of roots requires $\max\{|r_2|, |r_3|\} < 1$ which also yields

$$u + r_1 - 1 - \frac{u^3}{r_1} < \mu^2 + \sigma^2 < u + r_1 + 1 + \frac{u^3}{r_1} \tag{a10}$$

From Eq. (a7), we get

$$\mu^2 + \sigma^2 < \mu^2 + \frac{1+u}{1-u}\sigma^2 < (1+u)^2 \tag{a11}$$

Meanwhile, since $r_1 \in (0, 1)$, it's easily obtained that

$$u + r_1 + 1 + \frac{u^3}{r_1} \geq (1+u)^2 \tag{a12}$$

Hence the right side inequality of Eq. (a10) is not necessary. Besides, from the characteristic Eq. (a6), we have

$$\begin{aligned}\Phi(-1) &= -1 + \left(u - \mu^2 - \sigma^2\right) - \left(u\mu^2 - u\sigma^2 - u^2\right) - u^3 \\ &= -(1-u)^2(1+u) - (1+u)\mu^2 - (1-u)\sigma^2 < 0\end{aligned} \tag{a13}$$

On the other hand, according to Eq. (a8),

$$\Phi(-1) = (-1 - r_1)\left[1 - \left(u - \mu^2 - \sigma^2 + r_1\right) + \frac{u^3}{r_1}\right] < 0 \tag{a14}$$

which implies that the left side inequality of Eq. (a10) is also not necessary. Hence, in summary, Eq. (a7) serves as the sufficient and necessary condition for the stability of the map and compass operator. Substituting the real values of Eq. (a2) into Eq. (a7), simply we get the following equivalent condition

$$R_1 > -\ln\frac{5 + \sqrt{265}}{24} \approx 0.1203 \tag{a15}$$

References

[1] L. Huang, H. Qu, P. Ji, X. Liu, et al., A novel coordinated path planning method using k-degree smoothing for multi-UAVs, Appl, Soft. Comput. 48 (2016) 182–192.

[2] M. Shanmugavel, A. Tsourdos, B. White, R. Zbikowski, Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs, Control Eng. Pract. 18 (9) (2010) 1084–1092.

[3] F. Ahmed, K. Deb, Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms, Soft Comput. 17 (7) (2013) 1283–1299.

[4] H. Duan, X. Zhang, J. Wu, G. Ma, Max-min adaptive ant colony optimization approach to multi-UAVs coordinated trajectory replanning in dynamic and uncertain environments, J. Bionic Eng. 6 (2) (2009) 161–173.

[5] Z. Wang, L. Liu, T. Long, Minimum-time trajectory planning for multi-unmanned-aerial-vehicle cooperation using sequential convex programming, J. Guid. Control Dyn. 40 (11) (2017) 2976–2982.

[6] X. Sun, C. Cai, X. Shen, A new cloud model based human-machine cooperative path planning method, J. Intell. Robot. Syst. 79 (1) (2015) 3–19.

[7] Y. Chen, J. Yu, X. Su, G. Luo, Path planning for multi-UAV formation, J. Intell. Robot. Syst. 77 (1) (2014) 229–246.

[8] P. Bhattacharya, M. Gavrilova, Roadmap-based path planning - Using the Voronoi diagram for a clearance-based shortest path, IEEE Robot. Autom. Mag. 15 (2) (2008) 58–66.

[9] P. Maini, P. Sujit, Path planning for a UAV with kinematic constraints in the presence of polygonal obstacles, in: Proceedings of the International Conference on Unmanned Aircraft Systems, Arlington, VA, 2016, pp. 62–67. June 7–10.

[10] K. Yao, J. Li, B. Sun, J. Zhang, An adaptive grid model based on mobility constraints for UAV path planning, in: Proceedings of the International Conference on Control Science and Systems Engineering, Singapore, 2016, pp. 207–211. July 27–29.

[11] X. Hu, L. Chen, B. Tang, D. Cao, et al., Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles, Mech., Syst. Signal Proc. 100 (2018) 482–500.

[12] J. Fierrez, E. Navarro, Adaptive morphological time stamp segmentation based on efficient global motion estimation, in: Proceedings of the International Conference on Image Processing, 3, Rochester, NY, USA, 2002, pp. 289–292. September 22–25.

[13] N. Zeng, Z. Wang, H. Zhang, F.E. Alsaadi, A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay, Cogn. Comput. 8 (2) (2016) 143–152.

[14] N. Zeng, H. Zhang, W. Liu, F.E. Alsaadi, A switching delayed PSO optimized extreme learning machine for short-term load forecasting, Neurocomputing 240 (2017) 175–182.

[15] M. Phung, C. Quach, T. Dinh, Q. Ha, Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection, Autom. Constr. 81 (2017) 25–33.

[16] X. Zhang, H. Duan, An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning, Appl. Soft. Comput. 26 (2015) 270–284.

[17] Z. Sun, J. Wu, J. Yang, Y. Huang, et al., Path planning for GEO-UAV bistatic SAR using constrained adaptive multiobjective differential evolution, IEEE Trans., Geosci. Remote Sens. 54 (11) (2016) 6444–6457.

[18] H. Duan, L. Huang, Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning, Neurocomputing 125 (2014) 166–171.

[19] Y. Chen, Y. Mei, J. Yu, X. Su, et al., Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm, Neurocomputing 266 (2017) 445–457.

[20] Y. Chen, J. Yu, Y. Mei, Y. Wang, et al., Modified central force optimization (MCFO) algorithm for 3D UAV path planning, Neurocomputing 171 (2016) 878–888.

[21] A.Q. Faridi, S. Sharma, A. Shukla, R. Tiwari, et al., Multi-robot multi-target dynamic path planning using artificial bee colony and evolutionary programming in unknown environment, Intell., Serv. Robot. 11 (2) (2018) 171–186.

[22] H. Duan, P. Qiao, Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning, Int. J. Intell. Comput. Cybern. 7 (1) (2014) 24–37.

[23] H. Duan, X. Wang, Echo state networks with orthogonal pigeon-inspired optimization for image restoration, IEEE Trans. Neural Netw. Learn. Syst. 27 (11) (2016) 2413–2425.

[24] J. Zhao, R. Zhou, Pigeon-inspired optimization applied to constrained gliding trajectories, Nonlinear Dyn. 82 (4) (2015) 1781–1795.

[25] L. Xin, N. Xian, Biological object recognition approach using space variant resolution and pigeon-inspired optimization for UAV, Sci. Chin. Technol. Sci. 60 (10) (2017) 1577–1584.

[26] J. Yu, Y. Pei, H. Takagi, Accelerating evolutionary computation using estimated convergence points, in: Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, Canada, 2016, pp. 1438–1444. July 24–29.

[27] B. Zhang, H. Duan, Three-dimensional path planning for uninhabited combat aerial vehicle based on predator–prey pigeon-inspired optimization in dynamic environment, IEEE-ACM Trans. Comput. Biol. Bioinform. 14 (1) (2017) 97–107.

[28] M. Nagy, Z. Akos, D. Biro, T. Vicsek, Hierarchical group dynamics in pigeon flocks, Nature 464 (7290) (2010) 890–899.

[29] M. Nagy, G. Vasarhelyi, B. Pettit, I. Roberts-Mariani, Context-dependent hierarchies in pigeons, Proc. Natl. Acad. Sci. 110 (32) (2013) 13049–13054.

[30] H. Zhang, Z. Chen, T. Vicsek, G. Feng, et al., Route-dependent switch between hierarchical and egalitarian strategies in pigeon flocks, Sci. Rep. 4 (5805) (2014), doi:10.1038/srep05805.

[31] Q. Luo, H. Duan, Distributed UAV flocking control based on homing pigeon hierarchical strategies, Aerosp. Sci. Technol. 70 (2017) 257–264.

[32] H. Qiu, H. Duan, Multiple UAV distributed close formation control based on in-flight leadership hierarchies of pigeon flocks, Aerosp. Sci. Technol. 70 (2017) 471–486.

[33] Q. Liu, Order-2 stability analysis of particle swarm optimization, Evol. Comput. 23 (2) (2015) 187–216.

[34] C. Leboucher, H. Shin, P. Siarry, S. Le Menec, Convergence proof of an enhanced particle swarm optimisation method integrated with evolutionary game theory, Inf. Sci. 346–347 (2016) 389–411.

[35] R. Poli, Mean and variance of the sampling distribution of particle swarm optimizers during stagnation, IEEE Trans. Evol. Comput. 13 (4) (2009) 712–721.

[36] M. Bonyadi, Z. Michalewicz, Stability analysis of the particle swarm optimization without stagnation assumption, IEEE Trans. Evol. Comput. 20 (5) (2016) 814–819.

[37] M. Taboga, Lectures on Probability Theory and Mathematical Statistics, second ed., CreateSpace Independent Pub., 2012.

[38] J. More, S. Wild, Benchmarking derivative-free optimization algorithms, SIAM J. Optim. 20 (1) (2009) 172–191.

[39] Q. Liu, W. Chen, J. Deng, T. Gu, et al., Benchmarking stochastic algorithms for global optimization problems by visualizing confidence intervals, IEEE T., Cybern. 47 (9) (2017) 2924–2937.

[40] M.H. Tayarani-N, M.R. Akbarzadeh-T, Improvement of the performance of the quantum-inspired evolutionary algorithms: structures, population, operators, Evol. Intel. 7 (2014) 219–239.

Daifeng Zhang received his MS degree in Control Theory and Engineering from Beihang University (BUAA), Beijing, China, in 2016. He is currently pursuing the Ph.D. degree in Guidance, Navigation and Control of Aerial Vehicles at the School of Automation Science and Electrical Engineering, Beihang University. His current research interest includes navigation, control and decision of unmanned aerial vehicles, bio-inspired computation, and artificial intelligence.

Haibin Duan is currently a full professor of School of Automation Science and Electrical Engineering, Beihang University (BUAA), Beijing, China. He received the Ph.D. degree from Nanjing University of Aeronautics and Astronautics (NUAA) in 2005. He was an academic visitor of National University of Singapore (NUS) in 2007, a senior visiting scholar of The University of Suwon (USW) of South Korea in 2011. He is currently an IEEE Senior Member and IFAC TC7.5 Technical Committee Member. He has published 3 monographs and over 60 peer-reviewed papers in international journals. His current research interests include bio-inspired computation, advanced flight control, and biological computer vision.