

基于改进沙猫算法的多机协同任务预分配

周世曦^{1,2}, 张建东¹, 史国庆¹, 杨啟明¹, 曹毅¹, 成舒圆¹

(1.西北工业大学 电子信息学院, 陕西 西安 710072; 2.西安飞行自动控制研究所, 陕西 西安 710076)

摘要:针对异构平台协同任务预分配问题,在沙猫群算法基础上引入了非线性收敛因子和外部存储集更新策略,并将其与大规模邻域搜索算法相结合,设计了包含破坏算子和修复算子的改进沙猫算法。为验证该算法的性能,设计了2个仿真案例。仿真案例1基于带时间窗的车辆路径问题,与遗传算法、狐猴算法和鲸鱼算法进行对比。仿真案例2设计了一个多异构有人/无人机的对地侦察/攻击任务分配场景,进一步验证了算法的有效性。

关键词:多目标分配;沙猫算法;多机协同
中图分类号:TP18;V279 **文献标志码:**A

文章编号:1000-2758(2026)01-0102-10

现代战场环境充满了不确定性和复杂性,任务需求各不相同,而飞机的性能也存在显著差异^[1],这使得任务预分配面临着巨大挑战。任务预分配不仅需要在广阔的解空间中寻找最优或近优解,而且还需要满足多个约束条件,如任务的优先级、资源的分配限制、平台的性能特征等^[2-3]。由于任务需求的多样性和飞机性能特点各异,模型可能展现出多个潜在的局部最优值,进而使得求解流程变得复杂且计算成本上升。然而,在任务预分配阶段,得益于较为宽松的时间安排,研究者得以深入进行计算与评估。因此,他们更加重视构建合理的分配策略^[4-6],旨在确保任务执行的高效性以及平台资源的最优化部署。任务预分配阶段的优化方案将直接影响后续任务动态分配的效果,因此,对预分配方案的优化研究至关重要。

为了更系统、更高效地解决多机协同任务预分配问题,许多学者已经采用了多种数学模型和求解方法。例如,整数规划模型^[7]、旅行商问题(TSP)模型^[8-9]、资源分配模型^[10-11]等方法被广泛应用于任务分配研究。群体智^[12-15]凭借其快速的收敛速度和强大的种群多样性保持能力,在多目标组合优化问题的求解领域内得到了广泛应用。

针对任务预分配问题,将大规模邻域搜索算法(large neighborhood search, LNS)和沙猫群算法(sand cat swarm optimization, SCSO)结合,提出了改进沙猫算法(LNSCSO)。首先分析了沙猫群算法的基本原理,并指出其易早熟和种群多样性降低等问题。为此,引入了非线性收敛因子和外部存储集更新策略,并将其与大规模邻域搜索算法结合,添加了破坏算子和修复算子。最后在带时间窗的车辆路径问题和多异构有人/无人机的对地侦察/攻击任务分配场景中进行验证。仿真结果表明,改进沙猫群算法在任务分配问题中具有较强的有效性。

1 多机协同任务分配

1.1 问题描述

设飞机集合 U 由 M 架飞机组成,表示为 $U = \{U_i | i = 1, 2, \dots, M\}$,其中 U_i 代表第 i 架飞机。设任务集合 T ,包含 N 个任务,记作 $T = \{T_i | i = 1, 2, \dots, N\}$, T_i 代表第 i 个任务。

我方飞机划分为侦察与攻击2类,任务也分为侦察和攻击2种。在执行任务时,不同飞机在执行相同或不同任务时,收益和威胁均有所不同,这导致任务分配的结果可能多种多样。在多机协同任务分配流程中,首要步骤是全面搜集战场敌我双方的信息。然后,通过任务效能评估模型计算和判断每架飞机执行特定任务的适合程度。最后,依据这些计

收稿日期:2025-05-21

作者简介:周世曦(2000—),硕士研究生

通信作者:史国庆(1974—),副教授

e-mail:shiguqing@nwpu.edu.cn

算结果,生成一系列任务执行顺序的候选方案。多机协同任务分配模型框图如图 1 所示。

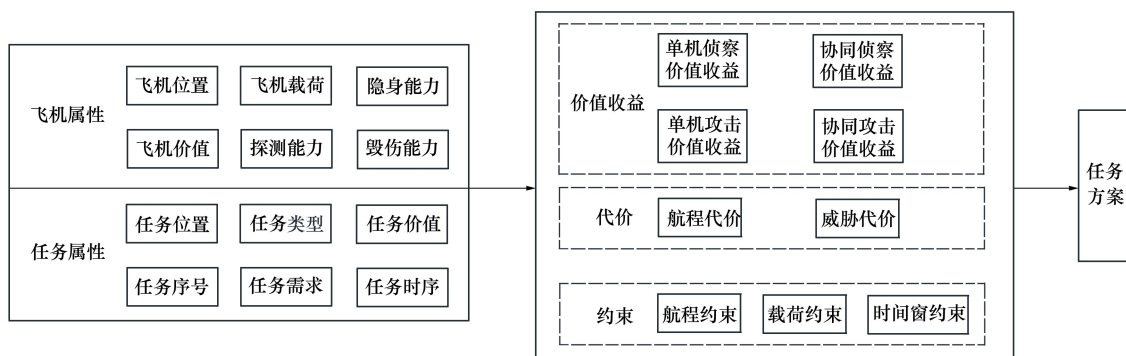


图 1 多机协同任务分配模型框图

1.2 任务分配评价指标

本文所提出的任务分配模型是在综合考虑毁伤概率、任务价值、任务时序等多个因素的基础上构建的。每一项因素都直接影响任务执行的效果和风险,因此必须全面评估并合理权衡。为便于后续表达,定义了相应的决策变量 X_{ij} 。

$$X_{ij} = \begin{cases} 0 & i = 1, 2, \dots, M; j = 1, 2, \dots, N \\ 1 & \end{cases} \quad (1)$$

式中: X_{ij} 代表飞机 U_i 执行任务 T_j 的决策变量; $X_{ij} = 0$ 代表飞机 U_i 不执行任务 T_j ; $X_{ij} = 1$ 代表飞机 U_i 执行任务 T_j 。

针对任务集合 T 中的任意元素 T_j , 其属性集包括 $I_T, L_T, V_T, S_T, Q_T, T_T, N_T$ 。 I_T 是任务的唯一身份标识; L_T 指明了任务发生的地理坐标; V_T 评估了任务的重要性或潜在价值; S_T 是一个矩阵, 用于定义任务间的执行先后关系, 其中 $S_T = 1$ 表明 T_j 紧随 T_i 之后执行, 若 $S_T = 0$ 则无特定顺序要求; 飞机需求 Q_T 代表了执行该任务所需的最少飞机数; 类型 T_T 将任务分为 2 类, 1 代表侦察, 2 代表攻击; 有效性 N_T 是一个标志位, $N_T = 1$ 意味任务处于有效状态, $N_T = 0$ 则表明任务无效。

针对集合 U 中的任意飞机实例 U_j , 其属性集包括 $I_U, L_U, V_U, D_U, W_U, P_U^D, P_U^W, P_U^I$ 。飞机编号 I_U 是每架飞机的唯一身份标识; 坐标位置 L_U 表明了飞机的当前地理位置; 飞机价值 V_U 是对飞机整体性能的评估; 最大侦察任务数 D_U 和最大攻击任务数 W_U 分别限定了飞机可执行侦察与攻击任务的最大数量; 目标探测概率集 P_U^D 详细列出了飞机对各个潜在目标的探测能力; 目标毁伤概率集 P_U^W 则揭示了飞机对各目标的破坏能力; 任务存活概率集 P_U^I 综合反映了

飞机在执行各类任务时面对不同威胁的生存几率。任务分配的作战收益、威胁代价、航程代价、总体效能数学模型如下所示。

1) 多机协同作战收益

飞机 U_i 完成侦察任务 T_j , 获得的收益 R_{ij}^D 为

$$R_{ij}^D = P_{U,i}^D \times V_{T,j} \quad (2)$$

飞机 U_i 完成攻击任务 T_j , 获得的收益 R_{ij}^W 为

$$R_{ij}^W = P_{U,i}^W \times V_{T,j} \quad (3)$$

当 m 架飞机, 编号为 U_i, i 取 $1, 2, \dots, m$ 均参与执行侦察任务 T_j 时, 它们分配任务后得到的侦察效益表示为 C_j^D 。

$$C_j^D = \left(1 - \prod_{i=1}^m (1 - P_{U,i}^D) \right) \times V_{T,j} \quad (4)$$

当 m 架飞机, 编号为 U_i, i 取 $1, 2, \dots, m$, 共同参与并执行任务 T_j 时, 这些飞机在任务分配后所能获得的收益记为 C_j^W 。

$$C_j^W = \left(1 - \prod_{i=1}^m (1 - P_{U,i}^W) \right) \times V_{T,j} \quad (5)$$

多机执行对地作战任务中的收益 R_T 为

$$R_T = \sum_{j=1}^N (C_j^D + C_j^W) + \sum_{i=1}^M \sum_{j=1}^N (R_{ij}^D + R_{ij}^W) \quad (6)$$

2) 多机协同作战威胁代价

当飞机 U_i 执行任务 $T_j (j = 1, 2, \dots, r)$ 时, 飞机 U_i 执行任务的威胁代价 C_i^k 为

$$C_i^k = \left(1 - \prod_{j=1}^r P_{U,i}^I \right) \times V_{U,i} \quad (7)$$

类比任务分配的总收益, 总体威胁代价 C_T^k 具体形式为

$$C_T^k = \sum_{i=1}^M C_i^k \quad (8)$$

3) 多机协同作战航程代价

假设飞机 U_i 执行任务 $T_j(j=1,2,\dots,r)$ 时, j 为飞机执行的顺序并且任务之间没有障碍, 单架飞机执行任务集合的航程代价 C_i^d 为

$$C_i^d = \sqrt{(L_{U,i}^x - L_{T,j_1}^x)^2 + (L_{U,i}^y - L_{T,j_1}^y)^2} + \sum_{k=j_1}^{j_r-1} \sqrt{(L_{T,k+1}^x - L_{T,k}^x)^2 + (L_{T,k+1}^y - L_{T,k}^y)^2} \quad (9)$$

类比可知总体航程代价 C_T^d 为

$$C_T^d = \sum_{i=1}^M C_i^d \quad (10)$$

在实战情境下, 飞行路线的成本会受到禁飞区、山地风险、防空系统等多种因素的制约。为应对这一问题, 构建了一个 $(M+N) \times N$ 的矩阵, 其初始赋值暂以飞机与任务、或任务之间的直接几何距离作为替代。随着算法迭代及航迹规划工作的深入, 矩阵会根据实际飞行中对障碍物的规避情况不断更新, 从而确保飞行成本估算的精确性。

4) 飞机协同作战的总体效能

多机协同作战鉴于任务效益、威胁损耗及飞行路程等因素间的内在冲突, 导致最终的最优方案并非固定不变。为此, 本文拟采用加权策略, 将多个性能指标整合成一个统一的优化目标。鉴于这些指标的量纲各不相同, 需要先进行归一化处理, 以确保它们在同一基准上进行比较。在此基础上, 整体效能的评估公式为

$$\max(E) = \omega_1 \frac{R_T}{V_{T,m}} + \omega_2 \left(1 - \frac{C_T^k}{V_{U,m}}\right) + \omega_3 \left(1 - \frac{C_T^d}{d_m}\right) \quad (11)$$

在此模型中, $\omega_1, \omega_2, \omega_3$ 被设定为任务总效益、威胁代价及飞行总距离的权重因子, 它们的具体数值由指挥决策者根据任务需求进行分配, 且必须满足总和为 1 的条件。 $V_{T,m}$ 代表任务集合内任务价值的最高值; $V_{U,m}$ 为飞机集合中性能最优飞机的价值; d_m 是利用凸包算法得出的最小包围周长, 作为评估飞行距离的一个基准值。

1.3 任务分配约束条件

为了更精确地模拟战场实际情况, 本文设计了以下约束条件模型。

1) 飞机任务承载量限制

鉴于飞机在执行任务时的载荷与性能限制, 每架飞机可执行的任务数量被设定在一定范围内。因此, 引入了任务承载量上限的约束, 确保每架飞机的任务分配量不超过其最大承载能力。

$$\begin{cases} \sum_{j=1}^N X_{ij} \leq D_{U,i} (T_j \text{ 为侦察任务}) \\ \sum_{j=1}^N X_{ij} \leq W_{U,i} (T_j \text{ 为攻击任务}) \end{cases} \quad (12)$$

2) 任务目标所需飞机数量约束

指挥中心根据任务类型, 确定每个任务需要多少架飞机参与执行。因此, 对于每个任务, 都有一个明确的飞机数量需求。该约束确保每个任务能够分配足够数量的飞机。

$$\sum_{i=1}^M X_{ij} = C_{T,j} \quad (13)$$

3) 飞机最大航程约束

由于飞机之间存在性能差异, 且每架飞机的油量有限, 因此需要保证飞机在完成的任务后, 能够安全返回机场。此约束确保飞机在执行任务后有足够的油量, 以完成往返航程并安全着陆。

$$C_i^d \leq R_{U,i} \quad (14)$$

2 改进沙猫算法

2.1 SCSO 算法

沙猫群算法 (SCSO) 由 Seyyedabbasi 等^[16] 研究者在 2022 年首次提出。该算法通过模拟沙猫在沙漠中捕猎和觅食的行为, 设计了一种高效的优化方法。沙猫群算法可分为探索阶段和开发阶段 2 个阶段^[17-18]。

1) 搜索阶段

沙猫能够感知并响应低于 2 kHz 的低频声波, 这使其在夜间或在视距较远的情况下寻找猎物时具有显著的优势。结合这种特性, 算法对沙猫个体灵敏度范围进行设计。在寻找猎物的搜索阶段, 根据其低频感知能力, 假设沙猫灵敏度范围从 2 kHz 到 0, 搜索时沙猫利用随机向量 \mathbf{R} 来指导其搜索过程。当 $|\mathbf{R}| > 1$ 时, 表明沙猫检测到的低频信号强度已经超过了某一临界值, 从而促使其进行进一步的猎物搜索。听觉范围 r_G 为

$$r_G = s_M - \left(\frac{s_M \times i_c}{i_m}\right) \quad (15)$$

式中: s_M 为初始收敛因子最大值, 由于 s_M 值受到沙猫听力特征的启发, 因此假设其值为 2。 i_c 代表种群当前迭代次数, i_m 代表种群最大迭代次数。

在沙猫算法的搜索过程中, \mathbf{R} 作为一个关键的引导参数, 决定了沙猫在当前迭代中是进行全局探

索(搜索)还是局部聚焦(开发)行为。随着迭代次数的增加,参数 R 会根据沙猫当前位置的动态变化进行调整,从而实现从广泛搜索到精确攻击的平滑过渡。 R 的更新方式为

$$R = 2 \times r_c \times \text{rand}(0,1) - r_c \quad (16)$$

在算法初始化阶段,各沙猫个体的搜索空间被随机设定在预设的边界范围内。搜索过程中,个体位置更新依据随机选取的参考点进行动态调整。为避免算法陷入局部最优,系统为每个个体设置了差异化的灵敏度参数: r 表征个体特定的灵敏度阈值,而 r_c 则表示全局基准灵敏度范围。参数 r 作用于搜索与攻击阶段的局部优化过程,而 r_c 则通过调节全局引导因子 R 来协调不同阶段间的转换机制。这种双层级灵敏度调控策略既保障了个体探索的自主性,又维持了种群整体的协同演化方向。

$$r = r_c \times \text{rand}(0,1) \quad (17)$$

沙猫算法在更新其位置时,采用了一种基于全局最优解 $P_b(t)$ 、自己当前位置 $P_c(t)$ 以及灵敏度范围 r 的策略,从而有效地引导沙猫朝着猎物的位置靠近。

$$P(t+1) = r \cdot (P_b(t) - \text{rand}(0,1) \cdot P_c(t)) \quad (18)$$

2) 开发阶段

沙猫攻击时的活动范围为圆形,其移动导向由圆上任一随机角度 θ 确定,该角度 θ 随机分布在 $0^\circ \sim 360^\circ$ 间,对应数值范围从 $-1 \sim 1$,在探索区域内,群体内的每只沙猫都能按照各自独特的圆周轨迹移动。SCSO 算法借助轮盘赌选择策略,为每只沙猫分配一个随机角度,从而避开局部最优解。位置更新如图 2 所示,其位置更新策略为:

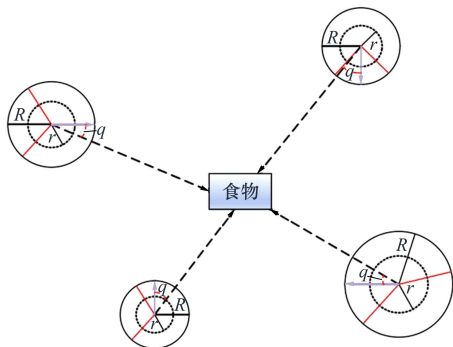


图 2 迭代中的位置更新

$$P_r = |\text{rand}(0,1) \cdot P_b(t) - P_c(t)| \quad (19)$$

$$P(t+1) = P_b(t) - r \cdot P_r \cdot \cos\theta \quad (20)$$

3) 搜索和开发阶段

搜索与开发阶段的切换通过自适应参数 r_c 和引导因子 R 协同实现。其中, R 的数学期望与 r_c 呈线性关联,其变化幅度随 r_c 的衰减而逐步收敛。具体而言, R 的取值范围被约束为 $[-2r_c, 2r_c]$,而 r_c 本身在迭代过程中由初始值 2 线性递减至 0。当 $|R| \leq 1$ 时,算法进入攻击模式,此时沙猫的位置更新策略调整为

$$P(t+1) = P_b(t) - r \cdot P_r \cdot \cos\theta \quad (21)$$

在引导参数 $|R| > 1$ 时进行搜索,其更新方式如(22)式所示。

$$P(t+1) = r \cdot (P_b(t) - \text{rand}(0,1) \cdot P_c(t)) \quad (22)$$

2.2 非线性收敛因子

在 SCSO 中,算法的搜索和开发过程主要通过收敛因子 r_c 来进行控制。LNCSO 算法流程图如图 3 所示。

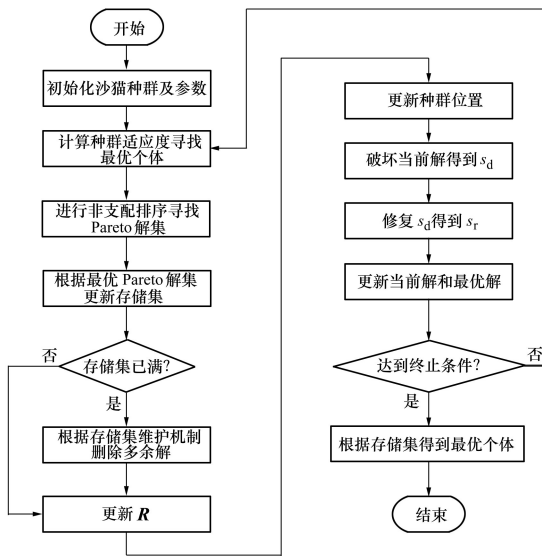


图 3 LNCSO 算法流程图

在传统方法中,收敛因子往往线性衰减,这意味着随着迭代次数增加,收敛因子值逐渐减小。然而,这种线性衰减方式可能无法充分挖掘搜索空间中的潜在解,容易导致搜索范围过于狭窄。本文提出了一种非线性收敛因子 r_c 的设计方案,具体表达式为

$$r_c = s_m - \left(s_m \times \left(\frac{t}{t_m} \right)^2 \right) \quad (23)$$

式中: s_m 为初始收敛因子的最大值; t 为当前的迭代

次数; t_m 为最大迭代次数。这种非线性设计使得收敛因子的变化不再是单纯的线性衰减,收敛因子在算法初期会保持较高值,从而增强探索能力;随着迭代次数增加,收敛因子逐渐减小,促进算法在解空间中聚焦于潜在的最优解。

2.3 外部存储集更新

在 LNSCSO 算法的迭代升级中,个体位置的更新始终围绕全局最优个体的选定展开,这一选择对 LNSCSO 的寻优性能与搜索速度具有重要影响。外部存储集更新策略借助非支配排序方法从种群中筛选出最优的 Pareto 前沿解集,随后将其整合至外部存储库中。随着种群的迭代更新,新的 Pareto 前沿解集会不断被添加到外部存储库,并重新进行非支配排序,以保障所有非支配解得以保留。为了有效管理外部存储库的大小,一旦存储库中的个体数目超出预设的最大容量,本研究设计了一种轮盘赌选择策略,依据个体对应的概率 p_i 决定并剔除冗余的个体。概率 p_i 的计算公式为

$$p_i = \frac{e^{-2n_i}}{\sum_{i \in N} e^{-2n_i}} \quad (24)$$

式中: n_i 指围绕第 i 个解决方案的相邻解决方案数量。对于全局最优解的挑选,采用轮盘赌选择机制,并依据特定的概率 q_i 完成,概率 q_i 的计算公式为

$$q_i = \frac{e^{2n_i}}{\sum_{i \in N} e^{2n_i}} \quad (25)$$

通过该机制,能够在外部储存集个体中根据拥挤度进行选择,使得较为拥挤的区域能够被有效减少,从而保持解集的多样性,并提高全局搜索的效率。

2.4 破坏算子和修复算子

大规模邻域搜索算法(LNS)在处理复杂的、规模较大的问题时,能够有效地探索解空间并寻找近似最优解^[19-20]。LNS 的核心思想是在当前解的基础上对其进行局部修改,从而探索更大的解空间,并在此过程中通过“破坏”当前解并“修复”其结构来改善解的质量。

破坏因子控制着从当前解中移除或修改部分元素的比例,从而“破坏”当前解的结构。破坏的目标是将当前解的某些部分进行随机删除或调整,以创造出一个新的解空间,避免陷入局部最优解。根据相似性,破坏算子从路径中剔除若干相似性较高的顾客,计算公式为

$$R(i,j) = \frac{1}{c'_{ij} + V_{ij}} \quad (26)$$

式中, c'_{ij} 为将 c_{ij} 标准化后的值,范围在 $[0,1]$; c_{ij} 为 i 与 j 之间的欧氏距离。

$$c'_{ij} = \frac{c_{ij}}{\max c_{ij}} \quad (27)$$

当任务点 i 和任务点 j 位于同一路径时, V_{ij} 值为 0;若不在同一路径,则 V_{ij} 值为 1。 $R(i,j)$ 用于衡量任务点和任务点 j 之间的关联度, $R(i,j)$ 越大,说明两者之间的关联性越强。设定总任务点数为 N ,需移除的任务点数为 q ,并引入随机变量 D 。破坏算子的伪代码如算法 1 所示。

算法 1 破坏算子的伪代码

输入: 当前解 S , 要移除的任务点数目 q , 随机元 D
 输出: 破坏后的解 S_d , 移除任务点的集合 I

- 1: while $|I| > q$ do
- 2: 随机选择任务点 i_c, i_c 属于集合 I
- 3: $i < j \Rightarrow R(i_c, L[i]) < R(i_c, L[j])$
- 4: $k \rightarrow [\text{rand}^D |L|]$ // 计算随机选择任务点序号
- 5: $I \rightarrow I \cup \{L_k\}$
- 6: end while
- 7: return S_d 和 I

修复算子在符合弹药装载约束和时间窗约束的前提下,将之前移除的客户重新分配到最合适的插入位置,从而最小化飞机行驶总距离。修复算子伪代码如算法 2 所示。

算法 2 修复算子的伪代码

输入: 破坏后的解 S_d , 移除任务点的集合 I
 输出: 修复后的解 S

- 1: 从解 S 中随机选出任务点 i_s , 并把 i_s 放入集合 I 中
- 2: $S \leftarrow S_d$
- 3: while $|I| > 0$ do
- 4: 计算 I 中任务点的最小插入成本 $z_i = \min \Delta f_{i,k}$ // $\min \Delta f_{i,k}$ 是距离增量
- 5: 计算 z_i 对应的插回路径编号 r_i 和该路径上的插回位置 p_{os_i}
- 6: $i < j \Rightarrow R(i_c, L[i]) < R(i_c, L[j])$ // 将排序结果存储在排序序列 L 中
- 7: 若无路径添加任务点 i , 新建路径添加任务点 i
- 8: 从 I 中选择 $\max z_i$ 的任务点 i_s

- 9: 将任务点 i_z 插入到 S 中第 $r_{i,z}$ 条路径上第 $p_{os_{i,z}}$ 个位置
- 10: 将任务点 i_z 从 I 中移除
- 11: end while
- 12: return S

破坏因子决定了当前解的破坏程度,进而决定了算法在解空间中的探索范围。修复因子则用于修复破坏后的解,确保修复后的解合法并且具备改进的潜力。破坏因子和修复因子在大规模邻域搜索算法中往往是紧密结合的,二者的相互作用决定了算法的搜索策略和局部搜索能力。

2.5 算法性能对比

为了验证改进沙猫算法在解决 VRPTW 优化问题中的有效性与优越性,本文采用了遗传算法、鲸鱼算法以及狐猴算法进行对比实验。这些算法均添加破坏算子和修复算子,以提高在复杂问题求解中的表现。在实验设计中,各个算法的种群规模均设置为 50,最大迭代次数设为 200。为了消除初始条件对实验结果的干扰,所有算法均采用相同的编码方式,并随机生成初始种群。

同一仿真实验中,4 种不同算法分别经 200 次迭代,以车辆行驶总距离为算法适应度值,最终得到了各自的适应度变化曲线,如图 4 所示。通过对比分析,改进的沙猫算法在初期迭代中表现出较快的适应度提升,而在后期阶段逐渐收敛至较优解,显示出较强的全局搜索能力。实验结果充分体现了各算法在不同阶段的优化能力,以及其在特定问题条件下的适应性和有效性。

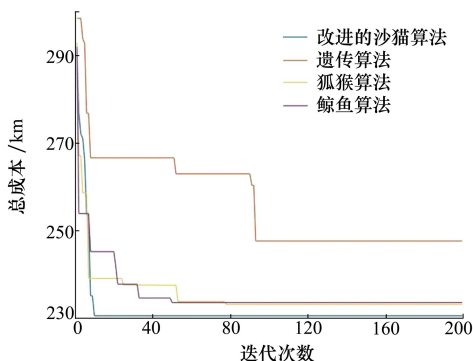


图 4 各类算法适应度变化曲线

由于智能算法本身具有一定的随机性,为了避免单次仿真实验中可能出现的偶然性影响,对上述 4 种算法进行了 30 次独立的仿真实验。最优解是

指算法最终得到的路径方案对应的最小行驶距离值,衡量了路径优化的质量。平均收敛代数表示算法在找到最优解过程中所经历的平均迭代次数,反映了算法的收敛速度。具体的统计数据如表 1 所示。

表 1 求解 VRPTW 问题的仿真结果对比

算法	最优解/km	平均收敛代数
改进沙猫	230.788 3	15.72
遗传	247.781 8	95.35
鲸鱼	233.786 4	52.42
狐猴	233.340 4	79.54

由各类算法在仿真实验中的结果可知,算法均在规定的迭代次数内达到收敛。对于加入破坏算子和修复算子的对比算法,如遗传算法、狐猴算法和鲸鱼算法,它们能够有效地提升搜索结果并确保在有限的迭代代数内收敛。但这些方法在增强整体搜索效能上有所欠缺,特别是在迭代过程后期,它们常常难以跳出局部最优解。因此,这些算法虽然能够在短时间内找到较好的解,但其全局优化能力和避免局部最优的能力仍有待加强。其中,遗传算法在仿真实验中存在收敛速度较慢的问题,尤其在面对需要快速求解的任务分配问题时,这一缺点尤为突出。收敛速度较慢可能会影响算法在时间限制较为严格的应用场景中的表现,导致求解过程不能在规定时间内完成。

相比之下,在给定的迭代次数内,改进沙猫算法找到的最优解比其他算法所找到的最优解更具优势,表现出了更优的优化效果。LNCSO 算法在保证较快收敛速度的同时,增强了个体跳出局部最优解的能力,尤其对于任务分配等离散型优化问题,展现了良好的优化效果。因此,LNCSO 算法不仅能有效避免陷入局部最优解,还能在较短的时间内找到较为优越的解,适合解决带时间窗的任务分配问题。

3 仿真实验与结果分析

3.1 任务分配问题粒子编码设计

LNCSO 算法将每个沙猫个体映射为任务分配方案解,通过群体智能的协同搜索与位置迭代优化实现问题最优解的逼近。动态环境中任务持续时间、目标消失时间及飞行速度的不确定性为异构平

台任务分配带来多重约束。任务执行需满足载荷资源限制、时序逻辑关联以及时间窗匹配等复合约束条件,同时需协调多机间的竞争协作关系以最大化整体作战效能。为了应对这些挑战,本文提出了一种基于实数向量编码方式的优化方法,确保任务分配方案满足以下关键约束条件:

- 1) 时序约束:确保侦察任务与打击任务的时序关系得到遵循;
- 2) 任务并行限制:任何时刻,每架飞机仅能专注于执行单一任务;
- 3) 任务与飞机类型匹配原则:确保每项任务被分配至相应类型的飞机执行。

在此基础上,本文采用了基于实数向量的编码方法,并在此编码方案中建立了沙猫位置与任务分配解的映射关系。沙猫的位置在 LNCSO 算法中由一个实数向量表示,其维度与任务数量相同。每个维度上的数值均代表着特定的任务分配情况,共同作用于整个任务调度的规划之中。

沙猫个体的定位包含整数与小数 2 个组成部分,它们共同界定了任务的分配策略。其中,整数部分 $[M]$ 标识了任务将被指派给的具体飞机编号,而拥有相同整数部分的任务将由同一架飞机来执行。为了保证飞机与任务的匹配性,本文设置了沙猫位置整数部分的上下界,以确保每个任务仅被分配给具有相应能力的飞机类型执行。小数部分 $\{M\}$ 则反映了任务在飞机任务队列中的排序,小数部分值越小,意味着该任务拥有更高的执行优先级,其被执行的时间点也会相对提前。通过这种方式,可以确保任务在各飞机上的执行顺序符合时序约束,特别是侦察任务和打击任务之间的时序依赖。

在任务分配方案的编码过程中,沙猫位置的整数部分定义了任务与飞机的匹配关系,而小数部分则为每架飞机的任务分配顺序。以下是一个具体的示例。假设有 3 架飞机和 7 个任务($T_1 \sim T_7$),沙猫的编码表示如表 2 所示。

根据编码规则,可以通过解码得到每架飞机的任务执行序列。

飞机 P_1 的任务序列为: $T_2 \rightarrow T_5$; 飞机 P_2 的任务序列为: $T_3 \rightarrow T_1 \rightarrow T_6$; 飞机 P_3 的任务序列为: $T_7 \rightarrow T_4$ 。

在解码过程中,任务序列会根据小数部分的值被排序,以确保任务的执行顺序符合实际时序需求。

这样, LNCSO 算法通过不断更新沙猫个体的位置,在多次迭代后搜索到一个最优的任务调度方案。通过这种编码与解码方式, LNCSO 算法能够在不确定环境下,满足异构平台任务分配问题中的时序约束、任务执行约束及匹配约束。

表 2 任务序列编码

任务	M	$[M]$	$\{M\}$	P
T_1	2.5	2	0.5	P_2
T_2	1.3	1	0.3	P_1
T_3	2.2	2	0.2	P_2
T_4	3.7	3	0.7	P_3
T_5	1.5	1	0.5	P_1
T_6	2.6	2	0.6	P_2
T_7	3.2	3	0.2	P_3

3.2 仿真实验与结果分析

验证模型功能后,设计实验对算法性能进行验证,假设我方有 8 架异构有/无人机,它们将共同执行 25 个对地任务,包括侦察任务和攻击任务。本实验采用基于加权评价指标的方法来计算任务分配的适应度值,具体通过任务收益、威胁代价和配送效率 3 项指标的加权组合来衡量。其中,任务收益 w_1 、威胁代价 w_2 和配送距离 w_3 的权重分别为 0.3、0.4 和 0.3。此外,为确保方案的可行性,引入了容量约束 α 和时间窗约束 β 的惩罚系数,分别为 10 和 100,以应对任务超出资源限制或时间窗口违规的情况。利用本节所提出的改进沙猫算法对任务分配模型进行求解,种群规模设置为 50,最大迭代次数为 300。最终的任务分配方案如表 3 所示,分配结果二维示意图如图 5 所示。

表 3 有/无人机对地任务分配方案

飞机编号	任务序列	最优适应度值
P_1	$M_3 \rightarrow M_2$	421.666 3
P_2	$M_{23} \rightarrow M_{14} \rightarrow M_4$	
P_3	$M_{13} \rightarrow M_{19} \rightarrow M_{25} \rightarrow M_1$	
P_4	$M_6 \rightarrow M_{21} \rightarrow M_{20}$	
P_5	$M_7 \rightarrow M_{22} \rightarrow M_{11}$	
P_6	$M_{18} \rightarrow M_{10} \rightarrow M_9 \rightarrow M_{24}$	
P_7	$M_5 \rightarrow M_8 \rightarrow M_{15} \rightarrow M_{12}$	
P_8	$M_{16} \rightarrow M_{17}$	

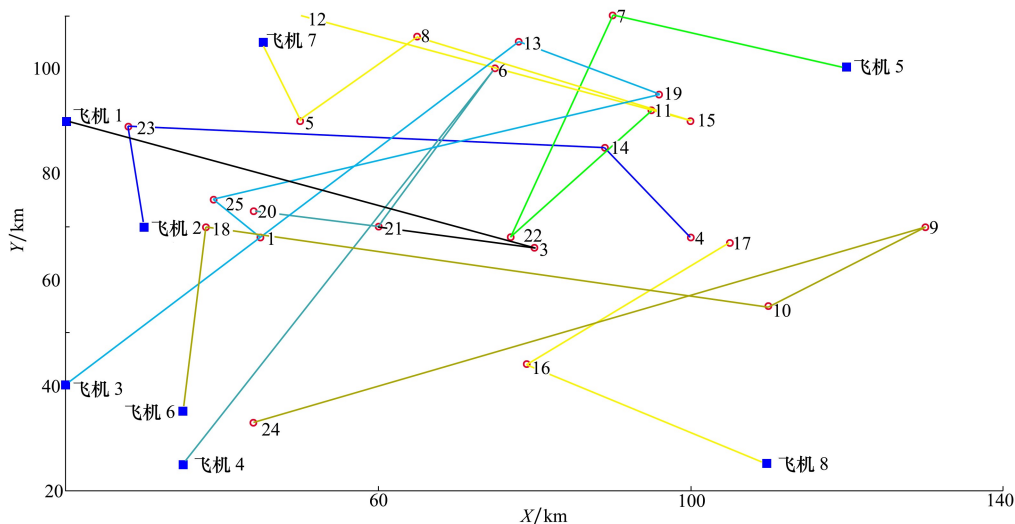


图 5 对地任务分配结果二维示意图

实验结果表明,不同飞机执行的任务序列呈现出明显的差异,这反映了沙猫算法在寻找最优解过程中考虑了任务之间的不同优先级、距离、资源消耗等因素。该方案能够有效平衡任务之间的关系,最大化任务的执行效益,最优适应度值为 421.666 3。算法的适应度变化曲线如图 6 所示。

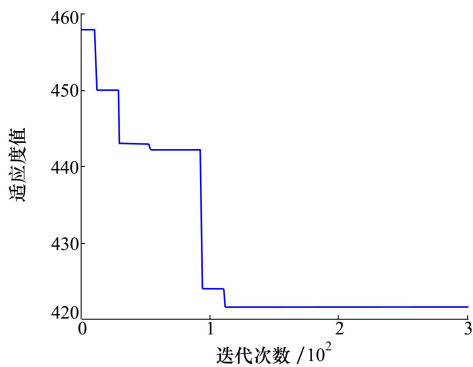


图 6 改进沙猫算法适应度变化曲线

带时间窗的任务甘特图作为一种可视化工具,能有效地展示任务调度的过程和结果。任务甘特图时间轴在横轴上表示,任务的执行按照时间顺序排列。任务条形表示每个任务的开始和结束时间,条形的长度表示任务的执行持续时间。本实验的任务甘特图如图 7 所示。通过图 7 能够直观地观察任务的排程情况以及不同时间段内资源的使用情况。通过观察任务条形的长度和分布,可以帮助分析任务的分布是否均衡,如飞机 1 的任务分布比较分散,而飞机 8 的任务分布比较紧密。

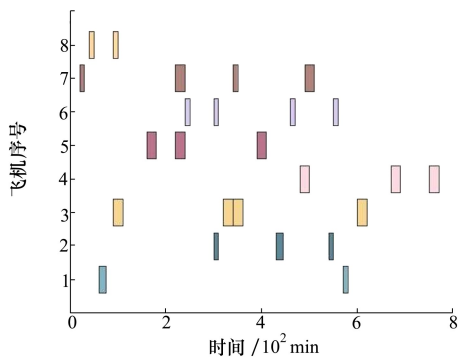


图 7 对地任务分配结果任务甘特图

实验结果表明,改进沙猫算法能够在综合考虑任务收益、威胁代价、配送效率以及约束条件的基础上,平衡各项因素,找出最优的任务分配方案。随着迭代次数的增加,适应度值最终收敛至较优解。在算法的迭代初期,最优适应度呈现出快速下降趋势,这表明算法在搜索过程中迅速排除了较差解并向较优解逼近;而在算法的中后期,收敛速度逐渐减缓,算法进入精细搜索阶段,围绕当前较优个体进行局部搜索,从而逐步逼近全局最优解。最终,经过约 110 代的迭代,算法达到了收敛,此时适应度值稳定,表明最优解已经被有效地确定。

4 结 论

本文对 SCSO 算法的改进,引入了非线性收敛因子和外部存储更新机制,并融合 LNS 算法,通过

整合破坏算子和修复机制,创新性地提出了 LNSCSO 算法。随后,将 LNSCSO 算法应用于解决复杂的带有时间窗的车辆路径问题(VRPTW),并通过与多种智能优化算法的对比,展示了其在处理此类组合优化问题时的显著效果和优势。针对异构多

无人机协同任务预分配的难题,设计了一种基于实数向量的编码策略。仿真实验的结果进一步验证了 LNSCSO 算法在解决任务预分配问题上的有效性和适用性。

参考文献:

- [1] Liang Zibin, Li Qing, Fu Guodong. Multi-UAV collaborative search and attack mission decision-making in unknown environments[J]. *Sensors*, 2023, 23(17): 7398-7418.
- [2] 王新尧,曹云峰,孙厚俊,等.基于 DoDAF 的有人/无人机协同作战体系结构建模[J]. *系统工程与电子技术*, 2020, 42(10): 2265-2274.
Wang Xinyao, Cao Yunfeng, Sun Houjun, et al. Modeling for cooperative combat system architecture of manned/unmanned aerial vehicle based on DoDAF[J]. *Systems Engineering and Electronics*, 2020, 42(10): 2265-2274. (in Chinese)
- [3] Cramer Martijn, Kellens Karel, Demeester Eric. Probabilistic decision model for adaptive task planning in human-robot collaborative assembly based on designer and operator intents[J]. *IEEE Robotics and Automation Letters*, 2021, 6(4): 7325-7332.
- [4] 叶文. 无人飞行器任务规划[M]. 北京: 北京国防工业出版社, 2015.
Ye Wen. Mission planning for unmanned aerial vehicles[M]. Beijing: National Defense Industry Press, 2015. (in Chinese)
- [5] Keller J. DARPA adds two companies to OFFSET swarm reconnaissance drone research project[J]. *Military & Aerospace Electronics*, 2018, 29(4): 30-31.
- [6] Khamis A, Hussein A, Ehnogy A. Multi-robot task allocation: a review of the state-of-the-art[J]. *Studies in Computational Intelligence*, 2015, 604: 31-51.
- [7] 倪谣,周德云,马云红,等.基于 MILP 模型的多无人机对地攻击任务分配[J]. *火力与指挥控制*, 2008, 33(11): 62-65.
Ni Yao, Zhou Deyun, Ma Yunhong, et al. The air-to-ground tasks assignment for multi-UAV based mixed integer linear programming[J]. *Fire Control & Command Control*, 2008, 33(11): 62-65. (in Chinese)
- [8] Xue Yang, Wang Jiasheng. Application of improved ant colony optimization algorithm on traveling salesman problem[C]//Chinese Control and Decision Conference, Yinchuan, China, 2016: 2156-2160.
- [9] Reinelt Gerhard. The traveling salesman: computational solutions for TSP applications[M]. Berlin: Springer-Verlag, 1994.
- [10] Fang Ye, Jie Chen, Yuan Tian, et al. Cooperative multiple task assignment of heterogeneous UAVs using a modified genetic algorithm with multi-type-gene chromosome encoding strategy[J]. *Journal of Intelligent & Robotic Systems*, 2020, 100(2): 1-13.
- [11] Bai Xiaoshan, Li Chang, Zhang Bo, et al. Efficient performance impact algorithms for multirobot task assignment with deadlines [J]. *IEEE Trans on Industrial Electronics*, 2024, 71: 14373-14382.
- [12] 毕文豪,张梦琦,高飞,等.无人机集群任务分配技术研究综述[J]. *系统工程与电子技术*, 2024, 46(3): 922-934.
Bi Wenhao, Zhang Mengqi, Gao Fei, et al. Review on UAV swarm task allocation technology[J]. *Systems Engineering and Electronics*, 2024, 46(3): 922-934. (in Chinese)
- [13] 胡嘉薇,贾泽群,孙延涛,等.多约束条件下多无人机协同任务规划问题分析及求解方法综述[J]. *计算机科学*, 2023, 50(7): 176-193.
Hu Jiawei, Jia Zequn, Sun Yantao, et al. Survey of analysis and solutions for multi-UAV cooperative mission planning problem under multi-constraint conditions[J]. *Computer Science*, 2023, 50(7): 176-193. (in Chinese)
- [14] 毛莺池,穆超,包威,等.空间众包中多类型任务的分配与调度方法[J]. *计算机应用*, 2018, 38(1): 6-12.
Mao Yingchi, Mu Chao, Bao Wei, et al. Multi-type task assignment and scheduling oriented to spatial crowdsourcing[J]. *Journal of Computer Applications*, 2018, 38(1): 6-12. (in Chinese)
- [15] Bai Xiaoshan, Yan Weisheng, Ge Shuzhi Sam. Efficient task assignment for multiple vehicles with partially unreachable target locations[J]. *IEEE Internet of Things Journal*, 2021, 8(5): 3730-3742.
- [16] Seyyedabbasi Amir, Kiani Farzad. Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems [J]. *Engineering with Computers*, 2022, 39(4): 2627-2651.

- [17] 胡坤, 莫愿斌. 混合策略的沙猫算法求解化工动态优化问题[J]. 化学工程, 2024, 52(5): 85-90.
Hu Kun, Mo Yuanbin. Hybrid sand cat algorithm for solving dynamic optimization problems in chemical engineering[J]. Chemical Engineering, 2024, 52(5): 85-90. (in Chinese)
- [18] Oluwatayomi Rereloluwa Adegboye, Afi Kekeli Feda, Oluwaseun Racheal Ojekemi, et al. DGS-SCSO: enhancing sand cat swarm optimization with dynamic pinhole imaging and golden sine algorithm for improved numerical optimization performance[J]. Scientific Reports, 2024, 14(1): 1491.
- [19] 展月, 姜兆勤, 刘振元. 多结构型任务驱动的上门服务调度优化模型与自适应大规模邻域搜索算法[J]. 控制与决策, 2024, 39(3): 947-955.
Zhan Yue, Jiang Zhaoqin, Liu Zhenyuan. Optimization model and revised adaptive large neighbourhood search algorithm for multi-structured on-site service scheduling problem[J]. Control and Decision, 2024, 39(3): 947-955. (in Chinese)
- [20] 白小山, 余桢奇, 郑心泉, 等. 基于改进最小边际代价算法的多 USV 多 AUV 任务分配[J]. 控制与决策, 2025, 40(1): 119-127.
Bai Xiaoshan, She Anqi, Zheng Xinquan, et al. Task assignment for multiple USVs and AUVs based on improved minimum marginal cost algorithm[J]. Control and Decision, 2025, 40(1): 119-127. (in Chinese)

Multi-agent collaborative task pre-allocation based on improved sand cat swarm optimization algorithm

Zhou Shixi^{1,2}, Zhang Jiandong¹, Shi Guoqing¹, Yang Qiming¹,
Cao Yi¹, Cheng Shuyuan¹

(1.School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China;
2.AVIC Xi'an Flight Automatic Control Research Institute, Xi'an 710076, China)

Abstract: To address the challenge of heterogeneous platform collaborative task pre-allocation, this study proposes an enhanced sand cat swarm optimization (SCSO) algorithm incorporating nonlinear convergence factors and an external archive update strategy. This improved algorithm integrates large neighborhood search (LNS) with dedicated destruction and repair operators. Two simulation cases validate the algorithm's efficacy: Case 1 compares its performance against genetic algorithm (GA), lemur algorithm, and whale optimization algorithm (WOA) on vehicle routing problems with time windows (VRPTW). Case 2 demonstrates its application in multi-heterogeneous agent reconnaissance/attack task allocation scenarios.

Keywords: multi-objective allocation; SCSO algorithm; multi-agent collaboration

引用格式: 周世曦, 张建东, 史国庆, 等. 基于改进沙猫算法的多机协同任务预分配[J]. 西北工业大学学报, 2026, 44(1): 102-111.

Zhou Shixi, Zhang Jiandong, Shi Guoqing, et al. Multi-agent collaborative task pre-allocation based on improved sand cat swarm optimization algorithm[J]. Journal of Northwestern Polytechnical University, 2026, 44(1): 102-111. (in Chinese)