Research Article

Multi-robot path planning using an improved self-adaptive particle swarm optimization

Biwei Tang¹, Kui Xiang¹, Muye Pang¹ and Zhu Zhanxia²

Abstract
Path planning is of great significance in motion planning and cooperative navigation of multiple robots. Nevertheless, because of its high complexity and nondeterministic polynomial time hard nature, efficiently tackling with the issue of multi-robot path planning remains greatly challenging. To this end, enhancing a coevolution mechanism and an improved particle swarm optimization (PSO) algorithm, this article presents a coevolution-based particle swarm optimization method to cope with the multi-robot path planning issue. Attempting to well adjust the global and local search abilities and address the stagnation issue of particle swarm optimization, the proposed particle swarm optimization enhances a widely used standard particle swarm optimization algorithm with the evolutionary game theory, in which a novel self-adaptive strategy is proposed to update the three main control parameters of particles. Since the convergence of particle swarm optimization significantly influences its optimization efficiency, the convergence of the proposed particle swarm optimization is analytically investigated and a parameter selection rule, sufficiently guaranteeing the convergence of this particle swarm optimization, is provided in this article. The performance of the proposed planning method is verified through different scenarios both in single-robot and in multi-robot path planning problems. The numerical simulation results reveal that, compared to its contenders, the proposed method is highly promising with respect to the path optimality. Also, the computation time of the proposed method is comparable with those of its peers.

Keywords
Multi-robot path planning, evolutionary game theory, coevolutionary strategy, convergence analysis of PSO, improvement of PSO

Introduction
Thanks to its significance in motion planning and cooperative navigation, the multi-robot path planning problem has recently aroused great research interest of researchers from the community of robotics.¹ Generally, the multi-robot path planning issue aims to generate an optimal or near-optimal obstacle-free path for each single robot from a start location to a destination location in a given obstacle-rich workspace, meanwhile optimizing the global cost function of the overall system.² However, the

¹ School of Automation, Intelligent Systems Research Institute, Wuhan University of Technology, Wuhan, Hubei, China
² National Key Laboratory of Aerospace and Flight Dynamics, School of Astronautics, Northwestern Polytechnical University, Xi’an, Shaanxi, China

Corresponding author:
Kui Xiang, School of Automation, Intelligent Systems Research Institute, Wuhan University of Technology, Wuhan, Hubei 430070, China.
Email: xkarcher@126.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
path planning problem is naturally proven to be a non-deterministic polynomial time hard (NP-hard) issue, which results in efficiently coping with this problem being difficult. Besides, since the space coordinations among robots need to be considered, the multi-robot path planning problem is more complicated than the single-robot path planning problem.

Basically, the issue of multi-robot path planning can be solved based on centralized or decentralized manners. Under the centralized scenario, a central path planner considers all interactions and constraints among all robots and simultaneously yields a candidate path for each robot. This would lead the centralized path planner to the production of a complicated configuration space and increase the likelihood of failure in generating paths. On the other hand, the decentralized planning assigns an independent planner for each robot. In such a case, each robot strives to separately generate a path in its own configuration space, which can not only simplify the planning issue to enhance the possibility of searching feasible paths for all robots but also save the computation time of the planner.

In virtue of the aforementioned advantages of the decentralized planning, developing efficient decentralized path planning methods becomes one of the most popular streams in the multi-robot path planning field. Implementing the lattice structure to be the road map, Yu et al. developed a decentralized algorithmic framework to deal with the multi-robot path planning problem in Yu and Rus. Based on the generalized connectivity, Solana et al. have addressed the issue of producing paths for a team of robots in a cluttered workspace. Recently, a decentralized model predictive control based multi-robot path planning algorithm has been proposed by Tallamraju et al. in their work. Integrating a coevolution strategy with genetic algorithm (GA), a coevolutionary improved GA (CIGA) has been developed for global path planning of multiple mobile robots in Qu et al. To efficiently tackle with the path planning issue of multiple mobile robots in continuous environments, an enhanced GA has been mixed with artificial potential field algorithm to solve the multi-robot path planning in Nazarahari et al.

As one of the most well-known evolutionary algorithms, probably thanks to its formidable capability over NP-hard problems, easy implementation, and fast convergence speed, particle swarm optimization (PSO) algorithm has recently drawn increasing attention in the field of multi-robot path planning. Unfortunately, the overall performance of the conventional PSO is constrained by its insufficient capability in adjusting the global and local search capabilities as well as the ease of plugging into stagnation in the case where the particle cannot search a position better than the previous ones. In addition, when developing a PSO-based optimization method, the convergence of PSO remains paramount and significantly affects the performance of PSO. Nevertheless, like the majority of the population-based algorithms, the stochastic nature of PSO leads the analytical convergence investigation on PSO to be challenging. This leads to the hypothesis that the optimization performance of PSO over the multi-robot path planning problem could be amended via remedying these aforementioned deficiencies of the conventional PSO.

Combining standard PSO 2011 (SPSO 2011) with evolutionary game theory (EGT), a self-adaptive evolutionary game-based PSO (SAEGBPSO) algorithm is first proposed in this article. Attempting to mitigate the stagnation in SAEGBPSO, particles in this algorithm update their position and velocity information based on moving rules defined in SPSO 2011. To well trade-off the global and local search abilities, a new self-adaptive parameter adaptation rule determined by the evolutionary strategy of EGT is proposed to update the three control parameters of particles in SAEGBPSO. Besides, a convergence-guaranteed parameter selection principle is provided for the proposed PSO algorithm followed after the analytical convergence investigation of this algorithm. Leveraging SAEGBPSO and the coevolutionary strategy, a SAEGBPSO-based decentralized path planner is then developed in this study to generate collision-free and intercollision-free paths for multiple robots and achieve simultaneous arrivals of all robots to the target position. In the designed path planning method, each robot assigns a SAEGBPSO subpopulation and searches its path in the given workspace based on two steps. Firstly, merely considering the physical constraints and the workspace information of each robot, each SAEGBPSO subpopulation searches several feasible and obstacle-free candidate paths for the associated robot. Next, only considering the space and time constraints among different robots, each SAEGBPSO subpopulation exchanges its information with the others among an information interaction operator, such that an elite intercollision avoidance path is generated for each robot to achieve simultaneous arrival of each robot to the destination position.

The remainder of this article is organized as follows. The second section formulates the studied multi-robot path planning problem. The third section mainly states the proposed SAEGBPSO algorithm. The fourth section investigates the analytical properties, mainly including the convergence analysis and the convergence-guaranteed parameter selection principle, of the proposed SAEGBPSO. The coevolution-based SAEGBPSO method for the multi-robot path planning issue is presented in the fifth section. The sixth section conducts the numerical simulations and comparisons. The seventh section ends this article by drawing conclusions.

### Problem statement and formulation

**Modeling of the workspace**

Under the background of multi-robot cooperative operation, the method stated in Dun-wei et al. is used to model
the workspace of robots in virtue of its easy implementation and small sensitivity to shapes of obstacles. As shown in Figure 1, a global coordinate system, represented by $o_{xy}$, is first established, where $st$ and $ta$, respectively, are the start and destination positions of a robot. The $x$-axis of $o_{xy}$ is assumed to coincide with line $st\ a$. The $y$-axis of $o_{xy}$ is supposed to be perpendicular to $st\ a$. In this global coordinate system, line $st\ a$ is equally divided into $ns \ + \ 1$ subsections by $ns$ waypoints. Here, $ns$ is a constant parameter that is predefined by users or decision-makers. As displayed in Figure 1, after gaining a set of vertical lines $l_1, l_2, \ldots, l_{ns+1}$, a candidate path for each robot, represented by $ph = [st, w_1, w_2, \ldots, w_{ns}, ta]$, can be then searched on lines $l_1, l_2, \ldots, l_{ns+1}$.

**Problem formulation**

This article focuses on the scenario where multiple robots cooperate with each other to execute some complex tasks in a hazard-rich workspace. To enhance the success possibility of performing tasks, the generated paths for robots need to satisfy the following conditions: (a) the generated paths need to be obstacle-free; (b) the generated paths ought to be intercollision-free; and (c) the generated paths must guarantee the simultaneous arrivals of robots to the destination location. Moreover, the velocity of each robot $i$ is assumed to be bounded within $[V_{imi}, V_{imax}]$, where $V_{imi}$ and $V_{imax}$, respectively, stand for the minimum and maximum velocities of robot $i$. During navigating a robot to the destination location, the velocity of each robot varies to guarantee the simultaneous arrivals of robots to the destination. Moreover, the velocity of each robot varies during each robot moving to the destination location, the yaw angle and moving distance of each robot need to stay within its maximum yaw angle and distance constraints, respectively.

In this study, the total path length of the multi-robot system, that is, the summation of each robot’s path length is regarded as the global objective function in the studied path planning issue. Assume that $w_{i,0}$ and $w_{i,ns+1}$, respectively, denote the start and destination locations of the $i$th robot. Let $ph_i = [w_{i,0}, w_{i,1}, \ldots, w_{i,ns}, w_{i,ns+1}]$ denote the generated path for the $i$th robot, where $w_{i,k} (k = 0, 1, \ldots, ns + 1)$ denote the $k$th waypoint of the generated path for the $i$th robot. The multi-robot path planning issue we concerned aims to generate an obstacle-free and collision avoidance path $ph_i$ for each robot $i$ to guarantee the simultaneous arrival of each robot to the destination, meanwhile minimizing the overall path length of the multi-robot system. Thus, the multi-robot path planning issue can be mathematically established as follows

$$\begin{align*}
\text{minimize: } & J_{\text{cost}} = \sum_{i=1}^{N} L_i \\
\text{finding: } & ph_i = [w_{i,0}, w_{i,1}, \ldots, w_{i,ns}, w_{i,ns+1}]
\end{align*}$$

Subject to

$$\begin{align*}
L_i & \leq L_{i} \max & (1) \\
\psi_{i,k} & \leq \psi_{i} \max, & 1 \leq k \leq ns & (2) \\
w_{i,k}w_{i,k+1} & \in \text{semi-free workspace}, & 0 \leq k \leq ns & (3) \\
ph_{i} \cap ph_{j} & \in \text{null}, & \forall i \neq j, i,j \in N & (4) \\
T_{i} \cap T_{j} \cap \ldots, \cap T_{N} & \notin \text{non-null} & (5)
\end{align*}$$

where

$$L_i = \sum_{k=0}^{ns} \text{dis}(w_{i,k}, w_{i,k+1})$$

$$\psi_{i,k} = \arccos \left[ \frac{(x_{i,k+1} - x_{i,k})(x_{i,k+2} - x_{i,k+1}) + (y_{i,k+1} - y_{i,k})(y_{i,k+2} - y_{i,k+1})}{\text{dis}(w_{i,k}, w_{i,k+1}) \cdot \text{dis}(w_{i,k+2}, w_{i,k+1})} \right]$$

where $N$ denotes the total number of robots. $L_i$ and $L_{i} \max$ denote the path length and the maximum moving distance constraint of the $i$th robot, respectively. $\psi_{i,k}$ represents the yaw angle of the $i$th robot at the $k$th path segment along the generated path. $\psi_{i} \max$ stands for the maximum yaw angle constraint of the $i$th robot. $T_i$ is arrival time of the $i$th robot to the destination location. $\text{dis}(w_{i,k}, w_{i,k+1})$ denotes the Euclidean distance between waypoints $w_{i,k}$ and $w_{i,k+1}$. $x_{i,k}$ and $y_{i,k}$ are, respectively, the $x$-axis and $y$-axis values of waypoint $w_{i,k}$. “Semi-free workspace” in (4) indicates the
entire space which is uncovered by any obstacle in the whole workspace. “Null” in (5) defines that the intersection of the paths of any two different robots is an empty set, requiring that the generated paths of any two different robots need to be intercollision-free. “Non-null” in (6) denotes that the intersection of the arrival time of any two different robots is a nonempty set, indicating that the generated paths of any two different robots can guarantee the simultaneous arrival of different robots to the destination location.

**Statement of the proposed SAEGBPSO**

To achieve the nonstagnation in SAEGBPSO, particles in this algorithm follow the moving rules defined in SPSO 2011 to update their positions and velocities as follows:

\[
\begin{align*}
V_m(t+1) &= wV_m(t) + X'_m(t) - X_m(t) \\
X_m(t+1) &= V_m(t+1) + X_m(t)
\end{align*}
\]  

where \( w \) is a real coefficient, standing for the inertia weight parameter of the particle. \( X_m(t) \) and \( V_m(t) \), respectively, denote the position and velocity of particle \( m \) at iteration \( t \). \( X_m(t) \) indicates a position which is randomly generated in a hypersphere.

Suppose that \( HP(BC_m(t), ||BC_m(t) - X_m(t)||) \) denotes the hypersphere for particle \( m \) at iteration \( t \), where \( BC_m(t) \) and \( ||BC_m(t) - X_m(t)|| \) denote the isobarycenter and radius of the hypersphere, respectively. Similar to SPSO 2011, the coordination of \( BC_m(t) \) in SAEGBPSO is calculated as

\[
BC_m(t) = X_m(t) + \frac{c_1[Bbest_m(t) - X_m(t)] + c_2[Gbest(t) - X_m(t)]}{3}
\]  

where \( c_1 \) and \( c_2 \) are two positive parameters, denoting the cognitive and social acceleration parameters of the particle, respectively. \( Bbest_m(t) \) and \( Gbest(t) \) represent the personal best position of particle \( m \) and the global best position of the swarm at iteration \( t \), respectively.

After obtaining the hypersphere \( HP(BC_m(t), ||BC_m(t) - X_m(t)||) \) for particle \( m \) based on (11) at each iteration, the random position point \( X'_m(t) \) in (9) is then randomly produced in this hypersphere. As given in (9), since the randomly generated position \( X'_m(t) \) is added to the velocity of the particle to be a disturbance, similar to SPSO 2011, particles in SAEGBPSO can keep searching in the search space with a non-null velocity, which could thus avoid particles plugging into stagnation. For more detailed information of SPSO 2011, the reader is referred to Maurice.

Despite being able to achieve nonstagnation in SPSO 2011, this algorithm could not well adjust its global and local search capabilities due to the fact that the three main control parameters (i.e. the inertia weight \( w \), the cognitive acceleration parameter \( c_1 \) and social acceleration parameter \( c_2 \)) remain constant and there exists no distinguish between the cognitive and the social acceleration parameters. To mitigate this flaw in our proposed SAEGBPSO, a self-adaptive parameter updating rule determined by the evolutionary stables strategy (ESS) of EGT and the iteration number of the particle is developed in this study.

Prior to introducing the proposed self-adaptive parameter updating rule in SAEGBPSO, the analogy between EGT and SAEGBPSO is first described as follows: (a) the players or individuals in EGT analogize particles in SAEGBPSO; (b) every particle in SAEGBPSO has three candidate strategies, these are, respectively, moving only according to its inertia weight, just following its personal best memory, and merely following the global best memory of the swarm; and (c) the payoff matrix of EGT is consisted by the mean performance obtained by each particle in SAEGBPSO following a specific strategy.

Let \( e_1, e_2, \) and \( e_3 \) represent the three aforementioned strategies, respectively. The payoff matrix applied in this article is then calculated by

\[
K = \begin{bmatrix}
\text{pf}(e_1) & \frac{\text{pf}(e_1) - \text{pf}(e_2)}{2} & \frac{\text{pf}(e_1) - \text{pf}(e_3)}{2} \\
\frac{\text{pf}(e_2) - \text{pf}(e_1)}{2} & \text{pf}(e_2) & \frac{\text{pf}(e_2) - \text{pf}(e_3)}{2} \\
\frac{\text{pf}(e_3) - \text{pf}(e_1)}{2} & \frac{\text{pf}(e_3) - \text{pf}(e_2)}{2} & \text{pf}(e_3)
\end{bmatrix}
\]  

where \( \text{pf}(e_l) \) \((l = 1, 2, 3)\) represents the payoff that the particle obtains by only following the \( l \)-th strategy.

In EGT, the replicator dynamics equations (RDEs) can be defined by a commonly accepted form as follows:

\[
\dot{p}_l = -p_l(e_l \cdot Kp + p \cdot Kp) 
\]

where \( p_l(l = 1, 2, 3) \) represents the probability distribution of strategy \( p_l \) over the pure strategy \( e_l \). \( K \) is the payoff matrix. \( p = (p_1, p_2, p_3) \) is the set of mixed strategies. Here, it is notable that each element in the set of mixed strategies satisfies:

\[ \sum_{l=1}^{3} p_l = 1 \]  

At each iteration \( t \), the ESS in EGT represents the ratio of each strategy when the population converges toward a stable point and is used to weight the average of the fitness gained by every particle. In this article, the ESS can be denoted as follows

\[
E_m(t) = [Z_1(t), Z_2(t), Z_3(t)]
\]  

Subject to

\[
\sum_{i=1}^{3} Z_i(t) = 1
\]

At each iteration \( t \), the value of \( \text{pf}(e_l) \) \((l = 1, 2, 3)\) is obtained based on the previous experience of each particle as follows
where \( t \) denotes the current iteration number of SAEGBPSO. \( F(X(t_1)) \) is the previously gained fitness value of the particle \( X \) at the previous iteration \( t_1 \) (1 \( \leq t_1 \leq t \)). For more details of EGT, the reader is referred to the literatures.  \(^{19,21,22}\)

Once the \( \text{pf}(e_l) \) (l = 1, 2, 3) of every particle in SAEGBPSO is obtained based on (16), it is then applied to fill the payoff matrix given by (12). Once the payoff matrix is filled, the RDE defined by (13) is used to calculate the corresponding ESS, namely \( E_{ss}(t) = [Z_1(t), Z_2(t), Z_3(t)] \). After the obtainment of \( E_{ss}(t) = [Z_1(t), Z_2(t), Z_3(t)] \), the three ratios \( Z_1(t), Z_2(t), \) and \( Z_3(t) \) in this \( E_{ss} \) are used to adaptively update the three aforementioned control parameters of particles in the proposed self-adaptive parameter updating rule in SAEGBPSO as follows

\[
\text{pf}(e_l) = \frac{\sum_{t_1=1}^{t-1} Z_l(t_1) F(X(t_1))}{t} \tag{16}
\]

where subscripts “s” and “f” in each control parameter denote the initial and final values of the corresponding control parameter, respectively. \( t_{\text{max}} \) is a predefined constant, denoting the maximum iteration number. \( \delta_4 \) is a sufficiently small positive parameter to avoid the denominator of \( \beta \) in (23) becoming zero (\( \delta_4 = 1e - 05 \) in this article). Here, it is noticeable that \( w_s > w_f, c_{1s} > c_{1f}, \) and \( c_{2s} > c_{2f} \) in the above self-adaptive parameter updating rule.

Since the three ratios \( Z_1(t), Z_2(t), \) and \( Z_3(t) \) are implemented to update the control parameters of particles in the proposed self-adaptive strategy in SAEGBPSO, particles could adapt the “shape” of the search space to optimize the search direction of the swarm as far as possible. Also, since the ESS potentially implies the stability of EGT, when the three ratios \( Z_1(t), Z_2(t), \) and \( Z_3(t) \) in ESS are implemented to update the control parameters of particles, they may could face the potential irregularity of the solution space to prevent them plugging into some local optimum. Thus, the implementations of the three ratios in the proposed self-adaptive strategy in SAEGBPSO may enhance the performance of the optimizer in finding high-quality solutions.

Also, it can be seen from (17) to (19) that the inertia weight and cognitive acceleration parameter decreases, whereas the social component increases with the iteration number increasing, which implies that the global search powers of particles are likely to promote in the early stages of the evolution and the local search abilities of particles would enhance in the late of the evolution. Besides, it is evident from (17) to (19) that the inertia weight and the cognitive acceleration parameter decrease, whereas the change in the social acceleration parameter grows greater with \( \beta \) increasing. This may indicate that the global search abilities of particles in SAEGBPSO would be more retained in the case where the value of \( \beta \) remains relatively big. Based on (23), a large value of \( \beta \) means that the value of \( (Z_1(t) + Z_2(t)) \) is relatively big. Since a relatively big of \( (Z_1(t) + Z_2(t)) \) denotes that the search direction of the particle is more stable when the particle mainly adopts the strategy of its inertial and personal best experience, the swarm can benefit more in the case where each particle follows these two strategies. Thus, in the case where \( \beta \) is great, it is reasonable to increase the inertia weight and the cognitive acceleration parameter, so that the global search abilities of particles in SAEGBPSO can be enhanced. On the other hand, \( Z_3(t) \) has a relatively big value in the case where \( \beta \) is small, based on (23). In such case, the searches of particles can be more stable when particles follow the strategy of the global best experience of the swarm. Therefore, it is natural to increase the social acceleration parameter of the particle to enhance the local search ability of the proposed SAEGBPSO in the case where \( \beta \) is small.

### Theoretical analysis for SAEGBPSO

#### Convergence investigation of SAEGBPSO

As stated previously, the convergence property of PSO remains an important issue when designing a PSO-based optimizer. The convergence investigation of PSO aims to find the control parameter boundaries to guarantee the convergence of the developed PSO optimizer. The convergence-guaranteed control parameter boundaries of different PSO algorithms could be different since different PSO algorithms may adopt different moving and control parameter updating rules. Thus, there exists necessity to theoretically analyze the convergence of different PSO algorithms.

Based on this logical flow noted above and inspired by some studies on the convergence analysis of their proposed
PSO algorithms, such as Tang et al. and Cédric et al., this study first investigates the convergence of SAEGBPSO, such that the convergence-guaranteed control parameter boundaries can be found. Based on the obtained convergence-guaranteed control parameter boundaries and the proposed control parameter updating rule defined by (17)–(22), this article then provides a convergence-guaranteed control parameter setting rule to sufficiently guarantee the convergence of the proposed SAEGBPSO.

In this section, the convergence of SAEGBPSO is investigated based on the deterministic model convergence analysis, that is, under the situation where $X' = BC_m(t)$. Recall that each particle in the proposed SAEGBPSO adopts the moving rule defined in (9)–(10) to renew its velocity and position information. Thus, without loss of generality and for simplicity, the subscript $m$ in each variable in (9)–(10) can be omitted in terms of the convergence investigation of the proposed SAEGBPSO. Also, please note that since every dimension in the velocity and position vectors of each particle in SAEGBPSO is updated independently from the others in the moving rules defined in (9)–(10), the moving rule defined by (9)–(10) in SAEGBPSO can be simplified and rewritten into a one-dimensional matrix form as follows

$$
\begin{bmatrix}
X(t+1) \\
V(t+1)
\end{bmatrix} =
\begin{bmatrix}
1-c & w \\
-c & w
\end{bmatrix}
\begin{bmatrix}
X(t) \\
V(t)
\end{bmatrix} + \begin{bmatrix}
c
\end{bmatrix} D
$$

(24)

where

$$
c = \frac{c_1 + c_2}{3}
$$

(25)

$$
D = \frac{c_1 P_{\text{best}} + c_2 G_{\text{best}}}{c_1 + c_2}
$$

(26)

Here, it is important to note that since the proposed SAEGBPSO is rewritten into the dynamic system denoted by (24), the convergence stability of the proposed SAEGBPSO is equivalent to that of the dynamic system denoted by (24). Thus, if the convergence stability of the dynamic system denoted by (24) is analytically investigated, the convergence of the proposed PSO algorithm is then obtained. Also, note that the dynamic system (24) is a first-order constant-coefficient nonhomogeneous difference equation. There are many mature methods to solve this difference equation. The characteristic equation could be one of the most typical and popular methods. The characteristic equation of system (24) is easily obtained as

$$
\eta^2 - (1 + w - c)\eta + w = 0
$$

(27)

Then, the two characteristic roots, denoted as $\eta_1$ and $\eta_2$, to (27) are easily gained as

$$
\eta_{1,2} = \frac{1 + w - c \pm \sqrt{(1 + w - c)^2 - 4w}}{2}
$$

(28)

According to the standard results of the dynamic system theory, system (24) converges iff (“iff” denotes “if and only if” in this study)

$$
\max\{|\eta_1|, |\eta_2|\} < 1
$$

(29)

Because $\eta_1$ and $\eta_2$ can be two real or complex roots in (28), both these two cases are investigated separately below.

1. **Case 1.** Both $\eta_1$ and $\eta_2$ are complex, denoted as $\eta_{1,2} \in \mathbb{C}$.

   **Lemma 1.** For (27), $\eta_{1,2} \in \mathbb{C}$, iff

   $$
   \begin{align*}
   1 + w - 2\sqrt{w} < c < 1 + w + 2\sqrt{w} \\
   w \geq 0
   \end{align*}
   $$

   (30)

   **Proof.** Clearly, we have that $\eta_{1,2} \in \mathbb{C}$, iff

   $$
   (1 + w - c)^2 - 4w < 0
   $$

   (31)

   Based on the classical mathematical approach, the proof of Lemma 1 can be easily completed.

   **Lemma 2.** System (24) converges under the situation where $\eta_{1,2} \in \mathbb{C}$, iff

   $$
   \begin{align*}
   1 + w - 2\sqrt{w} < c < 1 + w + 2\sqrt{w} \\
   0 \leq w < 1
   \end{align*}
   $$

   (32)

   **Proof.** Because the magnitude of an imaginary number $H$ is gained by $|H| = \sqrt{H_1^2 + H_2^2}$, where $H_1$ and $H_2$, respectively, stand for the real and imaginary parts this imaginary number, for $\eta_{1,2} \in \mathbb{C}$, we have

   $$
   \max\{|\eta_1|, |\eta_2|\} = \sqrt{w}
   $$

   (33)

   Therefore, for $\eta_{1,2} \in \mathbb{C}$, system (24) converges, iff

   $$
   \sqrt{w} < 1
   $$

   (34)

   For $\eta_{1,2} \in \mathbb{C}$, (30) holds, according to Lemma 1. Hence, considering conditions that $\eta_{1,2} \in \mathbb{C}$ and $\max\{|\eta_1|, |\eta_2|\} < 1$ together, system (24), that is, SAEGBPSO, converges under the situation where $\eta_1$ and $\eta_2$ are complex, iff

   $$
   \begin{align*}
   1 + w - 2\sqrt{w} \leq c \leq 1 + w + 2\sqrt{w} \\
   0 \leq w < 1
   \end{align*}
   $$

   (35)

   For $\eta_{1,2} \in \mathbb{C}$, the convergence domain of SAEGBPSO on different control parameter plans is demonstrated in Figure 2.

2. **Case 2.** $\eta_{1,2} \in \mathbb{R}$, where $\mathbb{R}$ represents real-valued domain.

   **Lemma 3.** The two roots $\eta_{1,2} \in \mathbb{R}$, iff

   $$
   \begin{align*}
   c & \in \mathbb{R}, w < 0 \\
   c & \leq 1 + w - 2\sqrt{w} \text{ or } c \geq 1 + w + 2\sqrt{w}, w \geq 0
   \end{align*}
   $$

   (36)

   **Proof.** Clearly, for (27), both $\eta_{1,2}$ are two real roots, iff

   $$
   (1 + w - c)^2 - 4w \geq 0
   $$

   (37)
The proof of Lemma 3 can be easily completed by expanding (37).

Next, in the case where \( h_1, h_2 \in \mathbb{R} \), conditions on \( w \) and \( c \), guaranteeing the convergence of system (24), need to be discovered.

**Lemma 4.** System (24) converges under the situation where \( h_1, h_2 \in \mathbb{R} \), iff

\[
\begin{align*}
0 & < c < 2w + 2, \quad -1 < w < 0 \\
0 & < c \leq 1 + w - 2\sqrt{w} \quad \text{or} \quad 1 + w + 2\sqrt{w} \leq c < 2w + 2, \quad 0 \leq w < 1
\end{align*}
\]  

(38)

**Proof:** From (28) and (29), for \( h_1, h_2 \in \mathbb{R} \), \( \max \{ |h_1|, |h_2| \} \) < 1 trivially meets, iff

\[
-1 < \frac{1 + w - c \pm \sqrt{(1 + w - c)^2 - 4w}}{2} < 1
\]  

(39)

In another item, iff

\[
c - w - 3 < \pm \sqrt{(1 + w - c)^2 - 4w} < c - w + 1
\]  

(40)

For \( h_{1,2} \in \mathbb{R} \), one can easily obtain from (40) that

\[
(39) \Leftrightarrow \begin{cases} \\
\frac{(1 + w - c)^2 - 4w}{4} < 3 + w - c \\
\frac{(1 + w - c)^2 - 4w}{4} < c - w + 1
\end{cases}
\]

(41)

Simplifying (41), yields

\[
(39) \Leftrightarrow \begin{cases} \\
2w + 2 - c > 0 \\
c > 0
\end{cases}
\]

(42)

According to Lemma 3, (36) needs to be held for \( h_{1,2} \in \mathbb{R} \). Thus, considering conditions: \( h_{1,2} \in \mathbb{R} \) and \( \max \{ |h_1|, |h_2| \} < 1 \) together, we can conclude that, in the case where \( h_{1,2} \in \mathbb{R} \), system (24) converges, iff

\[
\begin{align*}
0 & < c < 2w + 2, \quad -1 < w < 0 \\
0 & < c \leq 1 + w - 2\sqrt{w} \quad \text{or} \quad 1 + w + 2\sqrt{w} \leq c < 2w + 2, \quad 0 \leq w < 1
\end{align*}
\]  

(43)

Figure 3 displays the convergence region of SAEGBPSO on different parameter plans in the case where \( h_{1,2} \in \mathbb{R} \).

Finally, integrating the conclusion drawn in Lemma 2 with that in Lemma 4, it allows us to conclude that system (24), namely, SAEGBPSO, converges, iff

\[
\begin{align*}
0 & < c < 2w + 2 \\
-1 & < w < 1
\end{align*}
\]  

(44)

The real convergence region of the SAEGBPSO on different parameter plans is shown in Figure 4.
Parameter selection principle for SAEGBPSO

Please recall that, through the analytical convergence investigation of SAEGBPSO conducted in the above subsection, it is conclusive that if and only if conditions given by (44) are satisfied, SAEGBPSO converges. Now, for the sake of sufficiently satisfying the conditions given by (44), so that the convergence of SAEGBPSO can be sufficiently guaranteed, this subsection provides a convergence-guaranteed parameter selection principle for this algorithm.

Lemma 5. The convergence of SAEGBPSO can be sufficiently guaranteed, only if the three control parameters shown in the proposed self-adaptive updating strategy defined by (17)–(22) satisfy the following conditions

\[ 0 < c_1s + c_1f < 6w_f + 6 \]
\[ -1 < w_f < w_s < 1 \]
\[ c_{1s} = c_{2f} > c_{1f} = c_{2s} > 0 \]

Proof. Since \( c = (c_1 + c_2)/3 \), the sufficient and necessary condition given by (44) for the convergence of SAEGBPSO can be rewritten as follows, namely, SAEGBPSO converges, iff

\[ 0 < c_1 + c_2 < 6w + 6 \]
\[ -1 < w < 1 \]

It is trivial from (18), (19), (21), and (22) that \( c_1 + c_2 = c_{1s} + c_{1f} > 0 \) in the case where \( c_{1s} = c_{2f} > c_{1f} = c_{2s} > 0 \). Furthermore, from (18) to (20), it is clear that \( w_f \leq w \leq w_s \), \( c_{1f} \leq c_{1s} \), and \( c_{2s} \leq c_{2f} \). Therefore, we can obtain that

\[ \begin{aligned}
0 < c_{1s} + c_{1f} &< 6w_f + 6 \\
-1 < w_f &< w_s < 1 \\
c_{1s} = c_{2f} &> c_{1f} = c_{2s} > 0
\end{aligned} \]

\[ \begin{aligned}
0 < c_1 + c_2 &< 6w + 6 \\
-1 < w &< 1
\end{aligned} \]

(47)

Notice that the right-hand side conditions given in (47) denote the necessary and sufficient condition for the convergence of SAEGBPSO. Thus, the proof of Lemma 5 can be easily completed according to (47).

Note that the condition given by (45) is a sufficient condition for the convergence of SAEGBPSO, which means that only if the condition given by (45) is satisfied, the convergence of SAEGBPSO can be sufficiently guaranteed. Also, it is noticeable that, since \( w_s, w_f, c_{1s}, c_{1f}, c_{2s} \), and \( c_{2f} \) are predefined constant parameters, the condition given by (45) can satisfy easily through setting proper values of these predefined parameters. Here, the values of these mentioned parameters are empirically suggested as \( w_s = 0.9, w_f = 0.1, c_{1s} = c_{2f} = 2.5 \), and \( c_{1f} = c_{2s} = 0.1 \).
The convergence trajectories of the position and velocity of the particle in SAEGBPSO are visualized in Figure 5 under the suggested parameter settings mentioned here.

The coevolution-based SAEGBPSO method for multi-robot path planning

As stated above, a coevolutionary strategy presented in Qu et al. and Kala9,23 is incorporated with SAEGBPSO to solve the mutli-robot path planning problem based on a decentralized manner. In the developed planning method, each robot is assigned a SAEGBPSO subpopulation. At each iteration, only considering the physical constraints (i.e., the maximum moving distance and yaw angle constraints) of the robot and the path safety constraint caused by obstacles, each subpopulation first separately and independently evolves in its own configuration to find EP candidate paths for the associated robot. Then, each
subpopulation reports the EP candidate paths to an information interaction operator, so that each subpopulation can exchange information with the others. During the information exchange stage, considering the space and time constraints among different robots, an elite obstacle-free and intercollision avoidance path that can achieve simultaneous arrivals of robots to the destination position is selected from the EP candidate paths for each SAEGBPSO subpopulation.

Obviously, the value of EP greatly influences the performance of the path planner. The greater the value of EP is, with the higher possibility the planner can find an optimal path for the robot; however, the more computation time the planner consumes. Usually, compromisingly considering the path optimality and the computation time, EP is set to be a predefined constant. In this study, similar to the exact mechanisms or impacts regarding how the value of EP affects the performance of the path planner is uncovered, which could be considered as an extension of this paper in the near future. Besides, we must highlight that, similar to other terrific studies, such as literatures, using coevolutionary mechanism to handle multiple vehicles path planning, the coevolution-based SAEGBPSO method can also solve the single-robot path planning problem in the case where EP = 1 and only one SAEGBPSO subpopulation is considered.

The flowchart of the coevolution-based SAEGBPSO for the multi-robot path planning issue is illustrated in Figure 6. In the herein subsections, the ways of encoding particles and handling constraints as well as the design of the information interaction operator are presented during the application of the designed path planning method on the multi-robot path planning problem.

**Encoding particles**

The aim of encoding the particle is to find a mapping between the particle’s position and the potential solution to an optimization problem. As depicted previously, since the purpose of the studied path planning problem is to produce a path for each robot, constructed by waypoints \( w_1, w_2, \ldots, w_m \) that are randomly sampled in lines \( l_1, l_2, \ldots, l_{2w} \). Because the \( x \)-axis values of lines \( l_1, l_2, \ldots, l_{2w} \) are given beforehand after the establishment of the workspace of robot, values of waypoints \( w_1, w_2, \ldots, w_m \) lie only with the \( y \)-axis values decided by lines \( l_1, l_2, \ldots, l_{2w} \). Thus, these \( y \)-axis values, represented by \( y_{w_1}, y_{w_2}, \ldots, y_{w_m}, y_{w_{m+1}} \), are used to encode particles in the designed path planning method, where \( y_{w,n} \) and \( y_{w,n+1} \) are the \( y \)-axis values of the start and destination positions of the \( n \)th robot, respectively. For guaranteeing each particle to search within the workspace, the following saturation strategy is implemented to modify the designing variable \( y_{wi} (i = 1, ..., ns) \) in the case where \( y_{wi} \) locates outside the workspace as follows

\[
y_{wi} = \begin{cases} 
  \frac{\text{width}}{2}, & \text{if } y_{wi} > \frac{\text{width}}{2} \\
  \frac{-\text{width}}{2}, & \text{if } y_{wi} < -\frac{\text{width}}{2} \\
  y_{wi}, & \text{otherwise}
\end{cases}
\]  

(48)

where width is the width of the workspace, as shown in Figure 1.

**Constraints handling**

As given in (2)–(6), the studied path planning problem is modeled into a constrained optimization. To efficiently solve this problem and guarantee the feasibility of the generated path, the mission concerning how to tackle with constraints of the problem must address. To deal with the physical and path safety constraints given by (2)–(4), the total constraint violation degree of the physical and path safety constraints of each particle \( m \) in designed path planning method is calculated as follows

Figure 6. The flow chart of coevolution-based SAEGBPSO method for multi-robot path planning. SAEGBPSO: self-adaptive evolutionary game-based particle swarm optimization.
tdm = dxn + dfm + dxfn  

where Dxn, Dfm, and Dxfn stand for the degrees that particle m violates the path safety constraint caused by obstacles, maximum moving distance, and yaw angle constraints, respectively.

Given Nob obstacles, Dxn can be obtained by

\[ D_{xn} = \frac{1}{\text{Nob}} \sum_{j=1}^{\text{Nob}} V_{mo} \]

\[ V_{mo} = \begin{cases} 1 & \text{if } m \text{ collides with obstacle } o \\ 0 & \text{otherwise} \end{cases} \]

Given the maximum moving distance constraint \( L_i^{\text{max}} \), \( D_{fn} \) is calculated as follows

\[ D_{fn} = \begin{cases} 0 & \text{if } L_m \leq L_i^{\text{max}} \\ \frac{L_m - L_i^{\text{max}}}{L_i^{\text{max}}} & \text{otherwise} \end{cases} \]  

where \( L_m \) denotes the path length of particle \( m \) and is calculated according to (7).

Given the maximum yaw angle constraint \( \psi_i^{\text{max}} \), \( D_{yn} \) can be obtained as follows

\[ D_{yn} = \frac{1}{\text{ns}} \sum_{k=1}^{\text{ns}} A_{mk} \]

\[ A_{mk} = \begin{cases} 0 & \text{if } \psi_i \leq \psi_i^{\text{max}} \\ 1 & \text{otherwise} \end{cases} \]

where \( \text{ns} \) is the predefined parameter described in “Modeling of the workspace” section. \( \psi_i \) denotes the \( i \)th yaw angle along the path represented by particle \( m \) and is calculated based on (8).

After the calculation of the total constraint violation degrees of the physical and path safe constraints using (49)–(52), the feasibility-based rule presented in Deb et al. is then adopted to select the elite solution between any two candidate solutions in each SAEBGPSO subpopulation. In a minimization optimization problem, this rule can be summarized as (a) the solution having smaller fitness value is preferred over the solution with larger fitness value in the case where any two different candidate solutions have a same constraint violation degree and (b) the solution owning smaller constraint violation degree dominates the solution with greater constraint violation degree in the case where any two different candidate solutions have different constraint violation degrees.

Since the fitness value and the constraint violation degree of each candidate solution (or particle) are compared separately in the above feasibility-based rule, no additional penalty or control factor is needed when using this technology to deal with constraints, which can thus reduce the burden of the optimizer. Moreover, despite violating partial constraints, some nonfeasible candidate solutions may also involve some valuable information of the solution space. In such a case, the diversifications of solutions and the possibility of searching high-quality solutions can be increased when considering those nonfeasible solutions in the feasibility-based rule.

In addition to the physical constraints and path safety constraint, as shown in (2)–(4), the space and time constraints given by (5)–(6) also need to be handled in the studied multi-robot path planning problem. Here, please recall that the space and time constraints defined by (5)–(6) require that the generated paths for any two different robots are free of intercollision and can achieve simultaneous arrival of each robot to the destination location, respectively.

For dealing with the space constraint given by (5), the space constraint violation degree of any two different paths \( ph_i \) and \( ph_j \) is calculated as follows

\[ \text{SP}_{\text{viol}} = \begin{cases} 0 & \text{if } d_{ij} > 0 \\ 1 & \text{otherwise} \end{cases} \]

where \( d_{ij} \) denotes the minimum flight distance between any two different paths \( ph_i \) and \( ph_j \).

As depicted in “Problem formulation” section, since each robot \( i \) moves along its path \( ph_i \), with velocity constraints \( [V_i^{\min}, V_i^{\max}] \), the range of the arrival time of this robot to the destination position is then determined by \( T_i = [L_i / V_i^{\max}, L_i / V_i^{\min}] \). To achieve simultaneous arrivals of any two different robots \( i \) and \( j \) to the destination location, only the condition: \( T_i \cap T_j = [L_i / V_i^{\max}, L_i / V_i^{\min}] \cap [L_j / V_j^{\max}, L_j / V_j^{\min}] \), needs to be satisfied. Therefore, for tackling with the time constraint given by (6), the time constraint violation degree between any two different paths \( ph_i \) and \( ph_j \) can obtain as follows

\[ \text{CT}_{\text{viol}} = \begin{cases} 0 & \text{if } V_i^{\max} \leq V_j^{\min} \text{ and } V_i^{\min} \geq V_j^{\max} \\ 1 & \text{otherwise} \end{cases} \]

where \( V_i^{\min} \) and \( V_i^{\max} \) denote the minimum and maximum velocities of robot \( i \), respectively. \( V_j^{\min} \) and \( V_j^{\max} \) are, respectively, the minimum and maximum velocities of robot \( j \). \( L_i \) and \( L_j \) are the path lengths of robot \( i \) and \( j \), respectively.

Design of the information interaction operator

Only considering the space and time constraints among different robots, the primary mission of the information interaction operator is to determine an elite path from the generated EP candidate paths for each SAEBGPSO subpopulation. During the design of the information interaction operator, the first task needed to sort out is the model of the information interaction operator. So far, many existing models, such as the cooperative coevolution model,
competitive coevolution model, and island model, can be used as options.\textsuperscript{8} Thanks to its superiorities of maintaining solution diversifications and employing parallelism,\textsuperscript{8} the island model, as visualized in Figure 7, is applied to build the information interaction operator model in this study. For more information about the island model, the reader is referred to Qu et al.\textsuperscript{8}

The second task in the design of the information interaction operator is to decide the criterion for choosing the elite path from EP candidate paths found by each SAEGBPSO subpopulation. Here, two criteria are used to determine the elite path for each subpopulation at each iteration. The first criterion is the summation of the space and time constraint violation degrees of each robot. Note that the space and time constraint violation degrees are calculated by (53) and (54), respectively. The second criterion is the path length of each robot. After the determinations of the model of information interaction operator and criteria for choosing the elite path, the algorithmic steps of the information interaction operator are described as follows:

Step 1: Based on the island model, for a given SAEGBPSO subpopulation $A$, randomly select a different SAEGBPSO subpopulation $B$.

Step 2: Using the feasibility-based rule stated above, sort individuals in $A$ and $B$ in ascending order according to the total constraint violation degrees of the physical and path safety constraints of each individual. Suppose the sorted individuals in $A$ and $B$ are denoted as $(a_1, a_2, ..., a_{EP})$ and $(b_1, b_2, ..., b_{EP})$, respectively.

Step 3: Traverse each individual in $(a_1, a_2, ..., a_{EP})$. Compute the summation of the space and time constraint violation degrees of the current individual in $(a_1, a_2, ..., a_{EP})$ as well as those of each individual in $(b_1, b_2, ..., b_{EP})$.

Step 4: Select the individual in $(a_1, a_2, ..., a_{EP})$ with least summation of the space and time constraint violation degrees as the elite path for subpopulation $A$. If multiple individuals in $(a_1, a_2, ..., a_{EP})$ have the same least summation of the space and time constraint violation degrees, the individual with the least path length is selected as the elite path for subpopulation $A$ to minimize the total path length of the multi-robot system.

The algorithmic steps of the coevolution-based SAEGBPSO method for the multi-robot path planning are summarized in Table 1. Note that the main loop of the developed path planning method does not exist until the iteration number $t$ of each SAEGBPSO subpopulation reaches its given maximum iteration number $t_{\text{max}}$.

### Table 1. The coevolution-based SAEGBPSO for the multi-robot path planning problem.

1. Assign subpopulations and randomly initialize each subpopulation
2. Obtain $P_{\text{best}}$ and $G_{\text{best}}$ as well as $E_{\text{ss}}$ for each initial subpopulation
3. for each subpopulation do
4. while $t \leq t_{\text{max}}$ do
5. obtain $Z_1(t-1)$, $Z_2(t-1)$, and $Z_3(t-1)$ and the fitness value $f(X(t-1))$ of each particle
6. Calculate $fp(e_1)$, $fp(e_2)$, and $fp(e_3)$ for each particle based on (16)
7. Compute the payoff matrix $K$ for each subpopulation based on (12)
8. Calculate $Z_1(t)$, $Z_2(t)$, and $Z_3(t)$ for each subpopulation based on (13)
9. Update control parameters $w$, $c_1$, and $c_2$ of each particle using (17)–(23)
10. Calculate the barycenter $BC(t)$ of each particle based on (11)
11. Randomly obtain $X'(t)$ within the hypersphere $HP(BC(t), ||BC(t) - X(t)||)$ for each particle
12. Update the velocity of each particle based on (9)
13. Update the position of each particle based on (10)
14. Modify the position vector of each particle based on the saturation strategy given by (48)
15. Compute the payoff $f_{\text{ss}}$ for each initial particle based on (7)
16. Compute the constraint violation degree of each particle based on (49)–(52)
17. Update $G_{\text{best}}$ of each particle based on the feasibility-based rule
18. Preserve EP best candidate paths based on the feasibility-based rule
19. Send the EP candidate paths to the information interaction operator
20. After the information interaction operator, determine $G_{\text{best}}$ of the current subpopulation
21. Increase the iteration number $t$ by 1
22. end while
23. end for
24. Output $G_{\text{best}}$ of each subpopulation to navigate the associated robot

\textbf{SAEGBPSO:} self-adaptive evolutionary game-based particle swarm optimization.

### Numerical simulations

To verify the proposed method, its performance is compared against with those of three evolutionary algorithms: the CIGA,\textsuperscript{8} SPSO 2011,\textsuperscript{20} and the fitness-scaling adaptive chaotic PSO (FACPSO).\textsuperscript{31} The simulation parameters for SAEGBPSO are set to be $w_i = 0.9$, $w_f = 0.1$, $c_{1f} = c_{2f} = 2.5$, and $c_{1s} = c_{2s} = 0.1$ based on the analysis results in “Parameter selection principle for SAEGBPSO” section. The simulation parameters of the three compared methods are extracted from their corresponding literature.

**Figure 7.** The island model.
and presented in Table 2. In each numerical simulation, the final path of each subpopulation in each method is output after 40 particles evolve 300 iterations. As highlighted previously, since the proposed method can not only solve the multi-robot path planning problem, but also the single-robot path planning issue in the case where EP = 1 and the number of SAEGBPSO subpopulation equals to 1, the feasibility and effectiveness of the proposed method are verified in such two cases in the following contents of this section. Also, in each numerical simulation conducted below, SAEGBPSO is referred to be the coevolution-based SAEGBPSO method for the convenience of writing.

### Numerical simulations and comparisons on single-robot path planning

Two numerical simulations under different planning scenarios are conducted in this subsection to evaluate the performance of the proposed method on single-robot path planning. The needed simulation parameters and workspace information of these two numerical simulations are displayed in Table 3. The generated paths and corresponding cost curves of different methods for these two numerical simulations are displayed in Figures 8 and 9, respectively. The simulation results of each method for these two numerical simulations are reported in Tables 4 and 5, respectively.

From Figures 8(a) and 9(a), it is apparent that each planning method can generate an obstacle-free path for the robot, which can reflect the feasibility of these methods in the single-robot path planning problem. Besides, it is evident from Tables 4 and 5 that SAEGBPSO outperforms its contenders in terms of path optimality. Moreover, it can be easily observed from Tables 4 and 5 that SPSO 2011 and SAEGBPSO are, respectively, ranked the first and second among the four methods as far as the computation time is considered. Probably because the three control parameters of particles in SPSO 2011 are invariant, no additional computation resource is required to adjust these control parameters in this algorithm. Therefore, compared to the other three approaches, SPSO 2011 is the computationally cheapest.

However, it is important notice that, despite consuming the least computation time, SPSO 2011 provides the worst performance with respect to the path optimality, and even if the proposed SAEGBPSO is ranked the second in terms of the computation time, SAEGBPSO obtains the best performance in the path optimality, as confirmed in Tables 4 and 5. Also, it is worth mentioning from Tables 4 and 5 that the computation time difference between SPSO 2011 and SAEGBPSO is less than 3 s in these two numerical simulations, which may be neglected based on the fact that the studied path planning problem is executed off-line. Considering all concerns mentioned here, we can conclude that the proposed SAEGBPSO method dominates its opponents in terms of the path optimality. Besides, the proposed method is promising with respect to the computation time in the single-robot path planning problem.

### Numerical simulations on multi-robot path planning

Two numerical simulations are conducted in this subsection to evaluate the feasibility and effectiveness of the proposed method on the multi-robot path planning. In the first simulation scenario, two robots are initially located at a same start position in a 100 m × 80 m workspace and expected to move toward a same destination position. In the second simulation case, five robots are initially located at different positions in a 50 m × 50 m workspace and required to arrive at a same destination position. The position information and physical constraints of each robot for these two numerical simulations are presented in Tables 6 and 7, respectively.

| Methods      | Parameter setting                                                                 |
|--------------|----------------------------------------------------------------------------------|
| CIGA         | $Cr = 0.8$ and $Pc = 0.15$                                                       |
| SPSO 2011    | $w = 0.7213, c_1 = 1.1931$, and $c_2 = 1.1931$                                   |
| FACPSO       | $w_I = 0.9, w_f = 0.4, c_{I_u} = c_{f_l} = 2.5$, and $c_{I_u} = c_{f_l} = 0.5$ |

CIGA: coevolutionary improved genetic algorithm; SPSO: standard particle swarm optimization; FACPSO: fitness-scaling adaptive chaotic particle swarm optimization.

From Figures 10–17, it can be observed that each considered method can successfully generate an obstacle-free and intercollision avoidance path for each robot in each numerical simulation case, which, in a certain degree, reveals the feasibilities of these methods on the multi-robot path planning problem. From Tables 8 and 9, it is clear that the proposed method performs superior to its opponents in terms of the simultaneous arrival time and the total fitness value. Moreover, SPSO 2011 and the proposed method provide the best and second best performances over these two numerical simulations in terms of computation time, as confirmed in Tables 8 and 9.

Again, it is significant to note that although SPSO 2011 is the fastest method with respect to the computation time, this method provides the worst performance in terms of path optimality in these two multi-robot simulation cases,
Table 3. Needed simulation parameters and workspace information.

| Scenario | SW (m × m) | SP (m) | DP (m) | ns | $L_{\text{max}}$ | $\psi_{\text{max}}$ |
|----------|------------|--------|--------|----|----------------|-----------------|
| Scenario 1 | 100 × 100 | (0, −50) | (100.50) | 15 | 250 m | $4\pi/9$ |
| Scenario 2 | 95 × 90 | (0, −35) | (95,45) | 16 | 200 m | $\pi/3$ |

SW: size of the workspace; SP: start position; DP: destination position.

Figure 8. The (a) generated paths and (b) cost curves of different methods for the first numerical simulation on single-robot path planning.

Figure 9. The (a) generated paths and (b) cost curves of different methods for the second numerical simulation on single-robot path planning.

Table 4. Simulation results of different methods for the first numerical simulation on single-robot path planning.

| Method          | Fitness value (m) | Computation time (s) |
|-----------------|-------------------|----------------------|
| SAEGBPSo        | 145.51            | 27.53                |
| FACPSO          | 146.50            | 29.78                |
| CIGA            | 150.95            | 31.45                |
| SPSO 2011       | 181.54            | 24.86                |

SAEGBPSo: self-adaptive evolutionary game-based particle swarm optimization; CIGA: coevolutionary improved genetic algorithm; SPSO: standard particle swarm optimization; FACPSO: fitness-scaling adaptive chaotic particle swarm optimization.

Table 5. Simulation results of different methods for the second numerical simulation on single-robot path planning.

| Method          | Fitness value (m) | Computation time (s) |
|-----------------|-------------------|----------------------|
| SAEGBPSo        | 139.75            | 45.35                |
| FACPSO          | 146.10            | 48.07                |
| CIGA            | 151.01            | 51.33                |
| SPSO 2011       | 153.61            | 42.41                |

CIGA: coevolutionary improved genetic algorithm; SPSO: standard particle swarm optimization; FACPSO: fitness-scaling adaptive chaotic particle swarm optimization; SAEGBPSo: self-adaptive evolutionary game-based particle swarm optimization.

Bold values denote the best result obtained with regard to each metric.
Table 6. The position information and physical constraints of robots for the first numerical simulation on multi-robot path planning.

| Start position | Destination position | velocity bounds (m/s) | n_s | EP | \( L_{\text{max}} \) | \( \psi_{\text{max}} \) |
|----------------|----------------------|-----------------------|-----|----|----------------|----------------|
| Robot 1 (0,0)  | (100,0)              | [4, 6]                | 18  | 5  | 150 m          | 4/9 \( \pi \)  |
| Robot 2 (0,0)  | (100,0)              | [4, 6]                | 18  | 5  | 140 m          | 17/36 \( \pi \) |

Table 7. The position information and physical constraints of robots for the second numerical simulation on multi-robot path planning.

| Start position | Destination position | velocity bounds (m/s) | n_s | EP | \( L_{\text{max}} \) | \( \psi_{\text{max}} \) |
|----------------|----------------------|-----------------------|-----|----|----------------|----------------|
| Robot 1 (0, -25) | (50,0)               | [3, 7]                | 20  | 6  | 80 m           | 2/3 \( \pi \)    |
| Robot 2 (0, -15)  | (50,0)               | [3, 7]                | 20  | 6  | 80 m           | 2/3 \( \pi \)    |
| Robot 3 (0, 0)    | (50,0)               | [3, 7]                | 20  | 6  | 80 m           | 2/3 \( \pi \)    |
| Robot 4 (0, 15)   | (50,0)               | [3, 7]                | 20  | 6  | 80 m           | 2/3 \( \pi \)    |
| Robot 5 (0, 25)   | (50,0)               | [3, 7]                | 20  | 6  | 80 m           | 2/3 \( \pi \)    |

Table 8. The numerical simulation results by different methods for the first multi-robot path planning scenario.

| Methods   | Robot 1 | Robot 2 | Simultaneous arrival time (s) | Total fitness value (m) | Total computation time (s) |
|-----------|---------|---------|-------------------------------|-------------------------|----------------------------|
|           | PL (m)  | MV (m/s)| PL (m)                       | MV (m/s)                |                           |
| SPSO 2011 | 129.34  | 5.69    | 136.34                       | 6.00                    | 22.72                     |
| CIGA      | 119.24  | 5.51    | 129.81                       | 6.00                    | 21.63                     |
| FACPSO    | 119.40  | 5.64    | 127.08                       | 6.00                    | 21.18                     |
| SAEGBPSO  | 118.03  | 5.79    | 122.22                       | 6.00                    | 20.37                     |

Bold values denotes the best result obtained with regard to each metric.

Table 9. The numerical simulation results by different methods for the second multi-robot path planning scenario.

| Methods   | Robot 1 | Robot 2 | Robot 3 | Robot 4 | Robot 5 | Simultaneous arrival time (s) | Total fitness value (m) | Total computation time (s) |
|-----------|---------|---------|---------|---------|---------|-------------------------------|-------------------------|----------------------------|
|           | PL (m)  | MV (m/s)| PL (m)  | MV (m/s)| PL (m)  | MV (m/s)                      | PL (m)                 |                           |
| SPSO 2011 | 63.86   | 5.60    | 52.93   | 4.64    | 52.70   | 4.62                          | 55.32                  | 4.84                      |
| CIGA      | 69.84   | 7.00    | 53.71   | 5.38    | 53.66   | 5.38                          | 56.02                  | 5.61                      |
| FACPSO    | 63.84   | 7.00    | 52.76   | 5.78    | 52.51   | 5.76                          | 57.22                  | 6.27                      |
| SAEGBPSO  | 63.44   | 7.00    | 52.81   | 5.82    | 52.49   | 5.79                          | 54.10                  | 5.97                      |

PL: path length of each robot; MV: moving velocity of each robot; CIGA: coevolutionary improved genetic algorithm; SPSO: standard particle swarm optimization; FACPSO: fitness-scaling adaptive chaotic particle swarm optimization; SAEGBPSO: self-adaptive evolutionary game-based particle swarm optimization.

Bold values denotes the best result obtained with regard to each metric.

Figure 10. The generated paths, cost curves, and arrival time of different robots searched by SPSO 2011 for the first multi-robot planning case. (a) Generated paths. (b) Cost curves. (c) Arrival time of robots to destination. SPSO: standard particle swarm optimization.
and even if the proposed method is ranked the second in the computation time among the four methods, this proposed method outperforms the other three methods in the path optimality, as confirmed in Tables 8 and 9. Also, the difference of the computation time between our proposed method and SPSO 2011 is less than 6 s in each simulation scenario, as presented in Tables 8 and 9, which could be neglected due to the considered path planning problem is performed off-line. Taking all these concerns stated here into account, it allows us to conclude that the proposed method can be considered as a vital alternative in the multi-robot path planning problem.

Here, one may be confused by the obtainment of the simultaneous arrival time of each robot to the destination presented in Tables 8 and 9. To interpret by how the simulation arrival time of each robot is obtained, only the proposed method for the first multi-robot numerical simulation is set to be an example. As presented in Table 8, the path
lengths of Robot 1 and Robot 2 generated by the proposed method are, respectively, 118.03 m and 122.22 m in the first numerical simulation. In such a case, when Robot 2 and Robot 1, respectively, move at speeds of 6 m/s and 5.79 m/s, these two robots can simultaneously arrive at the destination location in 20.37 s, as reported in Table 8.

**Conclusions**

In this study, a novel-PSO-based method is proposed for solving the multi-robot path planning problem. To enhance the performance of the optimizer, a novel SAEGBPSo algorithm is first proposed via integrating SPSO 2011 and
the evolutionary stable strategy of EGT. Aiming at addressing the stagnation issue of PSO, particles in SAEGBPSO update their movement information according to the moving rules defined in SPSO 2011. Subsequently, for well-balancing the global and local search capabilities of particles, a novel self-adaptive strategy is proposed to update the three main control parameters of particles in SAEGBPSO based on the EGT and the iteration number of the algorithm. Moreover, the convergence of the proposed SAEGBPSO is analytically studied and a parameter selection principle which sufficiently guarantees the convergence of SAEGBPSO is presented in this article.

Leveraging the proposed SAEGBPSO and a coevolutionary strategy, this article develops of a coevolution-based SAEGBPSO for the multi-robot path planning problem. Finally, the performance of the proposed approach is validated through different numerical simulations both in single-robot and in multi-robot path planning problems. The simulation results confirm that the proposed method is highly competitive in terms of the path optimality. Moreover, the computation time of the proposed method is comparable with those of the other approaches compared in this article. Therefore, the proposed method can be regarded as an effective alternative in robot path planning.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is financially supported by the National Science Foundation of China under Grant No. 61903286.

References
1. Das PK, Behera HS, and Panigrahi BK. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. Swarm Evol Comput 2016; 28: 14–28.
2. Patle BK, Parhi DRK, Jagadeesh A, et al. Matrix-binary codes based genetic algorithm for path planning of mobile robot. Comput Electr Eng 2018; 67: 708–728.
3. Tang B, Zhu Z, and Luo J. Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning. Int J Adv Robot Syst 2016; 13(3): 1–17.
4. Ryan MRK. Exploiting subgraph structure in multi-robot path planning. J Artif Intell Res 2008; 31: 497–542.
5. Yu J and Rus D. An effective algorithmic framework for near optimal multi-robot path planning. In: Robotics research (eds A. Bicchi and W. Burgard), Berlin, Heidelberg, 27 July 2018, pp. 495–511. Cham: Springer.
6. Solana Y, Furci M, Cortés J, et al. Multi-robot path planning with maintenance of generalized connectivity. In: 2017 international symposium on multi-robot and multi-agent systems (MRS), Los Angeles, CA, USA, 4–5 December 2017, pp. 63–70. Los Angeles: IEEE.
7. Tallamraju R, Rajappa S, Black MJ, et al. Decentralized MPC based obstacle avoidance for multi-robot target tracking scenarios. In: 2018 IEEE international symposium on safety, security, and rescue robotics (SSRR), Philadelphia, USA, 6–8 August 2018, pp. 1–8. Philadelphia: IEEE.
8. Qu H, Xing K, and Alexander T. An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. Neurocomputing 2013; 120: 509–517.
9. Nazarahari M, Khanmirza E, and Doostie S. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. Expert Syst Appl 2019; 115: 106–120.
10. Das PK, Behera HS, and Panigrahi BK. Intelligent-based multi-robot path planning inspired by improved classical q-learning and improved particle swarm optimization with perturbed velocity. Eng Sci Technol Int J 2016; 19(1): 651–669.
11. Tharwat A, Elhoseny M, Hassanien AE, et al. Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm. Cluster Comput 2019; 22(2): 4745–4766.

Figure 17. The generated paths, cost curves, and arrival time of different robots searched by SAEGBPSO for the second multi-robot planning case. (a) Generated paths. (b) Cost curves. (c) Arrival time of robots to destination. SAEGBPSO: self-adaptive evolutionary game-based particle swarm optimization.
12. Mac TT, Copot C, Tran DT, et al. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. Appl Soft Comput 2017; 59: 68–76.
13. Akbari R and Ziarati K. A rank based particle swarm optimization algorithm with dynamic adaptation. J Comput Appl Math 2011; 235(8): 2694–2714.
14. Li C, Zhou J, Kou P, et al. A novel chaotic particle swarm optimization based fuzzy clustering algorithm. Neurocomputing 2012; 83: 98–109.
15. Chauhan P, Deep K, and Pant M. Novel inertia weight strategies for particle swarm optimization. Memetic Comput 2013; 5(3): 229–251.
16. Han F and Liu Q. A diversity-guided hybrid particle swarm optimization based on gradient search. Neurocomputing 2014; 137: 234–240.
17. Vitorino LN, Ribeiro SF, and Bastos-Filho CJA. A mechanism based on artificial bee colony to generate diversity in particle swarm optimization. Neurocomputing 2015; 148: 39–45.
18. Tang B, Zhu Z, and Luo J. A framework for constrained optimization problems based on a modified particle swarm optimization. Math Probl Eng 2016; 2016: 1–19.
19. Cédric L, Hyo-Sang S, Patrick S, et al. Convergence proof of an enhanced particle swarm optimisation method integrated with evolutionary game theory. Inform Sci 2016; 346: 389–411.
20. Clerc M. Beyond Standard Particle Swarm Optimisation. Int J of Swarm Intel Res 2010; 1(4): 46–61.
21. Gale JS and Eaves LJ. Logic of animal conflict. Nature 1975; 254(5499): 463–464.
22. Taylor PD and Jonker L.B. Evolutionary stable strategies and game dynamics. Math Biosci 1978; 40(1–2): 145–156.
23. Kala R. Multi-robot path planning using co-evolutionary genetic programming. Expert Syst Appl 2012; 39(3): 3817–3831.
24. Dun-wei G, Jian-hua Z, and Yong Z. Multi-objective particle swarm optimization for robot path planning in environment with danger sources. J Comput 2011; 6(8): 1554–1561.
25. Bonyadi MR and Michalewicz Z. Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm. IEEE Trans Evolut Comput 2015; 20(3): 370–385.
26. Besada-Portas E, Torre LDL, Jesus M, et al. Evolutionary trajectory planner for multiple UAVs in realistic scenarios. IEEE Trans Robot 2010; 26(4): 619–634.
27. Das PK, Behera HS, Das S, et al. A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment. Neurocomputing 2016; 207: 735–753.
28. Zheng C, Li L, Xu F, et al. Evolutionary route planner for unmanned air vehicles. IEEE Trans Robot 2005; 21(4): 609–620.
29. Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 2002; 6(2): 182–197.
30. Wang Z, Li S, and Sang Z. A new constraint handling method based on the modified Alopex-based evolutionary algorithm. Comput Ind Eng 2014; 73: 41–50.
31. Zhang Y, Wu L, and Wang S. UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization. Math Probl Eng 2013, 2013: 1–9.