自动化学报
ACTA AUTOMATICA SINICA

基于KnCMPSO算法的异构无人机协同多任务分配

王峰 黄子路 韩孟臣 邢立宁 王凌

A Knee Point Based Coevolution Multi-objective Particle Swarm Optimization Algorithm for Heterogeneous UAV Cooperative Multi-task Allocation

WANG Feng, HUANG Zi-Lu, HAN Meng-Chen, XING Li-Ning, WANG Ling

在线阅读 View online: https://doi.org/10.16383/j.aas.c210696

您可能感兴趣的其他文章

基于参考点预测的动态多目标优化算法

Dynamic Multi-objective Optimization Algorithm Based on Reference Point Prediction

自动化学报. 2017, 43(2): 313-320 https://doi.org/10.16383/j.aas.2017.c150811

动态多目标优化进化算法研究进展

Research Progress of Dynamic Multi-objective Optimization Evolutionary Algorithm

自动化学报. 2020, 46(11): 2302-2318 https://doi.org/10.16383/j.aas.c190489

基于非支配排序差异演化的应急资源多目标分配算法

Multi-objective Approach to Emergency Resource Allocation Using None-dominated Sorting Based Differential Evolution

自动化学报. 2017, 43(2): 195-214 https://doi.org/10.16383/j.aas.2017.c160076

燃料电池有轨电车能量管理Pareto多目标优化

Pareto-based Multi-objective Optimization of Energy Management for Fuel Cell Tramway

自动化学报. 2019, 45(12): 2378-2392 https://doi.org/10.16383/j.aas.c190044

LSTM逐层多目标优化及多层概率融合的图像描述

Multi-objective Layer-wise Optimization and Multi-level Probability Fusion for Image Description Generation Using LSTM

自动化学报. 2018, 44(7): 1237-1249 https://doi.org/10.16383/j.aas.2017.c160733

一种基于自适应模糊支配的高维多目标粒子群算法

Many-objective Particle Swarm Optimization Based on Adaptive Fuzzy Dominance

自动化学报. 2018, 44(12): 2278-2289 https://doi.org/10.16383/j.aas.2018.c170573

第 49 卷 第 2 期 自 动 化 学 报 Vol. 49, No. 2
2023 年 2 月 ACTA AUTOMATICA SINICA February, 2023

基于 KnCMPSO 算法的异构无人机协同多任务分配

王 峰[1] 黄子路[1] 韩孟臣[1] 邢立宁[2] 王 凌[3]

摘 要 随着无人机 (Unmanned aerial vehicle, UAV) 技术的广泛应用和执行任务的日益复杂, 无人机多机协同控制面临着新的挑战. 以无人机总飞行距离和任务完成时间为优化目标, 同时考虑异构无人机类型、任务执行时序等多种实际约束, 构建基于多种约束条件的异构无人机协同多任务分配模型. 该模型不仅包含混合变量, 同时还存在多个复杂的约束条件, 因此, 传统的多目标优化算法并不能有效地处理混合变量及对问题空间进行搜索并生成满足多种约束条件的可行解. 为高效求解上述模型, 提出一种基于拐点的协同多目标粒子群优化算法 (Knee point based coevolution multi-objective particle swarm optimization, KnCMPSO), 该算法引入基于拐点的学习策略来更新外部档案集, 在保证收敛性的同时增加种群的多样性, 使算法能搜索到更多可行的任务分配结果; 并基于二进制交叉方法, 引入基于学习的粒子更新策略来提升算法的收敛性及基于区间扰动的局部搜索策略以提升算法的多样性. 最后通过在四组实例上的仿真实验验证了所提算法在求解异构无人机协同多任务分配问题上的有效性.

关键词 无人机多任务分配, 多目标优化, 粒子群算法, 协同进化

引用格式 王峰, 黄子路, 韩孟臣, 邢立宁, 王凌. 基于 KnCMPSO 算法的异构无人机协同多任务分配. 自动化学报, 2023, **49**(2): 399−414

DOI 10.16383/j.aas.c210696

A Knee Point Based Coevolution Multi-objective Particle Swarm Optimization Algorithm for Heterogeneous UAV Cooperative Multi-task Allocation

WANG Feng[1] HUANG Zi-Lu[1] HAN Meng-Chen[1] XING Li-Ning[2] WANG Ling[3]

Abstract With the wide application of unmanned aerial vehicle (UAV) technology and the increasing complexity of UAV tasks, the multi-aircraft cooperative control on UAVs faces new challenges. In this paper, a heterogeneous UAV cooperative multi-task allocation model which takes the UAV total flight distance and task completion time as the optimization objectives is set up. The model includes mixed variables and multiple complex constraints, such as UAV type and task execution time. As a result, traditional multi-objective optimization algorithms cannot effectively search the problem space and generate feasible solutions. In this paper, a knee point based cooperative multi-objective particle swarm optimization algorithm, namely knee point based coevolution multi-objective particle swarm optimization (KnCMPSO), is proposed to solve the above model. In KnCMPSO, a knee point based learning strategy is employed to update the external archive set, which can help get more better solutions. The learning-based particle update strategy is proposed to improve the convergence and the interval disturbance based local search strategy is used to enhance the diversity. The experimental results on four sets of examples show that, the proposed KnCMPSO algorithm can solve the heterogeneous UAVs collaborative multi-task allocation problem more effectively than other existing methods.

Key words Unmanned aerial vehicle multi-task allocation, multi-objective optimization, particle swarm algorithm, coevolution

Citation Wang Feng, Huang Zi-Lu, Han Meng-Chen, Xing Li-Ning, Wang Ling. A knee point based coevolution multi-objective particle swarm optimization algorithm for heterogeneous UAV cooperative multi-task allocation. *Acta Automatica Sinica*, 2023, **49**(2): 399−414

收稿日期 2021-07-22 录用日期 2022-04-12
Manuscript received July 22, 2021; accepted April 12, 2022
国家自然科学基金 (62173258, 61773296), 高等学校全国优秀博士学位论文作者专项资金 (2014-92) 资助
Supported by National Natural Science Foundation of China (62173258, 61773296) and National Excellent Doctoral Dissertation Foundation of China (2014-92)
本文责任编委 贺威
Recommended by Associate Editor HE Wei
1. 武汉大学计算机学院 武汉 430072 2. 国防科技大学系统工程学院 长沙 410073 3. 清华大学自动化系 北京 100084
1. School of Computer Science, Wuhan University, Wuhan 430072 2. College of Systems Engineering, National University of Defense Technology, Changsha 410073 3. Department of Automation, Tsinghua University, Beijing 100084

无人机 (Unmanned aerial vehicle, UAV) 以其体积轻巧、隐身性能好、续航时间长、可回收重复使用等优势逐渐成为空中作战的主角. 相较于有人机, 无人机能够在偏远、危险或者人员不可达的极端环境中, 自主完成一些重要的诸如监测、搜救、侦察、打击等任务[1]. 无人机协同作战是指在实际的战场中, 多种类型的无人机充分发挥各自优势, 协同完成一项军事任务. 早期的无人机协同作战由于作战场景简单、任务类型单一、任务难度较低, 仅通过人工决策就能确定各无人机的作战序列. 然而随着作

战场景逐渐复杂、任务类型逐渐多样化、无人机智能化水平逐渐提高, 仅靠人工决策很难充分挖掘各种无人机的特性并发挥无人机系统的协同作战优势. 现实世界中的无人机往往是异构的, 异构无人机指的是多种类型的无人机 (侦察机、战斗机等)[2], 为了提高异构无人机执行任务的效率, 提升异构无人机系统的自主性, 开展异构无人机协同多任务分配问题研究具有十分重要的意义[3].

为了高效地制定无人机的任务分配计划、科学地控制无人机, 需要对无人机协同多任务分配问题建立合适模型.

在模型的约束条件研究方面, 文献 [4] 充分考虑了无人机执行任务之间的时序和多机协同等约束条件, 将无人机分配问题构建成为协同多任务分配问题模型. 文献 [5] 提出以无人机自身性能约束、协同约束和实际三维复杂环境构建约束的无人机任务分配模型. 然而上述模型中所考虑的各无人机都具有侦察、打击、评估三种能力, 不属于异构无人机. 在实际问题中, 由于不同无人机的能力有差异, 无法用一类无人机完成所有任务. 因此通常选择采用多架异构无人机组成集群来共同完成任务, 解决不同无人机的能力限制. 一些学者进一步对异构无人机协同任务分配模型提出了新的约束条件, 如文献 [6] 考虑无人机资源消耗、完成任务资源需求等约束, 文献 [7] 考虑任务优先级等约束, 文献 [8] 考虑无人机运动学等约束.

在模型的优化目标研究方面, 目前大部分研究工作建立的都是以最小化无人机飞行距离的单目标模型[9], 也有一些学者尝试构建无人机任务分配的多目标优化模型. 如文献 [10] 提出同时考虑无人机任务收益、任务执行时间和任务执行代价的多目标模型, 文献 [11] 以无人机总飞行距离和任务完成时间为优化目标构建了多目标优化模型.

在无人机协同多任务分配模型的求解方面, 近年来也有部分学者进行了深入研究[12], 其中粒子群算法[13] 作为群智能算法中的经典算法, 具有操作简单、收敛能力强等特点, 可以很好地求解无人机协同任务分配问题. 文献 [14] 将多种目标函数进行加权, 利用离散粒子群算法对构建的模型进行求解, 但是采用的交叉变异操作方式较为简单, 求解效率有待进一步提升. 文献 [10] 利用多目标量子粒子群算法对无人机任务分配问题进行求解, 但是算法编码较为复杂, 计算量比较大. 文献 [15] 提出采用改进的量子粒子群算法对无人机任务分配问题进行求解, 但是算法需要额外计算平均历史最优位置而且设定的参数较多. 近年来, 随着异构无人机协同多任务分配模型研究的深入, 一些学者针对异构无人机协同多任务分配问题的求解算法进行了探讨. 如文献 [11] 提出一种基于协同进化的混合变量多目标粒子群优化算法, 但其更新策略过于简单且随机性较大, 随着问题规模的增大, 求解效率将逐渐下降. 文献 [16] 提出改进的多目标量子粒子群算法, 但其重组方式较复杂, 求解效率不高. 文献 [17] 提出改进的遗传算法, 但该算法主要针对单目标优化模型, 无法很好地求解多目标异构无人机任务分配模型.

异构无人机协同多任务分配模型有多个复杂的约束条件, 因此, 如何处理不同类型的约束条件, 以提高算法的搜索效率, 是求解异构无人机协同多任务分配模型的重要问题. 近年来, 一些学者针对不同类型的应用问题, 研究采用不同约束处理方法进行求解. 求解方法大致可以分为基于罚函数 (Prnalty function, PF) 的约束处理方法[18−20]、基于可行性原则的约束处理方法 (Stochastic ranking, SR)[21−22]、基于随机排序的约束处理方法 (Feasibility rule, FR)[23] 和基于多目标的约束处理方法[24] 4 类 . 也有一些学者就约束处理技术理论方面提出了一些新的改进, 如基于分级方式的约束处理准则[25]、基于集成方式的约束处理准则[26] 或将约束转化为新的优化目标[27]. 而在异构无人机协同多任务分配模型中, 存在无人机类型约束、任务执行时序约束及多机协同约束等多种复杂约束, 上述方法无法直接用于求解.

为了使构建的模型更加符合现实场景, 现有对无人机任务分配问题的研究工作均在约束条件或目标函数上对模型做出了改进, 但是大部分研究工作均假设无人机的观测时间和一次弹药消耗都是充足的, 然而在实际作战过程中, 当侦察机一次观测时间过长时, 存在被敌方发现乃至击毙的风险, 战斗机一次发射的弹药数量也是有限的. 因此本文在现有研究工作[11] 的基础上, 进一步考虑无人机单次任务最大观测时间约束和无人机单次任务最大弹药约束, 建立基于多种约束条件的异构无人机协同多任务分配问题模型.

上述模型不仅包含混合变量, 同时还存在多个复杂的约束条件, 问题的求解难度较大, 传统的多目标优化算法并不能有效地求解上述模型. 同其他智能优化算法相比, 粒子群算法具有计算简单、鲁棒性好、搜索能力强且收敛速度快的特点, 在求解多约束多目标异构无人机协同任务分配问题时, 具有较大优势. 为了高效求解该模型, 本文提出了一种基于拐点的协同多目标粒子群优化算法 (Knee point based coevolution multi-objective particle swarm

optimization, KnCMPSO). 本文的主要贡献如下:

1) 针对模型包含混合变量以及多种约束的特点, 采用基于三维矩阵的混合编码方式以及基于约束处理的初始化方法, 有效避免不可行解的生成, 提高算法的搜索效率.

2) 采用协同进化策略, 将多目标优化问题转为多个单目标优化问题, 通过子种群间合作式协同降低求解复杂度, 通过子种群内竞争式协同进化加快算法收敛, 进一步提出基于学习的粒子更新策略来提升算法的收敛性, 粒子不仅学习父代优秀个体, 也学习精英个体或全局最优个体, 可以实现更快收敛.

3) 提出基于区间扰动的局部搜索策略来提升算法的多样性, 进一步提出基于拐点的学习策略来更新外部档案集, 加强了算法对帕累托前沿中心面的搜索, 在保证收敛性的同时提升算法的多样性, 使得算法能搜索到更多的可行任务分配结果.

本文首先对无人机协同多任务分配问题建模, 然后介绍基于拐点的协同多目标粒子群优化算法, 接着设计仿真实例来验证算法的有效性, 最后进行总结和展望.

1　异构无人机协同多任务分配模型

1.1　问题描述

表 1 展示了模型中涉及的各种符号以及相应说明.

异构无人机协同多任务分配问题描述如下: 假设在某二维已知区域内发现了 N_T 个敌方目标, 则目标集合 T 可表示为 $T=\{T_1, T_2, \cdots, T_{N_T}\}$. 无人机系统需要对每个目标依次执行三种不同类型的任务, 分别为观测、打击、打击结果评估, 即三种任务之间存在时序关系. 任务类型集合 M 可以表示为 $M=\{Observe, Attack, Evaluate\}$, 因此模型中需要调度的总任务数量 N_M 的值为 $N_M=3\times N_T$. 异构无人机系统中的无人机可以被分为侦察机和战斗机两种类型, 其中用于执行观测和打击结果评估任务的侦察无人机有 S 架, 执行打击任务的战斗无人机有 A 架, 则无人机集合表示为 $U=\{U_1, U_2, \cdots, U_S, U_{S+1}, \cdots, U_{S+A}\}$. 无人机执行不同的任务需要消耗一定的资源量. 同一个目标上的相同任务可以由多台无人机协同完成. 当所有目标上的所有任务执行完成, 整个军事作战任务顺利完成.

1.2　目标函数

为了能够更加合理地对无人机任务分配的结果进行评估, 本文的模型采用无人机总飞行距离和任务完成时间两个目标函数. 无人机的总飞行距离为无人机系统中所有参与任务的无人机飞行距离之和, 任务完成时间为整个作战任务中最后一个完成任务的无人机的总飞行时间. 根据以上定义, 本文模型的计算公式如下:

表 1　符号说明

Table 1　Symbol description

类别	属性
无人机 U	侦察无人机总数 S
	战斗无人机总数 A
	初始位置 $P_i=(x_i, y_i)$
	飞行速度 V_i
	战斗无人机最大携弹数目 $Load_i$
	战斗无人机单次任务最大发射弹药数目 $Launch_i$
	侦察无人机单次任务最大侦察时间 T_i
	无人机最大飞行距离 $\mathrm{Max}Dis_i$
	无人机 U_i 执行任务 M_k 消耗的资源 C_i^k
	目标总个数 N_T
目标 T	目标位置坐标 $P_j=(x_j, y_j)$
	目标 j 的任务个数 $N_M^j=3$
	任务总个数 $N_M=3\times N_T$
任务 M	任务完成所需资源 CR_k

$$\begin{cases} f_1=\displaystyle\sum_{i=1}^{S+A}\sum_{k\in M_i} Dis(M_k^i, M_{k+1}^i) \\ f_2=\displaystyle\max_{1\le k\le N_M} t_k^{Evaluate} \end{cases} \tag{1}$$

式中, $Dis(M_k^i, M_{k+1}^i)$ 表示第 i 个无人机执行第 k 个任务到第 $k+1$ 个任务的飞行距离, 表示完成第 k 个评估任务的时间, f_1 代表了所有无人机的总飞行距离, f_2 代表了无人机系统中最后一个任务执行完成的时间, 也就是任务的最大完成时间. 由于分配任务时希望无人机尽快完成任务的同时消耗较少的飞行资源, 因此, 本文提出的模型同时考虑了上述两个优化目标, 即同时最小化函数 f_1 和 f_2.

无人机航程和其消耗资源数紧密相关, 无人机飞行总航程越短, 代表无人机执行任务时消耗的资源越少. 然而, 飞行总航程最短并不一定能保证军事任务的完成时间最短. 例如, 一些无人机可能在同一目标上被分配较多的任务以满足所有无人机飞行总航程最短, 而这会导致该无人机完成所有任务的时间变长, 从而使得整个军事任务完成时间变长. 因此, 无人机飞行总航程和任务完成时间两个指标存在一定的冲突.

1.3　约束条件

为了能够对多台无人机协同完成同一项军事任

务进行更有效的刻画, 本文在文献 [11] 基础上, 进一步考虑无人机单次任务最大观测时间约束和无人机单次任务最大弹药约束来建立异构无人机多任务分配模型, 其所包含的约束条件如下.

1) 无人机类型约束. 由于无人机的异构性, 不同类型的任务只能由特定类型的无人机完成.

$$X_i^k = \begin{cases} 1, & \text{无人机 } i \text{ 执行任务 } k \\ 0, & \text{其他} \end{cases} \tag{2}$$

式中, i 表示无人机编号, j 表示目标编号, k 表示任务编号. 当 $k \in \{Observe, Evaluate\}$ 时, $i \in \{1, 2, \cdots, S\}$; 当 $k \in \{Attack\}$时, $i \in \{S+1, \cdots, S+A\}$.

2) 任务执行时序约束. 对于同一个目标, 无人机系统要先对其执行观测任务, 然后对其执行打击任务, 最后才能够执行打击结果评估任务. 因此对同一个目标上的三种任务需满足如下时序约束.

$$T_k^{Observe} \le T_k^{Attack} \le T_k^{Evaluate},\ k \in \{1, 2, \cdots, N_M\} \tag{3}$$

3) 最大携带弹药数目约束. 战斗无人机只能携带一定数目的弹药. 因此战斗无人机执行打击任务消耗的弹药数要小于其最大携带弹药数目.

$$\sum_{k=1}^{N_M} C_i^k X_i^k \le Load_i, \quad i \in \{1, 2, \cdots, A\} \tag{4}$$

4) 单次最大侦察时间. 为了避免暴露时间过长以至于被敌方发现, 侦察无人机在执行观测任务和打击结果评估任务时存在最大侦察时间.

$$C_i^k \le T_i^k, \quad i \in \{1, 2, \cdots, S\} \tag{5}$$

5) 单次最大发射弹药数. 战斗无人机在执行对某一个目标上的打击任务时, 由于自己性能的限制只能发射一定数量的弹药.

$$C_i^k \le Launch_i^k, \quad i \in \{1, 2, \cdots, A\} \tag{6}$$

6) 多机协同约束. 为了保证军事任务顺利完成, 针对不同目标的不同类型的任务需要分配给不同数量的无人机协同执行. 其中对同一个目标的观测和打击结果评估任务至少需被一架侦察无人机执行, 而打击任务需要分配给至少一架战斗无人机执行.

$$\sum_{i=1}^{S} X_i^k \wedge \sum_{i=S+1}^{S+A} X_i^{k+1} \wedge \sum_{i=1}^{S} X_i^{k+2} = 1 \tag{7}$$

7) 完成任务资源需求约束. 执行相同任务的所有无人机资源消耗总量需要大于等于完成当前任务所需要的资源总量.

无人机执行观测任务和打击结果评估任务需要消耗一定的时间资源, 所有执行同一任务的侦察无人机侦察总时间需要满足完成任务所需总时间.

$$\sum_{i=1}^{S} C_i^k X_i^k \ge CR_k, \quad k \in \{Observe, Evaluate\} \tag{8}$$

无人机执行打击任务需要消耗一定的弹药资源, 所有执行同一任务的战斗无人机消耗弹药数量需要满足完成任务所需总弹药数量.

$$\sum_{i=S+1}^{S+A} C_i^k X_i^k \ge CR_k, \quad k \in \{Attack\} \tag{9}$$

8) 最大航程约束. 假设无人机 i 执行的任务序列为 $M_i = (M_1, M_2, \cdots, M_m)$, $Dis(M_k, M_{k+1})$表示无人机执行第 k 个任务到第 $k+1$ 个任务的飞行距离, $\mathrm{Max}Dis_i$ 表示无人机最大飞行距离, 则有:

$$\sum_{k \in M} Dis(M_k, M_{k+1}) \le \mathrm{Max}Dis_i, \quad i \in \{1, 2, \cdots, S+A\} \tag{10}$$

上述异构无人机协同多任务分配模型所包含的决策变量既有表示任务分配结果的离散变量, 也有表示无人机资源消耗的离散和连续变量, 这些混合变量导致问题解空间变得更加复杂, 难以进行有效搜索. 同时模型还包含多个复杂的约束条件, 既有不等式约束, 也有等式约束, 这些约束条件进一步使得问题对应的解空间变得不规则, 增加了算法搜索到可行解的难度.

由于现有的算法无法对本文提出的多约束多目标异构无人机协同多任务分配模型进行有效求解, 本文提出一种基于拐点的协同多目标粒子群优化算法求解无人机协同多任务分配问题.

2 基于拐点的协同多目标粒子群算法

本文提出的基于拐点的协同多目标粒子群优化算法 KnCMPSO, 主要通过设计基于约束处理的初始化策略、协同进化策略、基于学习的粒子更新策略、基于区间扰动的局部搜索策略及基于拐点的外部档案集更新策略提升算法的求解性能.

算法流程如图 1 所示. 首先, 采用子种群间合作、子种群内竞争的协同进化策略, 将多目标优化问题转为各个目标维度上的单目标优化问题, 降低了问题的求解复杂度; 其次, 在各个子种群内部, 针对无人机协同多任务分配问题包含混合变量以及多种约束的特点, 以二进制交叉方法为基础, 设计了基于学习的粒子更新策略和基于区间扰动的局部搜索策略, 提升了算法的求解性能; 最后, 引入基于拐点的学习策略来更新外部档案集, 增强了算法对帕累托前沿中心面的搜索, 在保证收敛性的同时增强

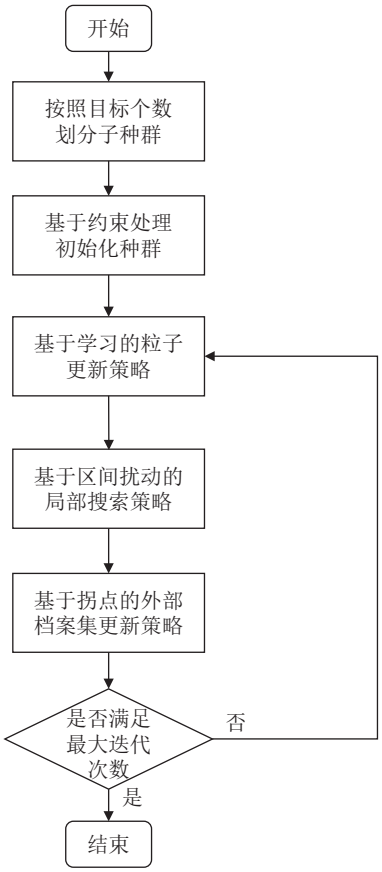


图 1　KnCMPSO 算法流程图

Fig.1　Flowchart of KnCMPSO algorithm

了算法的多样性.

2.1　编码以及初始化策略

在使用粒子群算法求解无人机协同多任务分配问题时, 首先需要解决的问题就是如何对粒子中个体进行编码操作. 为了处理模型中的混合变量以及各种约束条件, 本文将采用基于三维矩阵的混合编码方式对决策变量进行编码. 如表 2 所示, 编码矩阵中的第 1 行是目标编号, 每一个目标编号在粒子的第 1 行都会出现 3 次, 按照出现先后顺序分别代表了在此目标上执行的观测任务、打击任务、打击结果评估任务. 第 2 行是无人机编号, 代表了执行此任务的无人机. 由于目标编号在矩阵出现的先后顺序代表了不同的任务类型, 因此无人机编号 (无人机类型) 需要根据目标编号所代表的任务类型来设置. 第 3 行是无人机执行对应任务的资源消耗, 其中侦察机消耗的资源为时间, 是连续变量, 战斗机消耗的资源为弹药, 是离散变量.

基于三维矩阵的混合编码方式不但能够很好地描述目标、任务、执行任务的无人机以及无人机资源消耗的对应关系, 而且能够直观地表示出各台无人机执行任务的先后顺序, 方便后续初始化操作. 虽然每个个体位置向量编码长度仍然可能不同, 但是所有目标在编码中出现的次数是固定的, 也就是说所有粒子的编码矩阵中第 1 行目标编号的长度是固定的, 这就大大降低了问题的求解难度.

在上述编码方式的基础上, 本文提出基于约束处理的初始化策略, 具体过程如算法 1 所示.

表 3 和表 4 为初始化策略示例说明, 表 3 为目标队列集合, 表 4 为无人机集合. 首先, 建立一个 $4 \times N$ 的目标矩阵, 将所有的目标依次放入矩阵中的第 1 行, 每个目标上的任务按照观测任务、打击任务和打击结果评估任务的顺序依次排列到矩阵的第 2、3、4 行. 无人机按照侦察机和战斗机类别不同分别放入两个队列当中; 然后, 粒子将按照从左往右的顺序依次初始化位置向量. 每次均从目标矩阵中随机选择一个目标, 并将目标中尚未完成的任务取出进行分配. 根据任务类型选择相应的无人机类型, 并设置无人机对应的资源消耗. 如果此任务上的所有无人机的资源消耗量与完成此任务的资源需求量相等则代表当前任务执行完毕. 当某个目标对应的三种任务全部执行完毕代表此目标的分配全部完成; 最后, 当所有的目标都分配完成之后, 得到的粒子就是一个满足所有约束条件的可行解, 且可以有效避免出现由于某个任务的前置任务未完成而导致的死锁状态[28].

算法 1. 基于约束处理的初始化策略

输入. 目标集合 T, 任务集合 $TaskSet$, 无人机集合 U, 位置向量 $P = (M, U, C)$.

输出. 粒子的位置向量 $P = (M, U, C)$.

1) while 目标集合 $T \neq \emptyset$;

2) 从目标集合 T 中随机选择一个目标 T_j. 从该目标的任务集合 $TaskSet_j$ 中选择当前需要执行的任务 M_k. 将 k 添加到任务编号向量 M 中;

3) if $M_k \in \{Observe, Evalute\}$;

表 2　粒子编码方式

Table 2　Particle encoding scheme

目标编号 T	1		2			2			1		3	1	3	2	3	
无人机编号 U	1	3	2	3	4	5	6	5	6	1	2	4	1	2	3	
资源消耗 C	2.4	2.6	1.1	4.9	2.0	1.0	1.0	2.0	1.0	5.0	3.0	2.0	3.0	0.2	1.1	

表 3 目标队列集合
Table 3 Target formation collection

1	2	3	$\cdots$	N
T_1^O	T_2^O	T_3^O	$\cdots$	T_N^O
T_1^A	T_2^A	T_3^A	$\cdots$	T_N^A
T_1^E	T_2^E	T_3^E	$\cdots$	T_N^E

表 4 无人机集合
Table 4 Drone collection

侦察机 U_S	$U_1, U_2, \cdots, U_S$
战斗机 U_A	$U_{S+1}, U_{S+2}, \cdots, U_{S+A}$

4) 从无人机集合 U 中随机选择一架侦察无人机 U_i; 随机生成无人机 Ui 执行任务 M_k 的时间消耗 C_i^k;

5) else;

6) 从无人机集合 U 中随机选择一架战斗无人机 U_i; 随机生成无人机 U_i 执行打击任务 M_k 的导弹消耗 C_i^k;

7) end if;

8) 将 T_j 添加到无人机编号 U 中, 将 C_i^k 添加到资源消耗 C 中; 更新任务 M_k 完成所需资源 $CR_k \leftarrow CR_k - C_i^k$;

9) if $CR_k \leq 0$;

10) $TaskSet_j \leftarrow TaskSet_j - M_k$;

11) end if;

12) if $TaskSet_j$ 为空;

13) $T \leftarrow T - T_j$;

14) end if;

15) end while.

2.2 协同进化策略

协同进化是生物学中一种重要的进化机制, 可促进物种之间的信息交流并提高物种进化效率. 协同进化策略包括了合作式协同进化策略和竞争式协同进化策略两种类型[29]. 已有研究表明, 在优化算法中采用协同进化策略能够降低问题的求解难度, 加快算法的搜索效率, 提升算法的性能[30].

KnCMPSO 算法采用了子种群间合作式协同进化、子种群内竞争式协同进化的策略来生成新个体. 子种群合作式协同进化采用子问题与子种群一一对应的方式, 将多目标优化问题转为每一个维度上的单目标优化问题. 子种群内竞争式协同进化则采用子种群内的全局最优个体和外部档案集中的精英个体相互竞争的方式来选择较优的个体, 并利用该个体对子种群生成过程进行引导, 避免种群向极端点靠近. 其整体流程如图 2 所示. 针对一个 M 维的多目标优化问题, 算法生成 M 个子种群, 每个子种群分别针对每一个维度上的目标进行优化. 同时利用外部档案集 Archive 的方式保存迭代过程中寻找到的所有种群的非支配解, 从而实现种群之间搜索信息的共享.

子种群间合作式协同进化是指将原始问题分别看作某一个维度上的单目标优化问题, 同时利用外部档案集保存迭代过程中的所有种群的非支配解. 由于子问题与子种群一一对应, 子种群平行且单独地进化, 降低了问题的求解复杂度. 在传统的多目标问题中, 当两个个体互不支配的时候, 无法确定哪一个更优, 导致算法无法选择个体最优粒子和全局最优粒子对当前个体进行更新操作. 采用子种群间合作式协同进化可以有效避免上述问题. 每个子种群根据当前维度目标值的大小来进行个体最优和全局最优的选择, 实现在该维度上的求解. 子种群内竞争式协同进化是指通过各个子种群中的全局最优个体和外部档案集中的精英个体之间竞争, 获取

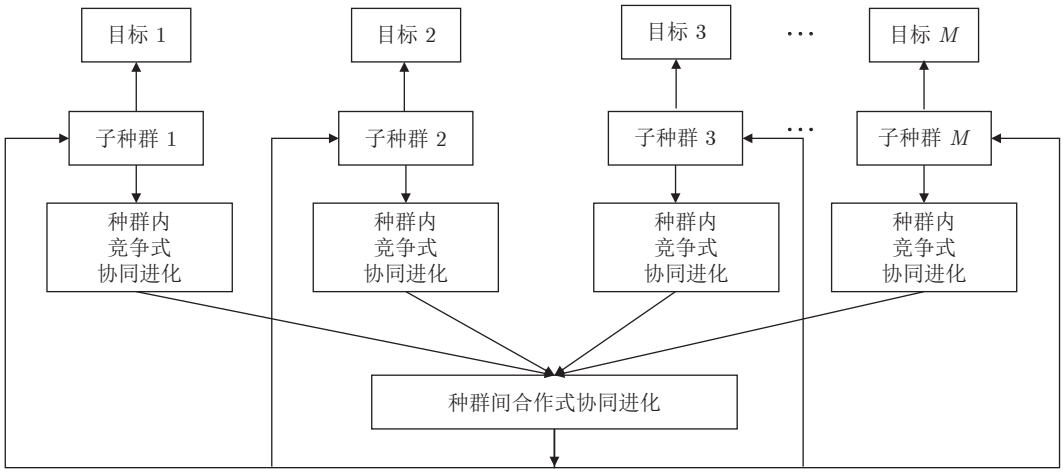


图 2 种群间合作种群内竞争的协同进化策略
Fig. 2 Coevolution strategy of inter-population cooperation and intra-population competition

对子种群中的其他个体的引导权, 具体过程如算法 2 所示.

算法 2. 基于竞争的个体更新策略

输入. 外部文档集 $Archive$, 全局最优个体 $Gbest$, 当前个体的位置向量 $cS = (M_{cS}, U_{cS}, C_{cS})$.

输出. 新个体的位置向量 $nS = (M_{nS}, U_{nS}, C_{nS})$.

1) 在外部文档集中随机选择一个精英个体 A.

2) if $\arccos(A, nS) < \arccos(Gbest, nS)$;

3) 将精英个体 A 作为待学习个体, 使用算法 3 对当前个体 cS 进行更新, 得到新个体 nS;

4) else;

5) 将全局最优个体 $Gbest$ 作为待学习个体, 使用算法 3 对当前个体 cS 进行更新, 得到新个体 nS;

6) end if.

如图 3 所示, 假设子种群 1 的优化目标为 f_1, $Gbest$ 为子种群 1 在迭代次数为 t 时候全局最优个体, 待更新的个体为 S. 首先, 在外部档案集中随机选择一个精英个体 A. 计算个体 A 和 S 之间的余弦相似度; 然后, 与 $Gbest$ 和 S 之间的余弦相似度进行比较. 当 A 与 S 之间的余弦相似度更小时, 说明个体 A 对个体 S 的引导能力更强, 更有利于算法的收敛, 因此选择 A 作为 S 的全局学习对象; 反之, 选择 $Gbest$ 作为 S 的全局学习对象.

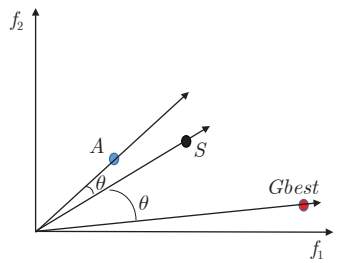


图 3　竞争式协同进化策略

Fig. 3　Competitive coevolution strategy

上述两种协同进化策略不仅能降低问题的求解复杂度, 而且能有效提高算法的收敛速度.

2.3　基于学习的粒子更新策略

在标准粒子群算法中, 种群中的个体通过对个体最优粒子和全局最优个体进行学习来更新速度, 再通过粒子的当前位置和速度来更新粒子的状态. 本文模型中包含了混合变量, 并且约束条件较多, 按照标准粒子群的更新方式将很难生成满足约束的粒子, 所以本文将对粒子的更新方式进行改进, 不再使用粒子速度这一个概念, 粒子在更新的时候只通过向优秀个体学习的方式来更新粒子的状态. 基于此, 在 KnCMPSO 中提出了一种基于学习的粒子更新策略, 具体过程如算法 3 所示.

算法 3. 基于学习的粒子更新策略

输入. 待学习个体的位置向量 $lS = (M_{lS}, U_{lS}, C_{lS})$, 当前个体的位置向量 $cS = (M_{cS}, U_{cS}, C_{cS})$, 学习概率 pro.

输出. 新个体的位置向量 $nS = (M_{nS}, U_{nS}, C_{nS})$.

1) 初始化新个体的位置向量 $nS = (M_{nS}, U_{nS}, C_{nS})$;

2) for 目标 j, $0 \le j \le N_T$;

3) if $random < pro$;

4) 将待学习个体 lS 位置向量第 1 行与当前目标 j 相等的三个位置上的值, 按照其在 lS 位置向量中出现的先后次序依次插入到新个体 nS 位置向量的对应位置;

5) 将各个任务对应的无人机以及无人机资源消耗量插入到 nS 位置向量的对应位置;

6) end if;

7) end for;

8) for 位置向量维度 s, $0 \le s \le nS$;

9) if 位置向量维度 s 为空;

10) 将当前个体 cS 中的目标、目标对应的无人机以及无人机资源使用量从左到右依次填入到新个体 nS 位置向量中的空位置;

11) end if;

12) end for.

图 4 中的 lS 表示待学习的个体, cS 表示当前个体, nS 代表经过学习方式之后生成的新个体. 整个学习过程如下: 首先, 初始化一个空的个体 nS. 依次选择模型中的目标编号, 以一定的概率选择当前目标编号进行学习操作. 假设当前选中的目标编号为 2, 则将待学习个体位置向量第 1 行与 2 相等的三个位置上的值, 按照其在 lS 位置向量中出现的先后次序依次插入到新个体 nS 位置向量的对应位置, 同时将各个任务对应的无人机以及无人机资源消耗量插入到位置向量的对应位置. 对目标编号进行一次遍历之后, 新个体 nS 中非空位置保存的就是待学习个体 lS 的信息. 然后, 将原始个体 cS 中的目标、目标对应的无人机以及无人机资源使用量从左到右依次填入到新个体位置向量中的空位置, 跳过新个体 nS 中已存在的目标.

通过向优秀个体进行学习, 粒子位置向量将按照式 (11) 进行更新. 其中 $X_i(t)$ 代表粒子 i 在第 t 次迭代中的位置, $Pbest_i$ 为粒子在第 t 次迭代中的个体最优值, A 为外部档案集中的个体, $Gbest$ 为第 t 次迭代中的全局最优, w 为权重系数, $competition$ 代表了粒子 A 和 $Gbest$ 相互竞争, F 代表了粒子向较优个体的学习过程.

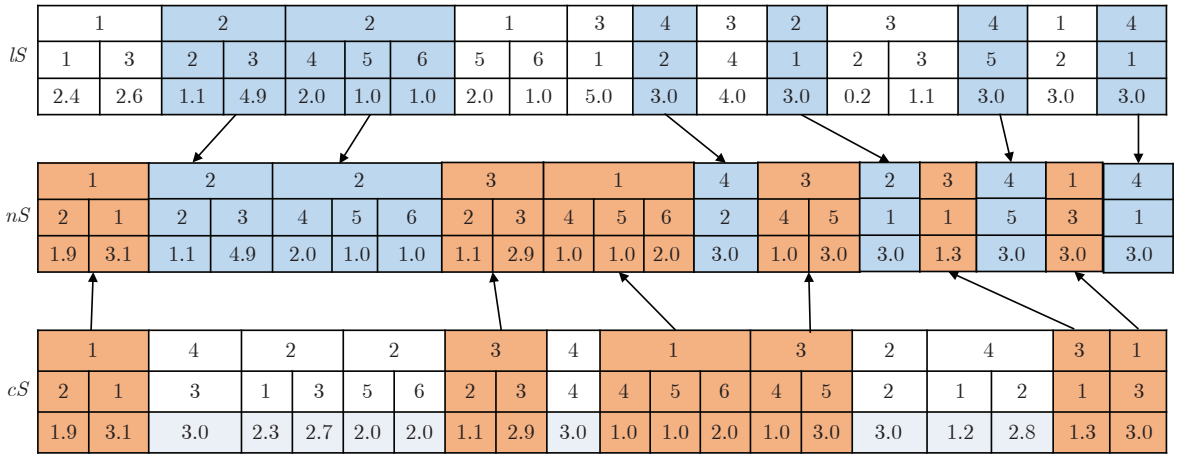


图 4 粒子向较优个体的学习过程

Fig. 4 The learning process of particles from better individuals

$$X_i(t+1) = (1-w) \otimes F(w \otimes F(X_i(t),\ Pbest_i),\ competition(A,\ Gbest)) \tag{11}$$

2.4 基于区间扰动的局部搜索

如前所述, 采用上述基于学习的粒子更新策略对种群中的个体进行更新能够有效地对问题空间进行搜索, 但如果粒子在更新的时候只采用该策略对群体中较优的个体进行学习, 会导致算法在迭代后期陷入局部最优. 为了进一步提升种群的多样性以及算法的搜索能力, 本文进一步提出了一种基于扰动的局部搜索策略, 具体过程如算法 4 所示.

算法 4. 基于区间扰动的局部搜索策略

输入. 当前个体的位置向量 $cS = (M_{cS},\ U_{cS},\ C_{cS})$.

输出. 新个体的位置向量 $nS = (M_{nS},\ U_{nS},\ C_{nS})$.

1) 初始化新个体的位置向量 $nS = (M_{nS},\ U_{nS},\ C_{nS})$;

2) 随机选择一个任务 M_k;

3) 根据 M_k 的任务类型, 找到任务 M_k 可插入的范围 $[startPos,\ endPos]$;

4) 在 $[startPos,\ endPos]$ 范围内随机选择任务插入的位置 $insertPos$;

5) 将 M_k 插入位置向量中的 $insertPos$, 其余位置的任务依次插入到 nS 的 $[startPos,\ endPos]$ 范围中;

6) 重新设置完成任务 M_k 的无人机以及相应的资源消耗.

具体实现方式见图 5. 图 5 中 cS 代表当前个体, nS 代表执行完局部搜索策略之后生成的新个体. 按照一定的概率在个体 cS 上随机选择某一个目标上的某一个任务 k. 按照任务执行的时序约束找到任务 k 可插入的位置范围 $[startPos,\ endPos]$. 图 5 中加下划线的方框表示随机选择的任务 k, 两个灰色背景方框的位置从左往右分别为 $startPos$

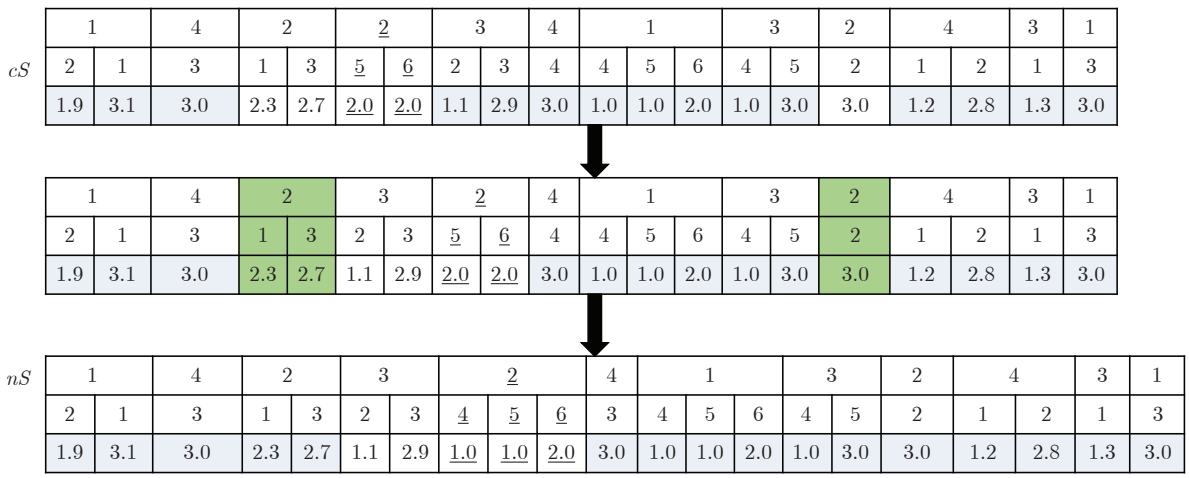


图 5 基于区间扰动的局部搜索策略

Fig. 5 Local search strategy based on interval disturbance

和 $endPos$. 在此范围内随机选择任务插入的位置 $insertPos$. 将任务 k 插入位置向量中的 $insertPos$. 其余位置的任务依次插入到 nS 的 $[startPos, endPos]$ 之中, 并重新设置完成任务 k 的无人机以及相应的资源消耗. 需要指出的是, 任务类型应满足无人机作战类型约束, 即该任务如果是观测和打击结果的评估任务, 则只能选择侦察无人机集合中的无人机, 否则选择战斗无人机集合中的无人机. 此外, 所选择的无人机仍需满足资源约束和最大航程约束.

2.5　基于拐点的外部档案集更新策略

将原始问题看作各个目标维度上的单目标优化问题, 虽然降低了问题的求解难度, 但是由于各个子种群分别优化某一个目标, 种群将会向每个目标维度上的极端点靠近. 最后得到的非支配解集也将更多地分布在各个目标维度的附近, 导致算法搜索不到帕累托面的中心部分, 降低算法的多样性. 如图 6 所示, 以两目标问题为例, 其中矩形点为极端点, 圆形点代表非支配解, 三角形点为支配解, 虚线框所在的区域为中心解集区域. 可以明显看出, 中心区域解集分布较少. 这是由于每一个子种群的全局最优为对应目标维度的极端点, 使得种群在不断迭代过程中大部分的解将向极端点靠近, 导致最后生成的非支配解集多样性不足.

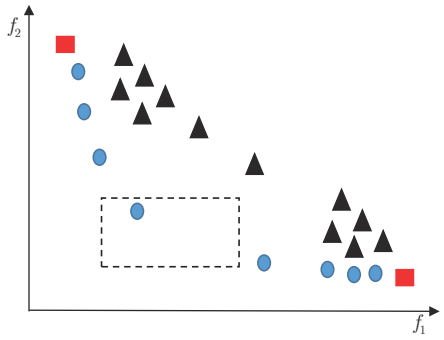


图 6　解集分布不均匀图

Fig. 6　Uneven solution set distribution graph

为了使得外部档案集中的解集分布更加均匀, 同时提升算法的收敛性, 本文利用拐点来引导档案集中的其他精英个体的更新. 如图 7 所示, 点 B 和点 C 为某个种群内部的边界点, 两者之间的连线为 L, 通过计算种群中其他点到直线 L 的距离, 选择距离最大的那个点作为拐点, 即点 A. 对于三目标或者是超多目标问题, 边界点所构成的是一个平面或是超平面, 则拐点也被定义为距离这个平面或者是超平面最远的点. 本文所提的利用拐点更新外部档案集的方式如式 (12) 所示, 外部档案集中的个体向

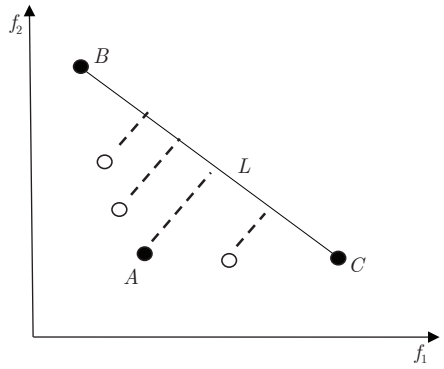


图 7　拐点图示

Fig. 7　Illustration of knee point

拐点学习更新位置向量.

式 (12) 中 A_i 代表外部档案集 $Archive$ 中任意的一个精英个体, $kneePt$ 代表拐点, F 为粒子向优秀个体的学习过程, 具体实现方式见第 2.3 节.

$$A_i = F(A_i, kneePt) \tag{12}$$

3　实验与结果分析

本文基于不同的任务数量和无人机类型设计了 4 种不同规模的实例并进行了仿真实验. 实验中的算法均采用 Java 编程, 运行环境为 JDK1.8, 编译器为 Eclipse, 处理器为主频 3.6 GHz 的 Intel Core i7-4790.

3.1　仿真实例设计

本文设计了 4 种不同规模的实例, 其中实例 1 包含 14 台无人机和 6 个军事任务目标, 实例 2 包含 16 台无人机和 12 个军事任务目标, 实例 3 包含 18 台无人机和 18 个军事任务目标, 实例 4 包含 20 台无人机和 24 个军事任务目标. 各个实例中的任务目标被随机地设置在一个固定的位置, 每一个任务目标上包含了三种属性, 分别代表了完成此目标上的某一个军事任务所需要的资源量. 各个实例中的无人机同样的被随机地设置在一个固定的位置, 每一个无人机包括了飞行速度、最大携带弹药数目和最远飞行距离. 实例 4 中各个目标和无人机的属性设置情况如表 5 和表 6 所示, 其他 3 个实例见附录 A. 上述实例包含不同规模、不同分布的无人机任务分配场景, 4 个实例的示意图如图 8 所示. 其中, 实例 1 和实例 2 中的目标和无人机在军事区域内分布的相对集中且规模较小, 这意味着在任务分配时可选择的合适的无人机较多, 因此在实例 1 和实例 2 的目标空间中存在较多的局部最优解, 算法在实例 1 和实例 2 上的实验结果能反映算法求解多峰优化问题时的性能. 而实例 3 和实例 4 中目标和

表 5 实例 4 中目标属性值

Table 5 The target attribute value in example 4

目标	坐标 (千米, 千米)	观测时间 (秒)	打击所需弹药数 (枚)	评估时间 (秒)
$T1$	(13, 41)	100	2	40
$T2$	(63, 83)	120	1	20
$T3$	(79, 12)	40	3	10
$T4$	(41, 98)	90	4	15
$T5$	(23, 65)	150	2	20
$T6$	(53, 19)	20	2	5
$T7$	(36, 49)	100	2	40
$T8$	(70, 47)	120	1	20
$T9$	(25, 15)	40	3	10
$T10$	(62, 89)	90	4	15
$T11$	(54, 42)	150	2	20
$T12$	(61, 39)	20	2	5
$T13$	(74, 29)	100	2	40
$T14$	(68, 33)	120	1	20
$T15$	(76, 38)	40	3	10
$T16$	(96, 26)	90	4	15
$T17$	(52, 55)	150	2	20
$T18$	(59, 98)	20	2	5
$T19$	(30, 43)	100	2	40
$T20$	(27, 82)	120	1	20
$T21$	(59, 13)	40	3	10
$T22$	(42, 28)	90	4	15
$T23$	(55, 39)	15	2	20
$T24$	(8, 46)	20	2	5

表 6 实例 4 中无人机属性值

Table 6 Attribute value of drone in example 4

无人机	坐标 (千米, 千米)	无人机类型	携带弹药数 (枚)	飞行速度 (千米/秒)
$U1$	(57, 5)	侦察机	0	0.10
$U2$	(42, 8)	侦察机	0	0.12
$U3$	(50, 99)	侦察机	0	0.12
$U4$	(62, 25)	侦察机	0	0.11
$U5$	(20, 61)	侦察机	0	0.1
$U6$	(83, 46)	侦察机	0	0.12
$U7$	(56, 90)	侦察机	0	0.11
$U8$	(39, 45)	侦察机	0	0.11
$U9$	(58, 69)	侦察机	0	0.11
$U10$	(13, 86)	侦察机	0	0.1
$U11$	(5, 57)	侦察机	0	0.11
$U12$	(86, 46)	战斗机	11	0.13
$U13$	(89, 34)	战斗机	8	0.09
$U14$	(53, 11)	战斗机	10	0.11
$U15$	(92, 86)	战斗机	9	0.09
$U16$	(96, 18)	战斗机	12	0.11
$U17$	(73, 76)	战斗机	8	0.16
$U18$	(47, 56)	战斗机	7	0.16
$U19$	(56, 80)	战斗机	8	0.12
$U20$	(78, 4)	战斗机	9	0.16

无人机的分布相对分散且规模更大, 算法在实例 3 和实例 4 上的实验结果能较好地反映算法求解模型的收敛性能. 通过在 4 个实例上做实验, 可以对算法的探索能力和勘探能力进行较全面的评估.

其中军事任务目标和无人机的坐标单位为千米, 无人机飞行速度单位为千米/秒, 无人机最大飞机距离为 2 000 千米. 无人机完成观测任务和打击结果评估任务的最大观测时间为 60 秒. 无人机完成一次打击任务最大消耗弹药数量为 4 枚.

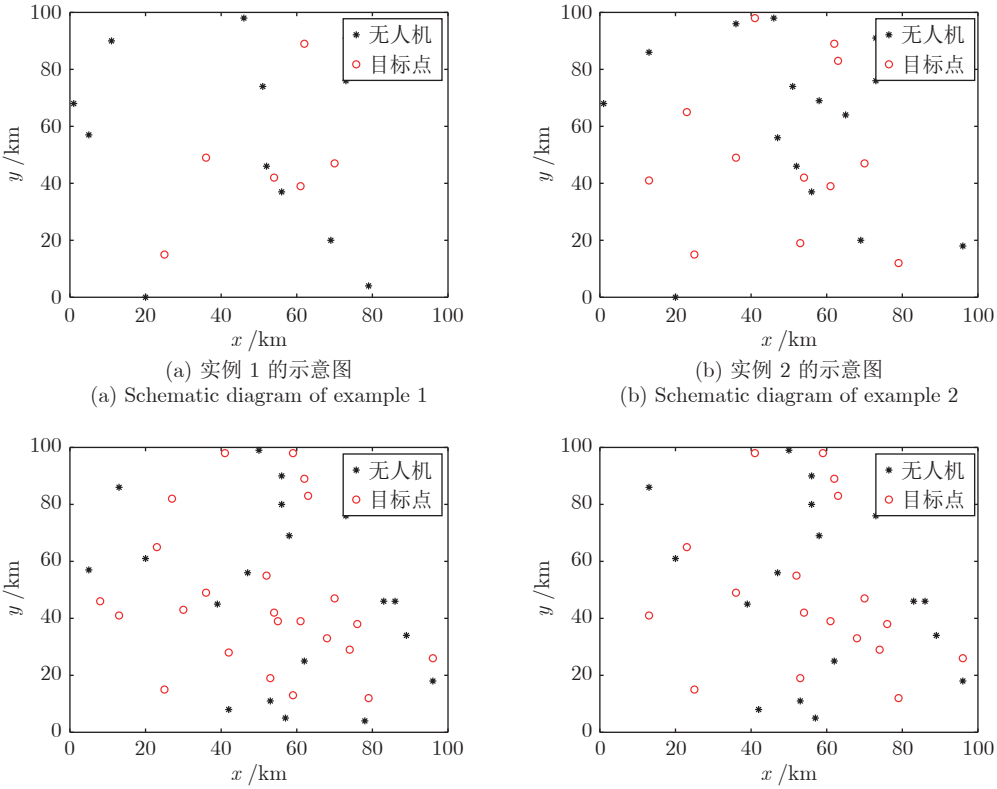
3.2 对比算法和评价指标

为了验证 KnCMPSO 算法性能, 本文选取求解无人机协同任务分配问题的基于协同进化的混合变量多目标粒子群优化算法 (Coevolution based mixed-variable multi-objective particle swarm optimization algorithm, C-MOPSO)[11] 与 3 个具有代表性的基于协同进化的多种群粒子群优化算法 (Coevolutionary multiswarm particle swarm optimization, CMPSO)[30]、基于协同进化的粒子群优化算法 (Coevolutionary particle swarm optimization, CPSO)[31] 和基于分解的协同进化多目标局部搜索算法 (Coevolutionary multiobjective local search based on decomposition, CoMOLS/D)[32] 进行对比实验. 所有算法的种群大小 N 设置为 300, 最大迭代次数为 500 次, 外部档案集大小均为种群大小的二分之一. 同时采用超体积 Hypervolum (HV) 指标评价各个算法获得非支配解集的优劣. HV 指标是一个综合性评价指标, 能够同时评估算法在收敛性和多样性上的表现. HV 指标的数值越大, 表示算法在收敛性和多样性上的表现越好. 本文 HV 指标参考点设置为 (15 000, 15 000).

3.3 实验结果与分析

为了使选取的各对比算法能够求解本文的模型, 上述对比算法均采用了与 KnCMPSO 相同的编码策略以及初始化方式. 需要指出的是, 为了保证实验结果的公平性, 除 C-MOPSO 算法之外, 其他的算法均采用本文所提出的粒子更新方式和局部搜索策略.

在实验过程中, 每个算法均将在上述无人机任务分配实例上执行 30 次独立实验. 各算法在所有实例上的 HV 指标数据如表 7 所示, 其中 Mean 表



(a) 实例 1 的示意图
(a) Schematic diagram of example 1

(b) 实例 2 的示意图
(b) Schematic diagram of example 2

(c) 实例 3 的示意图
(c) Schematic diagram of example 3

(d) 实例 4 的示意图
(d) Schematic diagram of example 4

图 8　4 个实例的示意图

Fig. 8　Schematic diagram of four examples

表 7　算法在各个实例上的 HV 值

Table 7　The HV value of the algorithm on each example

实例名称	HV	CMPSO	CoMOLS/D	CPSO	C-MOPSO	KnCMPSO
实例 1	Mean	1.941×10^8	1.846×10^8	1.934×10^8	1.873×10^8	$\mathbf{1.963} \times 10^8$
	Std	1.326×10^6	2.035×10^6	1.574×10^6	2.483×10^6	$\mathbf{1.053} \times \mathbf{10^6}$
	Worst	1.918×10^8	1.799×10^8	1.893×10^8	1.836×10^8	$\mathbf{1.942} \times \mathbf{10^8}$
	Best	1.969×10^8	1.881×10^8	1.958×10^8	1.937×10^8	$\mathbf{1.982} \times \mathbf{10^8}$
实例 2	Mean	1.586×10^8	1.312×10^8	1.554×10^8	1.371×10^8	$\mathbf{1.629} \times \mathbf{10^8}$
	Std	$\mathbf{2.916} \times \mathbf{10^6}$	3.387×10^6	3.158×10^6	3.523×10^6	3.114×10^6
	Worst	1.525×10^8	1.259×10^8	1.494×10^8	1.297×10^8	$\mathbf{1.544} \times \mathbf{10^8}$
	Best	1.640×10^8	1.371×10^8	1.637×10^8	1.464×10^8	$\mathbf{1.675} \times \mathbf{10^8}$
实例 3	Mean	1.326×10^8	9.365×10^7	1.284×10^8	1.032×10^8	$\mathbf{1.387} \times \mathbf{10^8}$
	Std	$\mathbf{3.133} \times \mathbf{10^6}$	4.722×10^6	3.174×10^6	4.292×10^6	4.271×10^6
	Worst	1.265×10^8	8.571×10^7	1.208×10^8	9.782×10^7	$\mathbf{1.289} \times \mathbf{10^8}$
	Best	1.383×10^8	1.029×10^8	1.365×10^8	1.130×10^8	$\mathbf{1.487} \times \mathbf{10^8}$
实例 4	Mean	1.103×10^8	6.209×10^7	1.027×10^8	7.068×10^7	$\mathbf{1.151} \times \mathbf{10^8}$
	Std	4.482×10^6	3.658×10^6	4.850×10^6	$\mathbf{3.056} \times \mathbf{10^6}$	6.337×10^6
	Worst	9.913×10^7	5.582×10^7	9.023×10^7	6.601×10^7	$\mathbf{1.033} \times \mathbf{10^8}$
	Best	1.192×10^8	6.970×10^7	1.116×10^8	7.631×10^7	$\mathbf{1.241} \times \mathbf{10^8}$

示各个算法 30 次实验计算得到的 HV 指标的平均值, 其大小反映算法求得的非支配解集的收敛性和多样性. Std 表示各个算法 30 次实验计算得到的 HV 指标的方差, 其大小反映算法的稳定性. Best

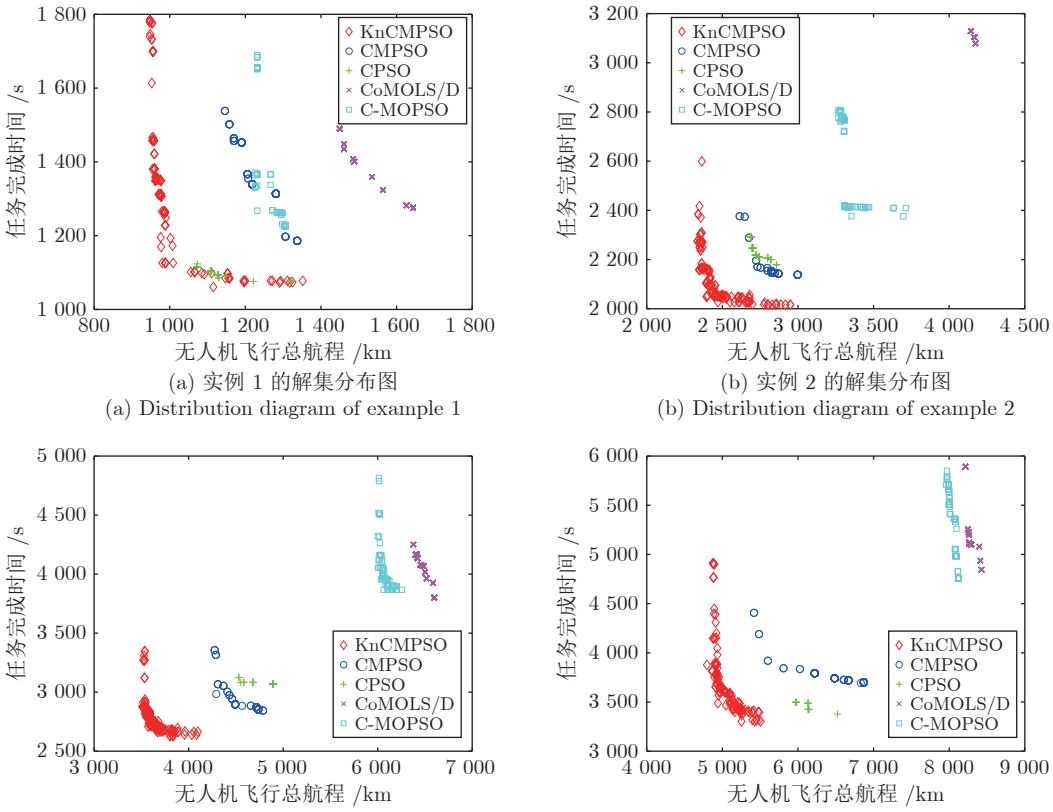
表示算法运行 30 次后得到的最优的 HV 值. Worst 表示算法得到的最差 HV 值. 表 7 中各个测试函数上的最优结果加粗标注, 次优结果用下划线标注.

由表 7 可以看出, 相较于其他对比算法, KnCMPSO 在 4 个实例上的 HV 指标的平均值上都能取得最优结果, 表明了算法在多样性和收敛性上均优于其他对比算法. 另外 KnCMPSO 算法求得 HV 指标中 Std、Best 和 Worst 值相比较于其他对比算法同样是占优的, 表明了 KnCMPSO 算法在稳定性上同样优于其他对比算法. 原因如下: 1) KnCMPSO 利用协同进化的策略降低了问题的求解难度; 2) 通过引入基于拐点的外部档案集更新策略增加了算法对帕累托前沿中心区域的搜索能力. CMPSO 算法和 CPSO 算法虽然同样采用了协同进化策略, 但是在 CMPSO 和 CPSO 算法中极端点对个体的引导能力过于强烈, 导致最后生成的解集更多分布在各个目标维度极端解的附近, 算法的多样性受损. CoMOLS/D 算法采用的权重和法和反转边界交叉惩罚均是基于权重向量的策略. 由于解集的好坏与权重向量的设置关系十分密切, 在求解帕累托前沿分布不均匀的实际问题上优势不明显. C-MOPSO 算法针对无人机任务分配问题设计了基于结构学习的重组方法进行求解, 但是这种重组方法不确定性较大, 尤其是随着问题规模的不断变大, 算法的求解性能逐渐降低, 与 C-MOPSO 算法的对比证明了本文所提粒子更新方式和局部搜索策略的有效性.

为了更加直观展示算法在各个实例上的结果分布情况, 算法在每个实例上所得解集的分布情况如图 9 所示. 由图中各实例解集分布情况可以看出, 相比较于其他对比算法, 本文算法求解得到的解集不论是在多样性还是收敛性上都明显占优, 这与上述 HV 指标的评估结果一致.

通过上述实验结果中本文算法和其他算法 HV 指标性能表现, 以及 5 个算法在各个实例中的解集分布情况可以发现, 相比较于其他 4 个算法, 本文算法无论在收敛性方面还是多样性方面均明显占优, 充分证明了本文算法求解无人机协同多任务分配问题的有效性.

为了验证 KnCMPSO 算法所提基于学习的粒



(a) 实例 1 的解集分布图
(a) Distribution diagram of example 1

(b) 实例 2 的解集分布图
(b) Distribution diagram of example 2

(c) 实例 3 的解集分布图
(c) Distribution diagram of example 3

(d) 实例 4 的解集分布图
(d) Distribution diagram of example 4

图 9 算法在各实例上的解集的分布图

Fig. 9 Distribution diagram of the solution set of the algorithm on each example

子更新策略的有效性, 本文将文献 [11] 的基于结构学习的生成算子 (Structure learning reproduction, SLR) 嵌入到 KnCMPSO 算法框架中, 记该算法为 KnCMPSO-SLR, 与现有结果进行了对比, 对比结果见表 8. 由表 8 可以看出, KnCMPSO 在实例 2 上与 KnCMPSO-SLR 的结果相近, 但在实例 1、实例 3 和实例 4 上都能取得最优结果, 表明该算法可以保持较好的多样性和收敛性, 其原因在于 KnCMPSO-SLR 采用基于结构学习的粒子更新策略只学习了父代个体的部分基因, 而 KnCMPSO 采用基于学习的粒子更新策略不仅学习父代个体, 还向精英个体或全局最优个体学习, 因此具有更好的收敛性能.

同时, 考虑到本文求解实际问题的特殊性, KnCMPSO 在解的初始化过程中提出了一种基于约束处理的初始化方法, 为了验证该方法的优越性, 本文选择采用其他 3 种经典约束处理方法与本文 KnCMPSO 算法框架结合构成新的算法, 包括与基于罚函数[18] 的约束处理方法结合 KnCMPSO-PF, 与基于可行性原则的约束处理方法[21] 结合 KnCMPSO-SR 和与基于随机排序的约束处理方法结合 KnCMPSO-FR[22], 并将这 3 种算法与本文算法进行实验对比, 实验结果见表 9. 由表 9 可以看出, KnCMPSO 在所有实例上均取得了最优结果, 这主要归因于 KnCMPSO 将约束处理融入到粒子初始化以及更新的过程中, 能有效地解决多种复杂约束, 保证算法迭代过程中生成的解都为可行解, 且在迭代过程中采用基于区间扰动的变异策略来提升种群的多样性, 提高了算法的搜索效率. 而其他算法在迭代过程中生成了部分不可行解, 导致算法搜索效率下降.

表 8　算法在各个实例上的 HV 值

Table 8　The HV value of the algorithm on each example

实例名称	HV	KnCMPSO-SLR	KnCMPSO
实例 1	Mean	1.951×10^8	$\mathbf{1.974 \times 10^8}$
	Std	$\mathbf{7.942 \times 10^5}$	1.046×10^6
	Worst	1.934×10^8	$\mathbf{1.959 \times 10^8}$
	Best	1.961×10^8	$\mathbf{1.989 \times 10^8}$
实例 2	Mean	$\mathbf{1.625 \times 10^8}$	1.622×10^8
	Std	3.290×10^6	$\mathbf{1.632 \times 10^6}$
	Worst	1.569×10^8	$\mathbf{1.598 \times 10^8}$
	Best	$\mathbf{1.690 \times 10^8}$	1.645×10^8
实例 3	Mean	1.347×10^8	$\mathbf{1.395 \times 10^8}$
	Std	$\mathbf{2.031 \times 10^6}$	3.099×10^6
	Worst	1.318×10^8	$\mathbf{1.361 \times 10^8}$
	Best	1.385×10^8	$\mathbf{1.449 \times 10^8}$
实例 4	Mean	1.146×10^8	$\mathbf{1.184 \times 10^8}$
	Std	$\mathbf{2.251 \times 10^6}$	3.492×10^6
	Worst	1.087×10^8	$\mathbf{1.124 \times 10^8}$
	Best	1.165×10^8	$\mathbf{1.241 \times 10^8}$

表 9　算法在各个实例上的 HV 值

Table 9　The HV value of the algorithm on each example

实例名称	HV	KnCMPSO-PF	KnCMPSO-FR	KnCMPSO-SR	KnCMPSO
实例 1	Mean	$\underline{1.800 \times 10^8}$	1.782×10^8	1.786×10^8	$\mathbf{1.861 \times 10^8}$
	Std	$\underline{3.005 \times 10^6}$	3.881×10^6	3.778×10^6	$\mathbf{1.948 \times 10^6}$
	Worst	$\underline{1.721 \times 10^8}$	1.703×10^8	1.705×10^8	$\mathbf{1.800 \times 10^8}$
	Best	$\underline{1.860 \times 10^8}$	1.837×10^8	1.839×10^8	$\mathbf{1.898 \times 10^8}$
实例 2	Mean	$\underline{1.362 \times 10^8}$	1.346×10^8	1.356×10^8	$\mathbf{1.449 \times 10^8}$
	Std	$\underline{4.110 \times 10^6}$	$\mathbf{3.981 \times 10^6}$	4.166×10^6	4.562×10^6
	Worst	1.275×10^8	1.257×10^8	$\underline{1.296 \times 10^8}$	$\mathbf{1.362 \times 10^8}$
	Best	1.435×10^8	1.426×10^8	$\underline{1.453 \times 10^8}$	$\mathbf{1.529 \times 10^8}$
实例 3	Mean	$\underline{1.046 \times 10^8}$	1.036×10^8	1.003×10^8	$\mathbf{1.175 \times 10^8}$
	Std	4.892×10^6	6.045×10^6	$\underline{4.629 \times 10^6}$	$\mathbf{4.027 \times 10^6}$
	Worst	$\underline{9.602 \times 10^7}$	8.900×10^7	8.618×10^7	$\mathbf{1.112 \times 10^8}$
	Best	$\underline{1.145 \times 10^8}$	1.139×10^8	1.066×10^8	$\mathbf{1.243 \times 10^8}$
实例 4	Mean	$\underline{7.920 \times 10^7}$	7.851×10^7	7.833×10^7	$\mathbf{9.603 \times 10^7}$
	Std	6.365×10^6	6.608×10^6	$\underline{6.008 \times 10^6}$	$\mathbf{5.376 \times 10^6}$
	Worst	$\underline{6.537 \times 10^7}$	6.412×10^7	6.007×10^7	$\mathbf{8.466 \times 10^7}$
	Best	$\underline{9.105 \times 10^7}$	8.916×10^7	8.653×10^7	$\mathbf{1.045 \times 10^8}$

4 结束语

本文以异构无人机协同作战为背景, 针对无人机协同多任务分配问题建立了包含多种约束条件的异构无人机协同多任务分配问题模型. 为了求解此模型, 本文提出了基于拐点的协同多目标粒子群优化算法 KnCMPSO, 并设计了 4 种不同规模的实例进行仿真实验. 实验结果显示本文所提的算法在多样性和收敛性上均优于其他的对比算法, 表明本文所提算法能够有效求解无人机协同多任务分配问题.

KnCMPSO 算法采用混合编码方式以及基于约束处理的初始化方法, 能有效避免不可行解的生成, 采用协同进化策略将多目标优化问题转为各目标维度上的单目标优化问题, 同时设计相应的搜索策略提升算法的收敛性和多样性, 因此可以很好地适用于多目标多约束的无人机任务分配问题. 但由于 KnCMPSO 算法的编码和搜索策略是根据异构无人机任务分配问题的特性进行设计, 对于其他的多目标多约束混合变量优化问题, 还需研究特定的编码方式和搜索策略来进行求解.

本文提出的模型虽然考虑了更加符合实际情况的约束条件和目标函数, 但是相比于现实作战环境的复杂性还存在一定的差距, 为了更进一步提高任务分配模型的有效性, 未来在构建无人机任务分配模型时将考虑更多符合实际的约束条件和目标函数.

附录 A

表 A1 和表 A2 分别展示了实例 3 的目标和无人机的各种属性. 表 A3 和表 A4 分别展示了实例 2 的目标和无人机的各种属性. 表 A5 和表 A6 分别展示了实例 1 的目标和无人机的各种属性.

表 A1 实例 3 中目标属性值

Table A1 The target attribute value in example 3

目标	坐标 (千米, 千米)	观测时间 (秒)	打击所需弹药数 (枚)	评估时间 (秒)
$T1$	(13, 41)	100	2	40
$T2$	(63, 83)	120	1	20
$T3$	(79, 12)	40	3	10
$T4$	(41, 98)	90	4	15
$T5$	(23, 65)	150	2	20
$T6$	(53, 19)	20	2	5
$T7$	(36, 49)	100	2	40
$T8$	(70, 47)	120	1	20
$T9$	(25, 15)	40	3	10
$T10$	(62, 89)	90	4	15
$T11$	(54, 42)	150	2	20
$T12$	(61, 39)	20	2	5

表 A1 实例 3 中目标属性值 (续表)

Table A1 The target attribute value in example 3 (continued table)

目标	坐标 (千米, 千米)	观测时间 (秒)	打击所需弹药数 (枚)	评估时间 (秒)
$T13$	(74, 29)	100	2	40
$T14$	(68, 33)	120	1	20
$T15$	(76, 38)	40	3	10
$T16$	(96, 26)	90	4	15
$T17$	(52, 55)	150	2	20
$T18$	(59, 98)	20	2	5

表 A2 实例 3 中无人机属性值

Table A2 Attribute value of drone in example 3

无人机	坐标 (千米, 千米)	无人机类型	携带弹药数 (枚)	飞行速度 (千米/秒)
$U1$	(57, 5)	侦察机	0	0.10
$U2$	(42, 8)	侦察机	0	0.12
$U3$	(50, 99)	侦察机	0	0.12
$U4$	(62, 25)	侦察机	0	0.11
$U5$	(20, 61)	侦察机	0	0.10
$U6$	(83, 46)	侦察机	0	0.12
$U7$	(56, 90)	侦察机	0	0.11
$U8$	(39, 45)	侦察机	0	0.11
$U9$	(58, 69)	侦察机	0	0.11
$U10$	(13, 86)	侦察机	0	0.10
$U11$	(56, 80)	战斗机	8	0.12
$U12$	(86, 46)	战斗机	9	0.13
$U13$	(89, 34)	战斗机	7	0.09
$U14$	(53, 11)	战斗机	9	0.11
$U15$	(92, 86)	战斗机	7	0.09
$U16$	(96, 18)	战斗机	9	0.11
$U17$	(73, 76)	战斗机	7	0.16
$U18$	(47, 56)	战斗机	5	0.16

表 A3 实例 2 中目标属性值

Table A3 The target attribute value in example 2

目标	坐标 (千米, 千米)	观测时间 (秒)	打击所需弹药数 (枚)	评估时间 (秒)
$T1$	(13, 41)	100	2	40
$T2$	(63, 83)	120	1	20
$T3$	(79, 12)	40	3	10
$T4$	(41, 98)	90	4	15
$T5$	(23, 65)	150	2	20
$T6$	(53, 19)	20	2	5
$T7$	(36, 49)	100	2	40
$T8$	(70, 47)	120	1	20
$T9$	(25, 15)	40	3	10
$T10$	(62, 89)	90	4	15
$T11$	(54, 42)	150	2	20
$T12$	(61, 39)	20	2	5

表 A4 实例 2 中无人机属性值

Table A4 Attribute value of drone in example 2

无人机	坐标 (千米, 千米)	无人机类型	携带弹药数 (枚)	飞行速度 (千米/秒)
$U1$	(73, 91)	侦察机	0	0.10
$U2$	(65, 64)	侦察机	0	0.12
$U3$	(56, 37)	侦察机	0	0.12
$U4$	(36, 96)	侦察机	0	0.11
$U5$	(20, 0)	侦察机	0	0.10
$U6$	(1, 68)	侦察机	0	0.12
$U7$	(51, 74)	侦察机	0	0.11
$U8$	(58, 69)	侦察机	0	0.11
$U9$	(13, 86)	侦察机	0	0.10
$U10$	(69, 20)	战斗机	10	0.10
$U11$	(52, 46)	战斗机	6	0.12
$U12$	(46, 98)	战斗机	7	0.13
$U13$	(92, 86)	战斗机	5	0.09
$U14$	(96, 18)	战斗机	7	0.11
$U15$	(73, 76)	战斗机	5	0.16
$U16$	(47, 56)	战斗机	5	0.16

表 A5 实例 1 中目标属性值

Table A5 The target attribute value in example 1

目标	坐标 (千米, 千米)	观测时间 (秒)	打击所需弹药数 (枚)	评估时间 (秒)
$T1$	(36, 49)	100	2	40
$T2$	(70, 47)	120	1	20
$T3$	(25, 15)	40	3	10
$T4$	(62, 89)	90	4	15
$T5$	(54, 42)	150	2	20
$T6$	(61, 39)	20	2	5

表 A6 实例 1 中无人机属性值

Table A6 Attribute value of drone in example 1

无人机	坐标 (千米, 千米)	无人机类型	携带弹药数 (枚)	飞行速度 (千米/秒)
$U1$	(73, 91)	侦察机	0	0.10
$U2$	(94, 92)	侦察机	0	0.12
$U3$	(56, 37)	侦察机	0	0.12
$U4$	(5, 57)	侦察机	0	0.11
$U5$	(20, 0)	侦察机	0	0.10
$U6$	(1, 68)	侦察机	0	0.12
$U7$	(51, 74)	侦察机	0	0.11
$U8$	(11, 90)	侦察机	0	0.11
$U9$	(73, 76)	战斗机	5	0.16
$U10$	(69, 20)	战斗机	10	0.10
$U11$	(52, 46)	战斗机	6	0.12
$U12$	(46, 98)	战斗机	7	0.13
$U13$	(92, 86)	战斗机	5	0.09
$U14$	(79, 4)	战斗机	7	0.11

References

1 Du Yong-Hao, Xing Li-Ning, Cai Zhao-Quan. Summary of intelligent dispatching technology of unmanned aerial vehicle cluster. *Acta Automatica Sinica*, 2020, **46**(2): 222−241
(杜永浩, 邢立宁, 蔡昭权. 无人飞行器集群智能调度技术综述. 自动化学报, 2020, **46**(2): 222−241)

2 Deng Q B, Yu J Q, Wang N F. Cooperative task assignment of multiple heterogeneous unmanned aerial vehicles using a modified genetic algorithm with multi-type genes. *Chinese Journal of Aeronautics*, 2013, **26**(5): 1238−1250

3 Jia Gao-Wei, Wang Jian-Feng. Overview of research on UAV cluster mission planning methods. *Systems Engineering and Electronics*, 2021, **43**(1): 99−111
(贾高伟, 王建峰. 无人机集群任务规划方法研究综述. 系统工程与电子技术, 2021, **43**(1): 99−111)

4 Shima T, Rasmussen S J, Sparks A G, Passino K M. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers & Operations Research*, 2006, **33**(11): 3252−3269

5 Huang Gang, Li Jun-Hua. Multi-UAV cooperative target allocation based on AC-DSDE evolutionary algorithm. *Acta Automatica Sinica*, 2021, **47**(1): 173−184
(黄刚, 李军华. 基于 AC-DSDE 进化算法多 UAVs 协同目标分配. 自动化学报, 2021, **47**(1): 173−184)

6 Huang L W, Qu H, Zuo L. Multi-type UAVs cooperative task allocation under resource constraints. *IEEE Access*, 2013, **6**: 17841−17850

7 Ye F, Chen J, Tian Y, Jiang T. Cooperative task assignment of a heterogeneous multi-UAV system using an adaptive genetic algorithm. *Electronics*, 2020, **9**(4): 687−703

8 Xu G T, Liu L, Long T, Wang Z, Cai M. Cooperative multiple task assignment considering precedence constraints using multi-chromosome encoded genetic algorithm. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference. Kissimmee, USA: 2018. 1859−1867

9 Lei C, He S T, Hui P, Ming C P. Multiple UAVs hierarchical dynamic task allocation based on PSO-FSA and decentralized auction. In: Proceedings of the IEEE International Conference on Robotics and Biomimetics. Bali, Indonesia: IEEE, 2014. 2368−2373

10 Han Bo-Wen, Yao Pei-Yang, Sun Yu. UAVS cooperative task allocation based on multi-objective MSQPSO algorithm. *Acta Electronica Sinica*, 2017, **45**(8): 1856−1863
(韩博文, 姚佩阳, 孙昱. 基于多目标 MSQPSO 算法的 UAVS 协同任务分配. 电子学报, 2017, **45**(8): 1856−1863)

11 Wang Feng, Zhang Heng, Han Meng-Chen, Xing Li-Ning. Hybrid variable multi-objective particle swarm optimization algorithm based on co-evolution to solve UAV cooperative multi-task assignment problem. *Chinese Journal of Computers*, 2021, **44**(10): 1967−1983
(王峰, 张衡, 韩孟臣, 邢立宁. 基于协同进化的混合变量多目标粒子群优化算法求解无人机协同多任务分配问题. 计算机学报, 2021, **44**(10): 1967−1983)

12 Du Yong-Hao, Xing Li-Ning, Yao Feng, Chen Ying-Guo. Survey on models, algorithms and general techniques for spacecraft mission scheduling. *Acta Automatica Sinica*, 2021, **47**(12): 2715−2741
(杜永浩, 邢立宁, 姚锋, 陈盈果. 航天器任务调度模型、算法与通用求解技术综述. 自动化学报, 2021, **47**(12): 2715−2741)

13 Shi Y, Eberhart R. A modified particle swarm optimizer. In: Proceedings of the IEEE International Conference on Evolutionary Computation. Anchorage, USA: IEEE, 1998. 69−73

14 Liang Guo-Qiang, Kang Yu-Hang, Xing Zhi-Chuan, Yin Gao-Yang. Cooperative multitask assignment of UAV based on discrete particle swarm optimization. *Computer Simulation*, 2018, **35**(2): 22−28
(梁国强, 康宇航, 邢志川, 尹高扬. 基于离散粒子群优化的无人机协同多任务分配. 计算机仿真, 2018, **35**(2): 22−28)

15 Deng Ke, Lian Zhen-Jiang, Zhou De-Yun, Li Xiao-Yang. Multi-UAV task allocation based on improved quantum particle

swarm algorithm. *Command Control and Simulation*, 2018, **40**(5): 32−36
(邓可, 连振江, 周德云, 李枭扬. 基于改进量子粒子群算法的多无人机任务分配. 指挥控制与仿真, 2018, **40**(5): 32−36)

16 Wang J F, Jia G W, Lin J C, Hou Z X. Cooperative task allocation for heterogeneous multi-UAV using multi-objective optimization algorithm. *Journal of Central South University*, 2020, **27**(2): 432−448

17 Tian Zhen, Wang Xiao-Fang. Cooperative multiple task assignment for heterogeneous multi-UAVs with multi-chromosome genetic algorithm. *Flight Dynamics*, 2019, **37**(1): 39−44
(田震, 王晓芳. 基于多基因遗传算法的异构多无人机协同任务分配. 飞行力学, 2019, **37**(1): 39−44)

18 He C, Cheng R, Tian Y, Zhang X Y, Tan K C, Jin Y C. Paired offspring generation for constrained large-scale multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2020, **25**(3): 448−462

19 Wang F, Li Y, Zhou A M, Tang K. An estimation of distribution algorithm for mixed-variable newsvendor problems. *IEEE Transactions on Evolutionary Computation*, 2019, **24**(3): 479−493

20 Chen W N, Jia Y H, Zhao F, Luo X N, Jia X D, Zhang J. A cooperative co-evolutionary approach to large-scale multisource water distribution network optimization. *IEEE Transactions on Evolutionary Computation*, 2019, **23**(5): 842−857

21 Menchaca-Méndez A, Montero E, Antonio L M, Zapotecas-Martínez S, Coello C A C, Riff M C. A co-evolutionary scheme for multi-objective evolutionary algorithms based on ϵ-dominance. *IEEE Access*, 2019, **7**: 18267−18283

22 Zheng F, Zecchin A C, Simpson A R. Self-adaptive differential evolution algorithm applied to water distribution system optimization. *Journal of Computing in Civil Engineering*, 2013, **27**(2): 148−158

23 Cervantes-Culebro H, Cruz-Villar C A, Peñaloza M G M, Mezura-Montes E. Constraint-handling techniques for the concurrent design of a five-bar parallel robot. *IEEE Access*, 2017, **5**: 23010−23021

24 He C, Cheng R, Zhang C, Tian Y, Chen Q, Yao X. Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers. *IEEE Transactions on Evolutionary Computation*, 2020, **24**(5): 868−881

25 Ma Z, Wang Y, Song W. A new fitness function with two rankings for evolutionary constrained multiobjective optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, **51**(8): 5005−5016

26 Wang B C, Li H X, Li J P, Wang Y. Composite differential evolution for constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018, **49**(7): 1482−1495

27 Zhou Y L, Zhu M, Wang J H, Zhang Z Z, Xiang Y, Zhang J. Tri-goal evolution framework for constrained many-objective optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018, **50**(8): 3086−3099

28 Edison E, Shima T. Integrated task assignment and path optimization for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers & Operations Research*, 2011, **38**(1): 340−356

29 Wang Ling, Shen Jing-Nan, Wang Sheng-Yao, Deng Jin. Advances in co-evolutionary algorithms. *Control and Decision*, 2015, **30**(2): 193−202
(王凌, 沈婧楠, 王圣尧, 邓瑾. 协同进化算法研究进展. 控制与决策, 2015, **30**(2): 193−202)

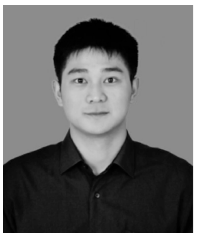
30 Zhan Z H, Li J J, Cao J N, Zhang J, Chung H S H, Shi Y H. Multiple populations for multiple objectives: A coevolutionary technique for solving multi-objective optimization problems. *IEEE Transactions on Cybernetics*, 2013, **43**(2): 445−463

31 Liu X F, Zhan Z H, Gao Y, Zhang J, Kwong S, Zhang J. Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2018, **23**(4): 587−602

32 Cai X, Hu M, Gong D W, Guo Y N, Zhang Y, Fan Z, et al. A decomposition-based coevolutionary multiobjective local search for combinatorial multi-objective optimization. *Swarm & Evolutionary Computation*, 2019, **49**: 178−193

王 峰 武汉大学计算机学院教授. 主要研究方向为进化计算, 智能信息处理和机器学习. 本文通信作者.
E-mail: fengwang@whu.edu.cn
(**WANG Feng** Professor at the School of Computer Science, Wuhan University. Her research interest covers evolutionary computation, intelligent information processing, and machine learning. Corresponding author of this paper.)

黄子路 武汉大学计算机学院硕士研究生. 主要研究方向为进化计算.
E-mail: huangzilu@whu.edu.cn
(**HUANG Zi-Lu** Master student at the School of Computer Science, Wuhan University. His main research interest is evolutionary computation.)



韩孟臣 武汉大学计算机学院硕士研究生. 主要研究方向为进化计算.
E-mail: hanmengchen@whu.edu.cn
(**HAN Meng-Chen** Master student at the School of Computer Science, Wuhan University. His main research interest is evolutionary computation.)

邢立宁 国防科技大学系统工程学院研究员. 主要研究方向为智能优化理论, 方法和应用.
E-mail: xinglining@gmail.com
(**XING Li-Ning** Professor at the College of Systems Engineering, National University of Defense Technology. His research interest covers intelligent optimization theory, method, and application.)

王 凌 清华大学自动化系教授. 主要研究方向为智能优化, 生产调度.
E-mail: wangling@tsinghua.edu.cn
(**WANG Ling** Professor in the Department of Automation, Tsinghua University. His research interest covers intelligent optimization and production scheduling.)