• 通信与信息工程 •

基于群体智能算法的无人机蜂群拓扑构型方法

杨彦祥[1,2]，张翔引[1,2*]，李　波[1]，秦开宇[1,2]

(1. 电子科技大学航空航天学院　成都　611731；2. 飞行器集群智能感知与协同控制四川省重点实验室　成都　611731)

【摘要】面向特定任务场景，无人机蜂群需要通过拓扑构型自主形成特定的拓扑形状以实现高效群体协同机制。拓扑构型通常包含从初始拓扑到目标拓扑的最佳映射和最优拓扑构型位置这两个问题，它们相互影响并直接关系到无人机蜂群的全局能量消耗。基于全局能耗最小化为目标的无人机蜂群拓扑构型联合优化模型，建立了基于群体智能算法的一般化求解框架，给出了基于灰狼优化算法 (GWO)、均衡优化算法 (EO) 和穷富优化算法 (PRO) 的具体求解方法，并讨论了基于群体智能算法求解优化模型的加速收敛策略。仿真结果证明了该无人机蜂群拓扑构型方法的有效性。在典型拓扑构型场景下，该优化方法在 8 次迭代内即可实现算法收敛。

关　键　词　全局能耗；　联合优化；　群体智能；　拓扑构型；　无人机蜂群

中图分类号　TN915; V279　　**文献标志码**　A　　doi:10.12178/1001-0548.2022091

UAV Swarm Topology Shaping Method Based on Swarm Intelligence Algorithm

YANG Yanxiang[1,2], ZHANG Xiangyin[1,2*], LI Bo[1], and QIN Kaiyu[1,2]

(1. School of Aeronautics and Astronautics, University of Electronic Science and Technology of China　Chengdu　611731;

2. Aircraft Swarm Intelligent Sensing and Cooperative Control Key Laboratory of Sichuan Province　Chengdu　611731)

Abstract　With the advantages of high performance, strong robustness, and large service range, unmanned aerial vehicle (UAV) swarm systems have been widely used in military and civil scenarios. UAV swarms need to form specific topology shapes autonomously through topology shaping to achieve efficient swarm collaboration mechanisms for specific mission scenarios. Topology shaping typically involves two aspects: the optimal mapping from the initial topology to the target topology and the optimal topology shaping location. These two aspects affect each other and are directly related to the global energy consumption of the UAV swarm. Based on the joint optimization model of UAV swarm topology shaping with global energy consumption minimization as the goal, a generalized solution framework based on the swarm intelligence algorithm is firstly established, and specific solution methods based on the gray wolf optimizer algorithm (GWO), the equilibrium optimizer algorithm (EO) and the poor and rich optimization algorithm (PRO) are proposed. Then the convergence acceleration strategy for solving the optimization model based on the swarm intelligence algorithm is proposed. Simulation results show the effectiveness of the proposed UAV swarm topology shaping method and indicate that the proposed optimization method can achieve convergence within 8 iterations in a typical topology shaping scenario.

Key words　global energy consumption;　joint optimization;　swarm intelligence;　topology shaping;　UAV swarm

相比于单无人机平台，无人机蜂群具备效率更高、鲁棒性更好、服务范围更大等优势，在军用和民用场景中得到了广泛的应用[1]。对于不同的应用场景，无人机蜂群需要形成特定的拓扑形状，如在基于无人机的无线通信方案中，空中无线网络的性能取决于无人机的三维位置 (被称为拓扑形状)[2]。是否形成任务所需的拓扑形状决定了无人机蜂群实现指定任务的能力和效率[3]。此外，无人机蜂群需要具备动态调整拓扑形状的能力，以适应任务变化、节点损失、区域干扰等问题。无人机蜂群拓扑

收稿日期：2022－03－31；修回日期：2022－04－27

基金项目：部级基金 (61403120404)

作者简介：杨彦祥 (1986－)，男，博士生，主要从事无人机蜂群拓扑构型、集群智能等方面的研究.

*通信作者：张翔引，E-mail：zhangzxy@uestc.edu.cn

构型，即确定蜂群中每个无人机目标位置的过程，被认为是无人机蜂群完成指定任务的最基本的程序之一，它将引导无人机蜂群自主形成所需的拓扑形状。因此，无人机蜂群的拓扑构型对无人机蜂群整体效能至关重要。另外，无人机对能耗极为敏感，为了加强蜂群系统的稳健性和生存能力，在拓扑构型过程中需要充分考虑蜂群系统全局能耗。

在拓扑构型过程中需要解决两个主要的问题[4]。一是在全局能耗最小条件下确定无人机蜂群各节点从初始位置到目标位置的映射关系。文献 [5] 通过逐个将蜂群中的无人机分配给距离最近的目标位置以获得从初始位置到目标位置的映射关系，但是这种映射关系对应的全局能耗不一定最小；文献 [6-8] 采用了如匈牙利算法、拍卖算法、Jonker-Volgenant 等动态线性分配算法，以获得最小全局能耗条件下的从初始拓扑位置到目标拓扑的映射关系。但是这些方法没有考虑到拓扑构型面临的第二个问题，即拓扑构型位置对能源消耗的影响。

在无人机蜂群拓扑构型过程中，映射关系和拓扑构型位置具有潜在的相互影响的特点，且两者都会影响无人机蜂群的拓扑构型的全局能耗。因此，必须对映射关系和拓扑构型位置进行联合建模以优化无人机蜂群的整体能耗。文献 [9] 将映射关系和拓扑构型位置联合建模，并采用匈牙利算法和粒子群优化算法 (particle swarm optimization, PSO) 的混合算法求解这个问题。仿真结果表明，此联合算法通常在 50 次内完成迭代，表现出相对缓慢的收敛性。

本文基于全局能耗最小化目标的无人机蜂群拓扑构型联合优化模型，建立了基于群体智能算法的通用求解框架，并给出了基于灰狼优化算法 (gray wolf optimizer, GWO)、均衡优化算法 (equilibrium optimizer, EO) 和穷人富人优化算法 (poor and rich optimization, PRO) 的具体求解步骤。结合无人机蜂群拓扑构建的应用场景，优化了群体智能算法的初始化阶段以加速算法收敛速度。仿真结果证明了所提无人机蜂群拓扑构型方法的有效性。在典型拓扑构型场景下，所提优化方法可在 8 次迭代内实现算法收敛。

1 系统模型

如图 1 所示，假设由 N 个无人机组成的无人机蜂群分布在三维欧式空间内。$\boldsymbol{x}_i=[a_i,b_i,c_i]$表示第 i 个无人机在初始拓扑中的位置 (图 1 中五边形)，$i=1,2,\cdots,N$；$\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N]^{\mathrm{T}}\in\mathbb{R}^{N\times3}$表示初始拓扑坐标矩阵，$\boldsymbol{z}_j=[a_j,b_j,c_j]$表示相同坐标系下目标拓扑中的第 j 个坐标位置 (图 1 中圆点)，$j=1,2,\cdots,N$；$\boldsymbol{Z}=[\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N]^{\mathrm{T}}\in\mathbb{R}^{N\times3}$表示目标拓扑坐标矩阵，目标拓扑形状如图 1 中实线表示。令 $\boldsymbol{M}(k)=[m(k)_{ij}]_{N\times N}$表示从初始拓扑到目标拓扑的映射关系 (图 1 中带箭头虚线)，当初始拓扑中的节点 i 被指派给目标位置 j 时，$m(k)_{ij}=1$，否则 $m(k)_{ij}=0$。假设$\boldsymbol{M}^{\mathrm{o}}(k)=[m^{\mathrm{o}}(k)_{ij}]_{N\times N}$表示当前$k$下全局能耗最小的最优映射关系。

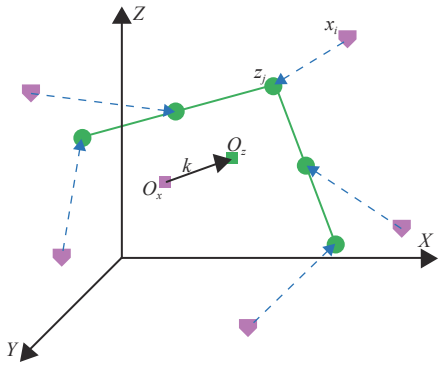

图 1 无人机蜂群拓扑构型示意图

假设$\boldsymbol{O}_x$和$\boldsymbol{O}_z$分别代表初始拓扑和目标拓扑的几何中心 (图 1 中 2 个正方形)，$\boldsymbol{k}\in\mathbb{R}^{1\times3}$表示$\boldsymbol{X}$和$\boldsymbol{Z}$的相对位置向量，那么：

$$\boldsymbol{k}=\boldsymbol{O}_z-\boldsymbol{O}_x \tag{1}$$

最优拓扑构型位置$\boldsymbol{O}^{\mathrm{o}}{}_z$可以表示为$\boldsymbol{O}^{\mathrm{o}}{}_z=\boldsymbol{k}^{\mathrm{o}}+\boldsymbol{O}_x$，其中$\boldsymbol{k}^{\mathrm{o}}$是待求的相对位置向量。

拓扑构型全局能耗C_g可以表示为蜂群中所有无人机从初始拓扑位置到目标拓扑位置的欧氏距离之和[6-8]。假设c_{ij}表示从初始拓扑中第 i 个节点到目标拓扑第 j 个位置的欧式距离，那么：

$$c_{ij}=\|\boldsymbol{x}_i-\boldsymbol{z}_j\|=\sqrt{(a_i-a_j)^2+(b_i-b_j)^2+(c_i-c_j)^2} \tag{2}$$

假设将目标拓扑平移向量$\boldsymbol{k}'=[a_k,b_k,c_k]\in\mathbb{R}^{1\times3}$，那么$\boldsymbol{Z}_k=\boldsymbol{Z}+\boldsymbol{k}'$，所有的目标拓扑位置$\boldsymbol{z}_j$会被平移至$\boldsymbol{z}_j+\boldsymbol{k}'$。此时平移后第 i 个节点和第 j 个目标位置对应的欧氏距离可以表示为：

$$c_{ij}=\|\boldsymbol{x}_i-\boldsymbol{z}_j-\boldsymbol{k}'\|=\sqrt{(a_i-a_j-a_k)^2+(b_i-b_j-b_k)^2+(c_i-c_j-c_k)^2} \tag{3}$$

从式 (3) 中可以看出，c_{ij}与目标拓扑平移向量$\boldsymbol{k}'$相关，而目标拓扑平移$\boldsymbol{k}'$后$\boldsymbol{X}$和$\boldsymbol{Z}$新的相对位置向量$\boldsymbol{k}$会变为$\boldsymbol{k}+\boldsymbol{k}'$，即在原来的相对位置向量上做了修正，因此可认为$c_{ij}$是与$\boldsymbol{k}$相关的函数，表示为

$c(k)_{ij}$，并通过迭代$\boldsymbol{k}$来求得最优拓扑构型位置。令$\boldsymbol{C}$表示从初始拓扑到目标拓扑的代价矩阵，即$\boldsymbol{C}=[c(k)_{ij}]_{N\times N}$。根据线性分配算法可知，从初始拓扑到目标拓扑的映射关系和通过线性分配算法求得的全局能耗会随$c(k)_{ij}$变化。因此，全局能耗C_g也和$\boldsymbol{k}$相关，表示为：

$$C(k)_g=\sum_{i=1}^{N}\sum_{j=1}^{N}c(k)_{ij}m^{\mathrm{o}}(k)_{ij} \tag{4}$$

在拓扑构型场景下，初始拓扑$\boldsymbol{X}$和目标拓扑初始坐标$\boldsymbol{Z}$一旦确定，那么就可以得到$\boldsymbol{k}$和对应的$c(k)_{ij}$，此时最佳映射关系$\boldsymbol{M}^{\mathrm{o}}(k)$及全局能耗$C(k)_g$可以通过线性分配算法求得，从式 (4) 可以看出，相对位置向量$\boldsymbol{k}$(拓扑构型位置) 和最佳映射关系相互影响，并且都会影响全局能耗。

本文的目标是在最小化全局能耗的前提下求得最优拓扑构型映射关系$\boldsymbol{M}^{\mathrm{o}}(k)$和最优拓扑构型位置$\boldsymbol{O}^{\mathrm{o}}{}_z$，使式 (4) 中$C(k)_g$在有限的三维空间中最小，即：

$$\min_{k\in\mathbb{R}^{1\times 3}}C(k)_g=\min_{k\in\mathbb{R}^{1\times 3}}\sum_{i=1}^{N}\sum_{j=1}^{N}c(k)_{ij}m^{\mathrm{o}}(k)_{ij} \tag{5}$$

基于给出的联合优化模型，本文调用群体智能算法进行求解。

2　基于群体智能优化的拓扑构型方法

具有简单、灵活、无微分机制和避免局部最优[10] 等优点的群体智能算法特别适合该联合优化求解模型。集群智能算法[10-13] 的主要思想是模仿自然界中动物或者物体在群体中的行为，这类算法的主要特点是使用随机方式来寻找最优解，并在求最优解的过程中共享所有个体的最优解信息。另外这类算法的求解过程与被优化的问题相互独立，适合寻找特定优化问题最优解，而不需要关注问题搜索空间的非线性类型及其约束。

通过调用群体智能算法求解这个联合优化模型，分为初始化和群体智能算法迭代两个阶段，如图 2 所示。为了公平比较，根据文献 [6-8]，确定相对位置向量$\boldsymbol{k}$后，最佳映射关系$\boldsymbol{M}^{\mathrm{o}}(k)$及对应的全局最小能耗$C(k)_g$可通过线性分配算法求得，通过线性分配算法求得匹配关系和全局最小能耗的具体过程不做详细讨论。

1) 初始化阶段

输入拓扑构型及群体智能算法相关参数，初始化算法的最大迭代次数 Max_iter、搜寻节点数 N、搜寻区间上限 ub、搜寻区间下限 lb、初始拓扑坐标矩阵$\boldsymbol{X}$、目标拓扑坐标矩阵$\boldsymbol{Z}$，初始化群体智能算法相关参数。

2) 群体智能算法迭代阶段

根据拓扑构型参数及群体智能算法初始化参数或所选群体智能算法位置，更新每个搜索点的相对位置向量$\boldsymbol{k}$。根据式 (3) 计算当前$\boldsymbol{k}$下初始拓扑到目标拓扑的代价矩阵$\boldsymbol{C}$。根据代价矩阵$\boldsymbol{C}$及线性分配算法求得当前$\boldsymbol{k}$对应的映射关系和全局能耗。记录本次迭代中最小全局能耗及对应的映射关系。

重复以上步骤直到满足结束条件。

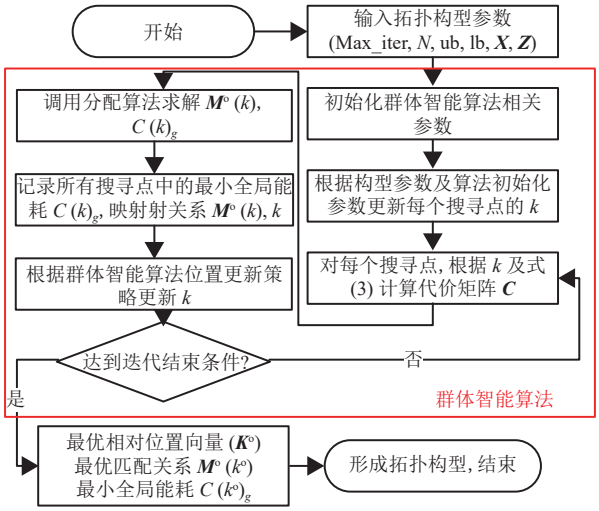

图 2　群体智能算法拓扑构型流程图

本文提出的优化模型求解框架中，所选群体智能算法中各个搜寻点的位置对应于相对位置向量$\boldsymbol{k}$。另外，图 2 中的群体智能算法并不局限于文献 [10-13] 中的 4 种算法，现有的群体智能算法均可适配于本框架。所有群体智能算法需要的通用参数包括算法最大迭代次数、搜寻节点数、搜寻区间上限和下限以及用于计算匹配关系和最小全局能耗的初始拓扑位置和最初的目标拓扑位置。最初的目标拓扑位置是指根据初始拓扑坐标矩阵和目标拓扑坐标矩阵选取一个初始的相对位置向量$\boldsymbol{k}$。

本文选取灰狼优化算法 (GWO)(伪代码见算法 1)、均衡优化算法 (EO)(伪代码见算法 2)、穷人富人优化算法 (PRO)(伪代码见算法 3) 来进一步说明基于所提求解框架的具体求解步骤。

算法 1　灰狼优化算法 (GWO)

输入：目标函数$C(k)_g$，最大迭代次数 Max_iter

输出：最小能耗及对应的位置X_α

初始化灰狼狼群大小及初始位置$\boldsymbol{X}_i\,(i=1,2,\cdots,N)$；

初始化 $\boldsymbol{a},\boldsymbol{A},\boldsymbol{C}$；

根据$\boldsymbol{X}_i$计算目标函数值$C(\boldsymbol{X}_i)_g$；

$\boldsymbol{X}_\alpha=\alpha$狼 (最佳) 能耗值对应位置；

$\boldsymbol{X}_\beta = \beta$狼 (次佳) 能耗值对应位置；
$\boldsymbol{X}_\gamma = \gamma$狼 (次次佳) 能耗值对应位置；
while iter < Max_iter do
for i=1:N do
通过$\boldsymbol{X}(t+1) = \dfrac{\boldsymbol{X}_\alpha + \boldsymbol{X}_\beta + \boldsymbol{X}_\gamma}{3}$更新当前狼位置；
end
更新向量$\boldsymbol{a}, \boldsymbol{A}, \boldsymbol{C}$；
计算所有狼所在位置的目标函数值$C(\boldsymbol{X}_i)_g$；
更新$\boldsymbol{X}_\alpha$，$\boldsymbol{X}_\beta$，$\boldsymbol{X}_\gamma$；
iter = iter + 1；
end
return $\boldsymbol{X}_\alpha$，$C(\boldsymbol{X}_\alpha)_g$

算法 2　均衡优化算法 (EO)
输入：目标函数$C(k)_g$，最大迭代次数 Max_iter
输出：最小能耗 $C(\boldsymbol{C}_i)_g$及对应位置$\boldsymbol{C}_i$
初始化粒子数目 N，参数$a_1 = 2$，$a_2 = 1$，GP = 0.5; 初始化粒子位置$C_i^{\text{initial}} = C_{\min} + \text{rand}_i(C_{\max} - C_{\min})$，$i = 1, 2, \cdots, N$；
while Iter < Max_iter do
for i=1:N do
计算第 i 个粒子的目标函数$C(\boldsymbol{C}_i)_g$的值；
更新 4 个最小$C(\boldsymbol{C}_i)_g$值及对应的位置$\boldsymbol{C}_i$；
end
更新平均状态$\boldsymbol{C}_{\text{ave}} = (\boldsymbol{C}_{\text{eq1}} + \boldsymbol{C}_{\text{eq2}} + \boldsymbol{C}_{\text{eq3}} + \boldsymbol{C}_{\text{eq4}})/4$，建立位置更新向量可选池$\boldsymbol{C}_{\text{eq,pool}} = \left\{\boldsymbol{C}_{\text{eq(1)}}, \boldsymbol{C}_{\text{eq(2)}}, \boldsymbol{C}_{\text{eq(3)}}, \boldsymbol{C}_{\text{eq(4)}}, \boldsymbol{C}_{\text{eq(ave)}}\right\}$；
令$t = (1 - \dfrac{\text{Iter}}{\text{Max_iter}})^{\left(a_2 \frac{\text{Iter}}{\text{Max_iter}}\right)}$；
for i = 1:N do
从均衡状态池随机选 1 个候选向量$\boldsymbol{C}_{eq}$；
生成随机向量$\boldsymbol{\lambda}$，$\boldsymbol{r}$；
构建$\boldsymbol{F} = a_1 \text{sign}(\boldsymbol{r} - 0.5)[\text{e}^{-\lambda t} - 1]$；
构建$\mathbf{GCP} = \begin{cases} 0.5 r_1 & r_2 \geqslant \text{GP} \\ 0 & r_2 < \text{GP} \end{cases}$；
构建$\boldsymbol{G}_0 = \mathbf{GCP}(\boldsymbol{C}_{\text{eq}} - \boldsymbol{\lambda}\boldsymbol{C})$；
构建$\boldsymbol{G} = \boldsymbol{G}_0 \boldsymbol{F}$；
更新$\boldsymbol{C} = \boldsymbol{C}_{\text{eq}} + (\boldsymbol{C} - \boldsymbol{C}_{\text{eq}})\boldsymbol{F} + \dfrac{\boldsymbol{G}}{\lambda V}(1 - \boldsymbol{F})$；
end
iter = iter + 1；
end
return $C(\boldsymbol{C}_i)_g$，$\boldsymbol{C}_i$

算法 3　穷人富人优化算法 (PRO)
输入：目标函数$C(k)_g$，最大迭代次数 Max_iter，变异概率 Pmut
输出：最小能耗$C(k)_g$及对应的位置X_α
初始化穷人富人群体大小及初始位置$\mathbf{XR}_i\,(i = 1, 2, \cdots, N)$，$\mathbf{XP}_i\,(i = 1, 2, \cdots, N)$；
对每一个穷人和富人计算目标函数值$C(\mathbf{XP}_i)_g$，$C(\mathbf{XR}_i)_g$，根据目标函数值大小区分穷人群体和富人群体；
while iter < Max_iter do
根据目标函数值区分穷人和富人群体，选择富人最小值$\boldsymbol{X}_{\text{rich,best}}^{\text{old}}$、平均值$\boldsymbol{X}_{\text{rich,mean}}^{\text{old}}$、最大值$\boldsymbol{X}_{\text{rich,worst}}^{\text{old}}$；
for i=1:N do % 富人群体
更新位置$\boldsymbol{X}_{\text{rich},i}^{\text{new}} = \boldsymbol{X}_{\text{rich},i}^{\text{old}} + r\left[\boldsymbol{X}_{\text{rich,i}}^{\text{old}} - \boldsymbol{X}_{\text{poor,best}}^{\text{old}}\right]$；
if rand < P_{mut} then
$\boldsymbol{X}_{\text{rich},i}^{\text{new}} = \boldsymbol{X}_{\text{rich},i}^{\text{new}} + \text{rand}n$；
end
计算新位置的目标函数值；
end
for i=1:N do % 穷人群体
计算模式值Pattern=$\dfrac{\boldsymbol{X}_{\text{rich,best}}^{\text{old}} + \boldsymbol{X}_{\text{rich,mean}}^{\text{old}} \boldsymbol{X}_{\text{rich,worst}}^{\text{old}}}{3}$；
更新位置$\boldsymbol{X}_{\text{poor},i}^{\text{new}} = \boldsymbol{X}_{\text{poor},i}^{\text{old}} + [r(\text{Pattern}) - \boldsymbol{X}_{\text{poor},i}^{\text{old}}]$；
if rand < P_{mut} then
$\boldsymbol{X}_{\text{poor},i}^{\text{new}} = \boldsymbol{X}_{\text{poor},i}^{\text{new}} + \text{rand}n$；
end
计算新位置的目标函数值；
end
选出最小$C(\mathbf{XP}_i)_g$及对应位置$\mathbf{XP}_i$；
iter = iter + 1；
end
return $\mathbf{XP}_i$，$C(\mathbf{XP}_i)_g$

在群体智能算法中，需要初始化各个搜寻点的初始位置。如在 PSO 算法初始化过程中，需要先随机生成所有粒子的初始位置，一般是通过随机函数生成随机的位置。

值得注意的是，初始化过程会影响群体智能算法的收敛速度。在本文的拓扑构型场景下，各个群体智能算法的初始位置可被进一步优化，加速算法收敛速度。从直观的角度来看，目标拓扑和初始拓扑节点重叠的越多，全局能耗就越小。因此在初始化群体智能算法各搜寻点的初始位置时，可以初始拓扑坐标作为各个搜寻点的初始位置，而不是按照原始算法中的随机生成初始位置。另外，在本场景中，目标拓扑位置与初始拓扑位置越接近全局能耗

越小，因此可在初始化阶段假设初始拓扑和目标拓扑坐标原点重合，即取$k=0$，并取其中的最小值作为所选群体智能算法的初始最小值。

3　仿真及结果分析

本文通过 MATLAB 仿真验证以全局能耗最小化为目标的无人机蜂群拓扑构建联合优化模型，以及本文提出的可以灵活调度不同群体智能算法实现模型优化求解的框架和群体智能算法加速收敛的策略。仿真所运行的 MATLAB 版本为 R2021b 64 位，电脑配置为 12 th Gen Intel(R) Core(TM) i9 CPU，128 GB RAM，Windows 11 Pro。

在仿真过程中，设定算法最大迭代次数 Max_iter=100，每次迭代中搜寻节点数 N=32，搜寻区间上限 ub=800，搜寻区间下限 lb=−800，初始拓扑坐标矩阵$\boldsymbol{X}$由随机分布在 (−400, 400) 的三维空间中的随机点组成，目标拓扑坐标矩阵$\boldsymbol{Z}$为边长为 400 的立方体，初始拓扑和目标拓扑初始位置的原点重合。PSO 算法初始设置为 $c1$=$c2$=2，w=0.9；EO 算法初始值设置为 $a1$=2，$a2$=1，GP=0.5，V=1；PRO 算法初始值设置为 Pmut=0.06。

3.1　拓扑构型验证

本节选取 PSO、EO、PRO、GWO 这 4 种群体智能算法求解联合优化模型。仿真结果如图 3 所示，其中星号代表节点的初始位置。

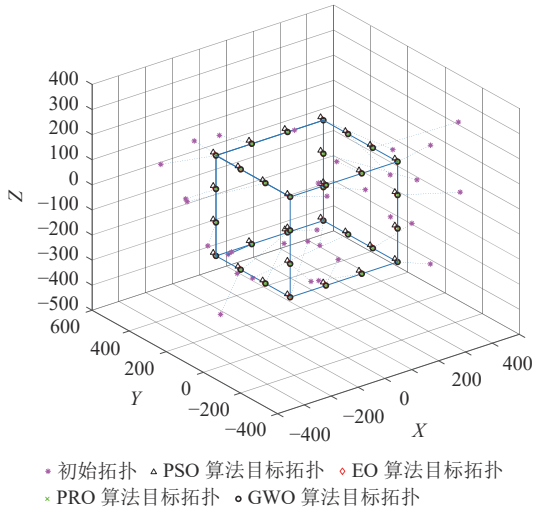

图 3　立方体形状目标拓扑构建仿真结果

图 3 中，实线勾勒出的是 GWO 算法求得的目标拓扑构型位置上的拓扑外形。虚线表示初始拓扑到 GWO 算法得到的拓扑构型位置的映射关系。仿真结果表明，通过所提方法可以实现期望的最佳拓扑构型以及最优拓扑构型位置，PRO 和 GWO 最终收敛结果一致，二者得到的全局最小能耗低于 PSO 算法和 EO 算法相应值。

3.2　群体智能算法加速收敛策略验证

为了验证初始位置对收敛速度的影响，根据前面分析，本文进行了 2 组 100 次独立仿真，2 组仿真分别设置了$\boldsymbol{k}\neq\boldsymbol{0}$(见图 4) 和$\boldsymbol{k}=\boldsymbol{0}$(见图 5) 以观察各个群体智能算法的收敛速度。仿真结果表明，在$\boldsymbol{k}\neq\boldsymbol{0}$时，GWO 算法、PRO 算法和 PSO 算法平均在 35 次内收敛，而 EO 算法平均需要 75 次才能收敛。在$\boldsymbol{k}=\boldsymbol{0}$条件下，EO 算法、GWO 算法和 PRO 算法均在可以在 8 次迭代内收敛，而 PSO 算法依然需要 30 余次迭代才能收敛。

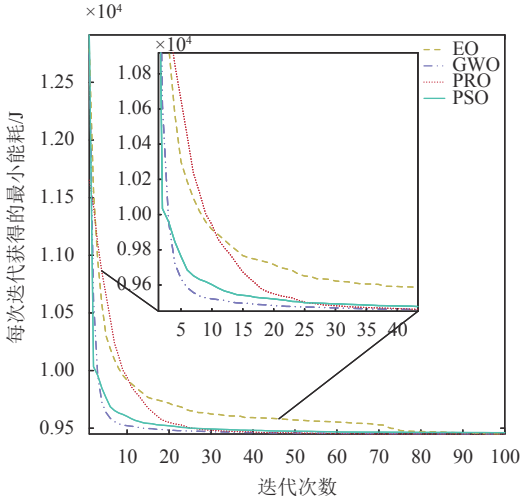

图 4　$\boldsymbol{k}\neq\boldsymbol{0}$时各群体算法收敛速度

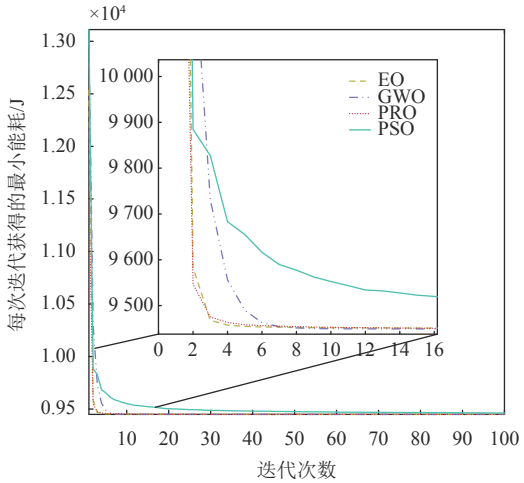

图 5　$\boldsymbol{k}=\boldsymbol{0}$ 时各群体算法收敛速度

为进一步研究各加速收敛策略的有效性，将 GWO 算法中各狼的初始位置设置为随机生成 (设置 1)，各狼的初始位置选取初始拓扑位置且目标拓扑与初始拓扑原点重合，即$\boldsymbol{k}=\boldsymbol{0}$(设置 2)，并进行 100 次独立仿真以对比各狼初始位置对收敛速度的影响，其余仿真参数如前所述，仿真结果如图 6 所示。仿真结果表明，目标拓扑的初始位置选取对收

敛影响较大，当目标拓扑原点与初始拓扑原点不重合时，算法经过 25 次迭代后才能收敛，$k=0$ 算法可以在 8 次迭代内收敛，所提的加速收敛策略提高了群体智能算法收敛速度。

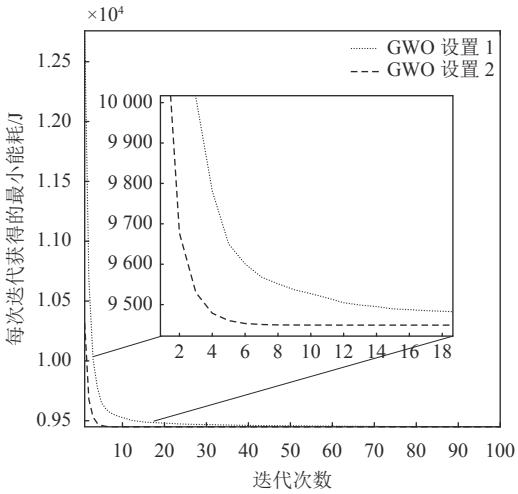

图 6 GWO 算法不同初值及位置更新收敛对比

3.3 群体智能算法计算时间分析

为了对比上述 4 种群体智能算法的性能，本文进一步分析了各算法 100 次独立仿真的计算时间。图 7 为上述 4 种群体智能优化算法 100 次独立仿真的计算时间箱式图统计。仿真结果表明，EO 算法和 GWO 算法计算时间相对较短，平均计算时间分别是 1.6 s/100 次迭代和 1.7 s/100 次迭代，PRO 平均计算时间为 3.2 s/100 次迭代。

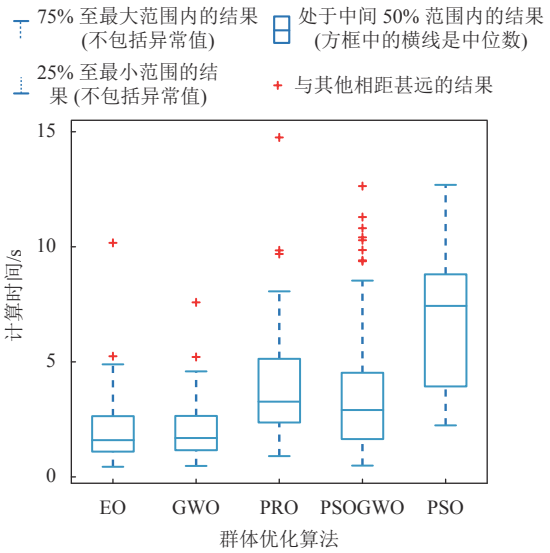

图 7 群体智能算法计算时间对比

4 结 束 语

无人机蜂群已被广泛应用于军用和民用场景，在应用中需要通过拓扑构型来形成特定的拓扑形状以完成指定任务。拓扑构型过程中需要确定最佳映射关系和最优拓扑构型位置。这两个方面相互影响并直接关系到无人机蜂群的全局能量消耗。本文对映射关系和拓扑构型位置进行联合建模，给出了以全局能耗最小化为目标的无人机蜂群拓扑构建联合优化模型，并提出了可以灵活调用不同群体智能算法实现模型优化的求解框架。此外还讨论了群体智能算法加速收敛的策略。

参 考 文 献

[1] SKOROBOGATOV G, BARRADO C, SALAMÍ E. Multiple UAV systems: A survey[J]. Unmanned Systems, 2020, 8(2): 149-169.

[2] ZENG Y, ZHANG R, LIM T J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges[J]. IEEE Communications Magazine, 2016, 54(5): 36-42.

[3] KAMEL M A, YU X, ZHANG Y. Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: A review[J]. Annual Reviews in Control, 2020, 49: 128-144.

[4] MORGAN D, SUBRAMANIAN G P, CHUNG S J, et al. Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming[J]. The International Journal of Robotics Research, 2016, 35(10): 1261-1285.

[5] BRANDÃO A S, SARCINELLI-FILHO M. On the guidance of multiple UAV using a centralized formation control scheme and Delaunay triangulation[J]. Journal of Intelligent & Robotic Systems, 2016, 84(1): 397-413.

[6] HERNÁNDEZ D, CECÍLIA J M, CALAFATE C T, et al. The Kuhn-Munkres algorithm for efficient vertical takeoff of UAV swarms[C]//2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring). [S.l.]: IEEE, 2021: 1-5.

[7] FU X, PAN J, WANG H, et al. A formation maintenance and reconstruction method of UAV swarm based on distributed control[J]. Aerospace Science and Technology, 2020, 104: 105981.

[8] YANG Y, ZHANG X, ZHOU J, et al. A relative coordinate-based topology shaping method for UAV swarm with low computational complexity[J]. Applied Sciences, 2022, 12(5): 2631.

[9] SUI Z, PU Z, YI J. Optimal UAVs formation transformation strategy based on task assignment and particle swarm optimization[C]//2017 IEEE International Conference on Mechatronics and Automation (ICMA). [S. l.]: IEEE, 2017: 1804-1809.

[10] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.

[11] DHAWAN S, GUPTA R, RANA A, et al. Various swarm optimization algorithms: Review, challenges, and opportunities[C]//Soft Computing for Intelligent Systems. Singapore: [s.n.], 2021: 291-301.

[12] FARAMARZI A, HEIDARINEJAD M, STEPHENS B, et al. Equilibrium optimizer: A novel optimization algorithm[J]. Knowledge-Based Systems, 2020, 191: 105190.

[13] MOOSAVI S H S, BARDSIRI V K. Poor and rich optimization algorithm: A new human-based and multi populations algorithm[J]. Engineering Applications of Artificial Intelligence, 2019, 86: 165-181.

编 辑 税 红