文章编号:1671-4598(2025)04-0147-08 DOI:10.16526/j.cnki.11-4762/tp.2025.04.020 中图分类号:TP4081;V219 文献标识码:A

多无人机协同路径规划计算资源分配方法研究

聂铭涛[1]，刘颖珂[1]，程海峰[2]，刘小雄[1]
(1. 西北工业大学 自动化学院，西安 710072；
2. 中国人民解放军 94521 部队，山东 淄博 255399)

摘要：针对多目标多机协同路径规划问题，在 MOEA/D 算法求解的基础上，采用深度强化学习方法对 MOEA/D 算法中计算资源分配方法进行了研究；对多机协同路径规划问题进行了研究，分析了相关约束以及优化目标，建立多机协同路径规划多目标优化模型；结合协同进化思想，对基于分解的多目标进化协同路径规划进行了研究；对基于强化学习的计算资源分配策略进行了研究，实现了深度强化学习在多目标优化计算资源分配问题上的应用；实现了多机协同路径规划仿真验证；经仿真测试，算法以更高性能完成多机协同路径规划任务，提高了多机协同路径规划中计算资源分配策略的能力。

关键词：多目标进化；深度强化学习；计算资源分配；协同路径规划；基于分解

Research on Resource Allocation Method for Multi-UAV Cooperative Path Planning

NIE Mingtao[1], LIU Yingke[1], CHENG Haifeng[2], LIU Xiaoxiong[1]
(1. School of Automation, Northwestern Polytechnical University, Xi'an 710072, China;
2. Unit 94521, PLA, Zibo 255399, China)

Abstract: Aiming at the problem of multi-objective and multi-UAV cooperative path planning, based on the solution of MOEA/D algorithm, a deep reinforcement learning method is adopted to study the computational resource allocation method in the MOEA/D algorithm; Study the multi-UAV cooperative path planning, analyze the relevant constraints and optimization objectives, and establish the multi-objective optimization model of multi-UAV cooperative path planning; Investigate the multi-objective evolutionary cooperative path planning based on decomposition by combined with the idea of co-evolution, study the computational resource allocation strategy based on reinforcement learning, realize the application of deep reinforcement learning in the multi-objective optimization of computational resource allocation, and achieve the simulation verification of the multi-UAV cooperative path planning; Though simulation test, the algorithm completes the multi-UAV cooperative path planning task with a higher performance and improves the performance of the computational resource allocation strategy.

Keywords: multi-objective evolutionary; deep reinforcement learning; computational resource allocation; cooperative path planning; decomposition-based

0 引言

随着科学技术进步以及空中装备性能的快速发展，无人机在现代战争中发挥着越来越重要的作用。然而单架无人机的视野范围、飞行范围及承载能力有限，难以完成复杂的任务，因此多无人机协同执行任务的作战形式逐渐成为无人机应用的主流。相比于单机路径规划问题，多机协同路径规划面对的情况更加复杂，主要体现在：1）约束条件更多，不仅包括无人机自身的飞行性能约束，还要考虑无人机间的时间协同关系以及空间协同约束等；2）问题解的维度更大，需要更有效的处理方法。综合看来，多机协同路径规划问题仍然是一个极

收稿日期:2024-12-25； **修回日期**:2025-02-21。
作者简介:聂铭涛(2000-),男,硕士。
通讯作者:刘小雄(1973-),男,博士,副教授。
引用格式:聂铭涛,刘颖珂,程海峰,等.多无人机协同路径规划计算资源分配方法研究[J].计算机测量与控制,2025,33(4):147-154,161.

具挑战的研究课题[1-4]。

多机协同路径规划本质上是一种多目标优化问题(MOPs，multi-objective optimization problems)。近些年来，多目标优化问题取得了大量理论进展，基于分解的多目标进化算法（MOEA/D，multi-objective evolutionary algorithm based on decomposition）在权重向量、分解方法、重组策略、替换策略、计算资源分配 5 个方向上分别取得了针对性的进展[5]，其中计算资源分配方向的研究使得 MOEA/D 以更高的计算效率和求解质量应用于现实世界中的复杂问题，为多机协同路径规划问题提供了一种更先进的求解思路。2009 年，文献［6］于 MOEA/D 框架提出了一种计算资源自动分配算法(MOEA/D-DRA，multi-objective evolutionary algorithm based on decomposition with dynamical resource allocation)。算法中每 50 次迭代，为每个子问题更新一个效用函数 ，并选择效用函数较大的一组子问题进化，从而实现计算资源分配。2016 年，文献［7］对 MOEA/D-DRA 算法进行拓展，引入了概率向量的概念，并比较了离线资源分配和在线资源分配两种概率向量更新策略，事实证明，在线资源分配更加实用。文献［8］在 2015 年提出在权重向量中额外引入随机权重向量实现收敛困难子问题的求解，提高算法在不规则前沿面上的求解性能。遗憾的是，无论是效用函数的设计、概率向量的计算以及随机权重向量的选择，都需要依赖大量的专家知识，且对问题缺少一定的泛用性，导致在解决不同问题时需要付出更多工作量。

强化学习算法由于其自我学习能力以及在复杂场景中的出色表现为计算资源分配策略的改进提供了一种思路。在除计算资源配策略以外，强化学习在进化算法中的已经有大量的应用，2006 年，文献［9］第一次用 Q-Learning 和 Sarsa 算法对进化算法进行了改进，实验证明了算法性能的提升，文献［10］使用近端策略优化(PPO，proximal policy optimization）算法对进化算法不同子问题配置不同的进化参数。然而目前强化学习在计算资源分配领域应用的研究暂时空白，仍需大量的探索研究。

因此，本文以多机协同路径规划问题为任务背景，提出了一种基于强化学习计算资源分配方法，研究了深度强化学习方法在基于分解的多目标优化算法计算资源分配研究方向的应用与改进，提供了一种解决计算资源分配问题并在实际问题中应用的新思路。

1 问题描述

本文针对多机协同突防任务，同时出动多架飞机，绕过敌方障碍并在同一时间到达目标的上空实施饱和打击，这个过程中需要综合考虑无人机性能、地形环境、地方威胁、时间协同关系、空间协同关系等要素，可以建模成一个多目标优化问题。一般多目标问题可以表示为如下形式：

$$\begin{aligned} &\text{minimize} \quad f(x)=[f_1(\boldsymbol{x}),f_2(\boldsymbol{x}),\cdots,f_m(\boldsymbol{x})] \\ &\text{subject to} \quad g_1(\boldsymbol{x})\leqslant 0,\cdots,g_I(\boldsymbol{x})\leqslant 0, \\ &h_1(\boldsymbol{x})=0,\cdots,h_J(\boldsymbol{x})=0, \\ &\boldsymbol{x}\in\Omega \end{aligned} \tag{1}$$

其中：m 是 MOP 的目标数量，$\boldsymbol{x}=[x_1,x_2,\cdots,x_d]$ 是 d 维决策变量，Ω 表示决策空间，$f_1,f_2,\cdots,f_m$ 表示 m 个相互冲突的目标函数，$g_1,g_2,\cdots,g_I$ 和 $h_1,h_2,\cdots,h_J$ 分别是由 I 个不等式和 J 个等式定义的约束条件。

在本文任务场景下，无人机需要规划较短的路径，绕过地方障碍安全到达目标点，并在飞行过程中不断调整速度或延长路径以保证和其他无人机同时到达目标，且需要保证机间有安全的距离，防止机间碰撞。因此，对问题建模时需要考虑路径长度、时间协同等因素以及环境中敌方障碍、空间协同以及无人机性能等约束。

本文将多机协同路径规划问题建模为包含 3 个目标的多目标优化问题，3 个目标分别是总路径长度目标、环境威胁目标和时间协同目标。总路径长度目标要求所有无人机的路径长度尽可能短，其表示形式设计如下：

$$f_1(\boldsymbol{x})=\sum_{k=1}^{N}(L^k/L_{\min}^k) \tag{2}$$

其中：N 为无人机数量，$L_{\min}^k$ 为第 k 架无人机的起点到终点直线距离，L^k 为第 k 架无人机规划的路径长度。

环境威胁目标要求无人机尽可能避开环境中敌方障碍。本文设计了警戒雷达和防空火炮两种环境威胁，环境威胁目标函数值为两者威胁值的总和，其表示形式如下：

$$f_2(\boldsymbol{x})=\sum_{k=1}^{N}(\sum_{i=1}^{W}{U_{ai}}^k+\sum_{j=1}^{U}{P_{dj}}^k) \tag{3}$$

其中：W 为环境中防空火炮数量，U 为环境中警戒雷达数量，U_{ai}^k 为第 k 架无人机的路径受到的防空火炮 i 的威胁大小，其计算方法参考文献［11］，P_{dj}^k 为第 k 架无人机的路径上受到警戒雷达 j 的威胁值大小，其计算方法参考文献［12］。

时间协同目标是基于多无人机协同突防、饱和打击的任务背景，要求所有无人机尽可能同时到达目标点。目标函数用飞机间路径长度差值的最大值表示：

$$f_3(\boldsymbol{x})=max\mid L^k-L^p\mid,k\in[1,N],p\in[1,N] \tag{4}$$

其中：L^k 为第 k 架无人机规划的路径长度，L^p 为第 p 架无人机规划的路径长度。

在实际规划中，3 个目标往往是相互冲突的，如何取舍需要依赖具体情况以及决策者的经验分析，多目标优化能够给出一组非支配解集，方便决策者从中挑选合

适的解以应对复杂多变的战场态势。

本文主要考虑了无人机性能约束和空间协同约束两种约束条件。无人机性能约束包括如下几种：

1）最小转弯半径约束：

$$R_i > R_{min},(i = 1,2,\ldots n) \tag{5}$$

其中：R_i 为所规划路径中第 i 段到第 $i+1$ 段转弯的转弯半径，$R_{\min}$ 为无人机最小转弯半径。

2）最小路径段长度约束：限制了无人机在改变姿态前必须经过的最小距离，在实际规划中可以通过控制规划单位步长满足约束。最小路径段长度约束表示为：

$$dS > l_{\min} \tag{6}$$

其中：dS 为规划单位步长，$l_{\min}$ 为最小路径段长度。

3）最大路径坡度角约束：最大路径坡度角约束体现无人机的爬升以及俯冲性能。假设路径段在水平面上投影长度为 a，垂直方向落差为 b，则最大航迹坡度角约束表示为：

$$\frac{b}{a} \leqslant \tan\theta_{\max} \tag{7}$$

其中：$\theta_{\max}$ 为最大航迹坡度角。

4）最小离地间隔约束：最小离地间隔保证无人机与地面有足够安全的距离：

$$z_i \geqslant H_{\min},(i = 1,2,\ldots,n) \tag{8}$$

其中：z_i 为第 i 个路径点的高度，$H_{\min}$ 为最小离地间隔。

5）空间协同约束：通过限制两条路径中路径点之间的距离来避免两机发生碰撞：

$$|\ p_a - p_b\ | > d_{\min},(a = 1,2,\cdots,n_1),(b = 1,2,\cdots,n_2) \tag{9}$$

其中：p_a 为飞机 1 路径点坐标，p_b 为飞机 2 路径点坐标，$d_{\min}$ 为最小路径点间距离。

另外，本文设置禁飞区约束，规定无人机规划路径不可经过禁飞区。

2　基于强化学习计算资源分配的协同路径规划算法设计

在本节中，我们在 MOEA/D[13] 框架下提出了一个基于强化学习的计算资源分配策略，并用来求解本文提出的协同路径规划问题。首先，采用分解策略将建模的 MOP 分解为若干子问题，结合协同进化思想构建迭代求解过程。然后，将迭代求解过程建模为马尔可夫决策过程，用深度神经网络充当计算资源分配模块，并运用双延迟深度确定性策略梯度（TD3，twin delayed deep deterministic policy gradient）算法对网络进行训练，得到一个基于强化学习的计算资源分配策略。

2.1　基于分解的多目标进化协同路径规划设计

基于分解的多目标进化算法是一种有效的多目标问题处理框架，其主要思想是将多目标问题分解为若干个单目标子问题，在迭代过程中并行地求解每个子问题，最后通过不同的最优解组合来近似求解多目标问题的帕累托解集。不同分解方法的应用和先进的单目标问题处理方法使得 MOEA/D 算法具有较高的泛用性，已经在很多复杂的现实问题上取得了较好的成果[14-15]。因此本文使用 MOEA/D 作为本文求解框架，实现了无人机协同路径规划问题的求解。

2.1.1　分解策略

分解是一种有效的 MOP 处理方法，其思想促生了大量相关领域的研究成果，例如 MOEA/D、MOEA/DD[16]、NSGA-III[17] 等。本文使用分解的思想，将提出的多目标协同路径规划分解为一组单目标子问题，并通过改进的遗传算法进行求解。具体来说，本文使用切比雪夫分解方式处理多目标协同路径规划问题。首先，生成一组均匀的权重向量 $\boldsymbol{\lambda}^1,\cdots,\boldsymbol{\lambda}^{Np}$，其中 $\boldsymbol{\lambda}^i = (\lambda_1^i,\cdots,\lambda_m^i)^T$，$Np$ 为子问题数量，本文中设置为 200，$m = 3$ 为优化目标数量；接着随机初始化一个参考点 $z^* = (z_1^*,\cdots,z_m^*)^T$，由此得到分解后的第 i 个子问题可以表示为：

$$\begin{aligned}&\text{minimize}\quad g^{te}(x\mid\boldsymbol{\lambda}^i,z^*) = \\ &\max_{1\leqslant j\leqslant m}\{\boldsymbol{\lambda}_j^i\mid f_j(x) - z_j^*\mid\}\end{aligned} \tag{10}$$

2.1.2　协同进化思想

对于单独某个子问题来说，同时处理多架无人机的路径规划问题仍然需要大量的组合计算，且其中对进化有贡献的优势组合只占其中很小部分。为了提高计算效率，本文使用协同进化思想结合遗传算法对子问题进行求解。基于协同进化的思想，在子问题 i 中，不同无人机的路径构成不同的子种群，每个子种群单独完成遗传过程，由于遗传算法并非本文研究重点，因此使用之前工作中提出的交叉和遗传算子构建遗传过程[18]。在对路径进行评价时，需要结合其他无人机信息。具体表现为，选择其他无人机的路径与该路径组合，构成完整组合解，并基于目标函数评价路径的适应性。一般情况下，可以使用最优选择方法选择其他子种群最优个体作为代表，参与组合解的构建。在 MOEA/D 框架下，子问题与其最优解一一对应，因此本文选择当前子问题对应的其他无人机路径作为代表，与当前无人机路径一次组合，基于当前子问题目标函数对路径进行评价。无人机与其他无人机的信息交互如图 1 所示。

2.2　基于强化学习的计算资源分配策略

MOEA/D 中分解子问题时使用了固定的权重向量，子问题也随之固定，且演化过程中每个子问题分配了相同的计算资源。然而在复杂的现实问题中，不同子问题优化难易程度不同，给所有子问题分配相同的计算资源，难免造成计算资源的浪费[19]。

计算资源分配是 MOEA/D 改进算法的一个主要研

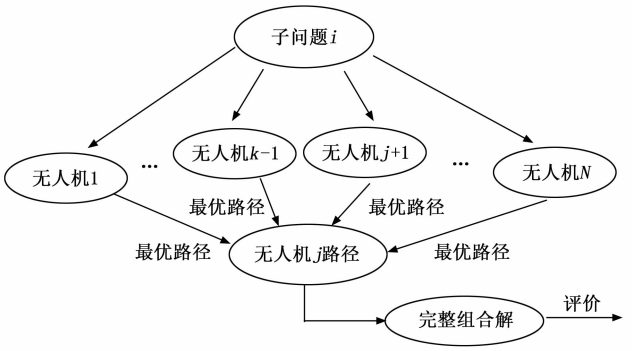


图 1 第 k 架无人机与其他无人机的信息交互示意图

究方向，其思想是为不同子问题分配不同的计算资源，实现计算资源的合理分配，提高算法收敛速度和求解效率。然而现有的大部分计算资源分配方法依赖专家知识，对问题的泛用性不高。因此，本文利用深度强化学习算法实现计算资源分配功能，强化学习有着较强的自我学习能力，能够减少专家知识的使用，同时提高算法泛用性。

定义概率向量 $\boldsymbol{P}=(p_1,\cdots,p_{N_p})$，其中每一个元素对应不同的子问题，将迭代过程中概率向量的自适应调整建模为马尔可夫过程：某次迭代中，策略生成一个概率向量作为动作 a，经过 MOEA/D 结合概率向量进行一次迭代获得新解，并对解集进行更新；为了兼顾解集的收敛性以及多样性，通过解集质量的变化程度作为环境状态量 s，衡量环境的变化，并给出奖励 r。

TD3 是一种针对连续动作的确定性策略强化学习算法，在 DDPG[20] 的基础上解决了过拟合以及稳定性不佳的问题，受到了广泛的应用[21]。本文使用 TD3 算法[22]构建计算资源分配智能体，包括一个策略网络 $\mu(s\mid\theta)$ 和两个价值网络 $Q_i(s,a\mid\omega),i=1,2$，其中 ω 和 θ 是网络参数。策略网络训练的目的是使价值网络评分最大化，其目标函数可以表示为：

$$J(\theta)=\max E_s[Q(s,\mu(s\mid\theta)\mid\omega)] \tag{11}$$

利用梯度上升的方法更新策略网络参数，目标函数对于策略网络参数 θ 的梯度为：

$$\nabla_\theta J(\theta)=E_s[\nabla_\theta Q(s,\mu(s\mid\theta)\mid\omega)] \tag{12}$$

该梯度可以蒙特卡洛近似为：

$$g_\theta=\nabla_\theta q(s_t,\mu(s_t\mid\theta)\mid\omega)=\nabla_\theta\mu(s_t\mid\theta)\cdot\nabla_{\mu(s_t\mid\theta)}Q(s_t,\mu(s_t\mid\theta)\mid\omega) \tag{13}$$

使用梯度上升更新网络参数：

$$\theta\leftarrow\theta+\beta\cdot g_\theta \tag{14}$$

价值网络通过基于时间差分的方法训练，为了解决自举和最大化带来的高估问题，算法引入了两套结构相同的目标价值网络 $Q_1(s,a\mid\omega_1')$ 和 $Q_2(s,a\mid\omega_2')$，通过两者的较小评分估计下一步的价值。另外，为了消除目标估计中的误差影响，通过正则化策略平滑目标策略，具体做法是在目标策略网络中引入随机噪声 ε。由此 TD 目标可以表示为：

$$y_t=r_t+\gamma\cdot\min_{i=1,2}Q_i(s_{t+1},\overline{a}_{t+1}+\varepsilon\mid\omega_i'),\quad \varepsilon\sim clip(N(0,\sigma),-c,c) \tag{15}$$

接着通过最小化损失函数 L_i 对两个价值网络参数进行更新：

$$L_i=\frac{1}{N}\sum_{t=1}^{N}(y_t-Q_i(s,a\mid\omega_i))^2,i=1,2 \tag{16}$$

算法中价值网络的更新频率高于策略网络，以求用稳定的价值网络对策略网络进行更新，同时对所有目标网络进行软更新：

$$\omega_i'\leftarrow\tau\omega_i+(1-\tau)\omega_i'$$
$$\theta\leftarrow\tau\theta+(1-\tau)\theta \tag{17}$$

完成训练后，在后续执行中不再需要价值网络，只需通过策略网络做出决策，策略网络能够根据当前迭代的解输出概率向量用于下一代计算资源在不同子问题间的分配。

2.3 算法流程框架设计

基于强化学习计算资源分配方法的协同路径规划框架如图 2 所示。算法在基于分解的多目标进化算法的协同路径规划算法基础上，维持一个候选子问题集合 CS。每一次迭代开始时，根据概率向量 $\boldsymbol{P}$ 从所有子问题中选择一部分子问题放入 CS 中，这一代中只有 CS 中的子问题会参与遗传过程产生新的子代，结合协同进化思想的遗传算法依次对 CS 中的子问题进行求解，这一代结束后 CS 被重置为空，并进行下一次迭代，直到迭代次数到达最大代数或满足停止条件，算法输出外部种群 EP 作为最后的非支配解集。

环境状态空间描述了智能体能够从环境中获取的信息。我们希望智能体的策略函数能够根据进化进程自主学习，因此将迭代代数 gen 作为其中一个状态量。另外，为了兼顾解集的收敛性以及多样性性能，本文通过超体积指标（HV，hypervolume）指标评估算法在收敛性和多样性的优劣。HV 指标代表算法所获得的解集与参照点在目标空间中围成区域的超体积，同时独立地评价解集的收敛性以及多样性[23]，能够间接反映出子问题概率向量，即策略决策的优劣，将其作为其中一个状态量。

另外，为了具体描述单个子问题分配计算资源后对解集做出的贡献，本文分别设计了每个子问题的收敛性和多样性评估方法。其中收敛性评估为一次迭代中子问题及其邻域适应度的改进大小的平均值，表示为：

$$u_i^{gen}=\frac{1}{T}\sum_{j\in N_i(i)}max[0,g_j^{te}(x_j^{new,gen})-g_j^{te}(x_i^{gen})] \tag{18}$$

其中：u_i^{gen} 为子问题 i 及其邻域子问题解的适应度在第 gen 代的改进量，x_i^{gen} 为子问题 i 在第 gen 代的初始

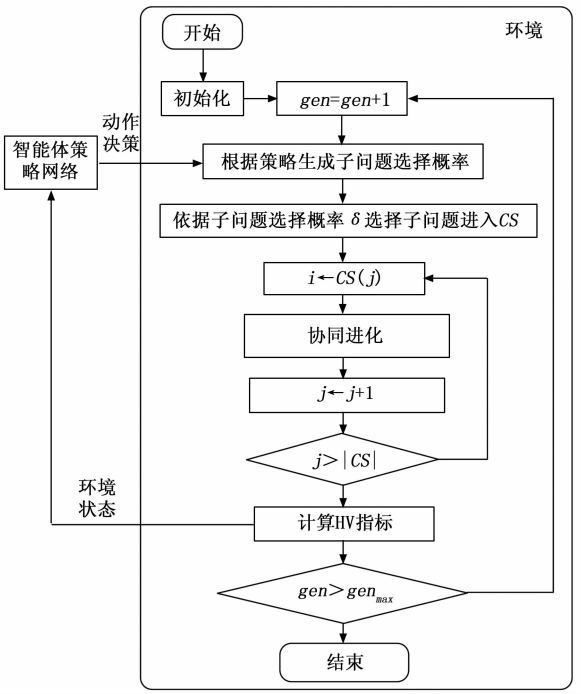


图 2　基于强化学习计算资源分配方法的协同路径规划框架图

解，$x_j^{new,gen}$ 为子问题 j 在第 gen 代的新解，子问题 j 是子问题 i 的邻域子问题，即 $j \in N_b(i)$，T 为邻域大小，本文设置为 5。

收敛性评估状态量为所有子问题评估的集合，即：

$$s_u^{gen} = \{u_i^{gen}\}, i = 1, \cdots, N_p \tag{19}$$

本文根据子问题解的稀疏性间接表示解集多样性，每个子问题的稀疏性通过目标空间中子问题解到邻域子问题解的距离的平均值表示：

$$s_i^{gen} = \frac{1}{T} \sum_{j \in N_b(i)} \left| \boldsymbol{F}(x_i^{gen}) - \boldsymbol{F}(x_j^{gen}) \right| \tag{20}$$

其中：$\boldsymbol{F}(\cdot)$ 表示解的目标值向量，则多样性评估状态量为所有子问题评估的集合：

$$s_s^{gen} = \{s_i^{gen}\}, i = 1, \cdots, N_p \tag{21}$$

综上所述，状态空间设计为：

$$s_t = [gen, HV_{gen}, s_u^{gen}, s_s^{gen}] \tag{22}$$

2.4　奖励函数设计

进行计算资源分配时，不仅要考虑解集的收敛性和多样性性能，还要考虑迭代过程中解集性能的变化，因此智能体的奖励来源于过程奖励 R_{pro} 和回合结束奖励 R_{end} 两部分。

迭代过程中，通过本代次中所有子问题改进值比例的平均值表示整体问题的改进，因此过程收敛性奖励为：

$$r_u = \begin{cases} \dfrac{1}{N_p} \sum_{i=1}^{N_p} \dfrac{\Delta g_i}{max \Delta g_i} & max \Delta g_i \neq 0 \\ 0 & max \Delta g_i = 0 \end{cases} \tag{23}$$

其中：$\Delta g_i = g_i^{te}(x_i) - g_i^{te}(x_i^{new})$ 表示子问题新解适应度的改进值。

计算子问题选择概率时，希望选择的子问题更新后能使得解集在目标空间中的分布更加均匀，可以通过每个子问题到邻域子问题距离的标准差衡量子问题解的均匀程度：

$$Spa_i = std[|F(x_i) - F(x_j)|], j \in N_b(i), j \neq i \tag{24}$$

其中：Spa_i 体现了子问题 i 的均匀性，其值越大，说明不同子问题解之间均匀性越差，因此综合所有子问题均匀性程度给予相应惩罚作为过程均匀性奖励：

$$r_s = -\sum_{i=1}^{N_p} std[|F(x_i^k) - F(x_j^k)|], j \in N_b(i), j \neq i \tag{25}$$

迭代过程奖励 R_{pro} 为收敛性奖励与均匀性奖励之和：

$$R_{\text{pro}} = \begin{cases} r_u + r_s & gen < gen_{\max} \\ 0 & gen = gen_{\max} \end{cases} \tag{26}$$

每个回合结束时，说明迭代到达停止条件，此时通过最终解集的 HV 指标大小给予回合结束奖励：

$$R_{\text{end}} = \begin{cases} HV(EP) & gen = gen_{\max} \\ 0 & gen < gen_{\max} \end{cases} \tag{27}$$

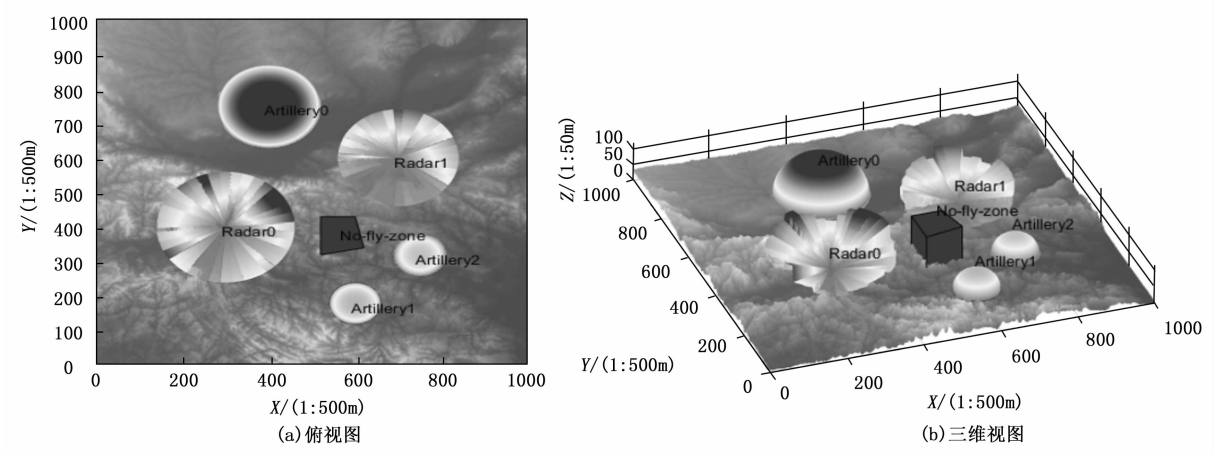
总奖励为过程奖励与回合结束奖励之和，即 $R = R_{\text{pro}} + R_{\text{end}}$。

3　仿真结果分析

3.1　实验环境及参数设置

为验证所设计的基于强化学习计算资源分配方法在协同路径规划问题中的有效性，本节设计的区域环境如图 3 所示，其中环境威胁中心位置坐标为：雷达 0、雷达 1 的坐标分别为（300，400，20）、（700，600，25），有效范围分别为 80 km、70 km；防空火炮 0、防空火炮 1、防空火炮 2 的位置坐标分别为（400，750，20）、（600，180，25）、（750，320，20），有效范围分别为 60、30、30 km，禁飞区位置为（520，320）、（620，340）、（600，430）和（520，430）4 个点围成的四边形所表示的棱柱区域。该场景设计包含多种常见障碍，且障碍范围设计合理，符合真实协同突防场景，因此具有代表性。

TD3 智能体中，价值网络和策略网络都为隐藏层数量为 3 的深度神经网络，隐藏层神经元数量分别设置为（64，32，1）和（128，64，200），隐藏层激活函数为 Relu 函数，策略网络输出层激活函数为 tanh 函数。训练参数设置如表 1 所示。



(a)俯视图 (b)三维视图

图 3 环境地图

表 1 训练参数设置

训练参数	参数值	训练参数	参数值
策略网络学习率	0.000 1	价值及网络学习率	0.000 5
策略网络更新延迟	2	目标网络更新延迟	2
批大小	64	折扣率	0.96
经验池规模	10 000	采样权重因子初值	0.6
目标策略网络平滑噪声方差	0.2	软件更新权重	0.005

3.2 路径规划仿真结果分析

本文设计四机协同路径规划场景对算法进行测试，无人机起点和目标点位置设置为：无人机 1 起点（200，100，50）、无人机 2 起点（800，60，50）、无人机 3 起点（100，400，60）、无人机 4 起点（895，300，60）、目标点（600，920，50）。基于强化学习计算资源分配方法的协同路径规划求解得到的非支配解集如图 4 所示。

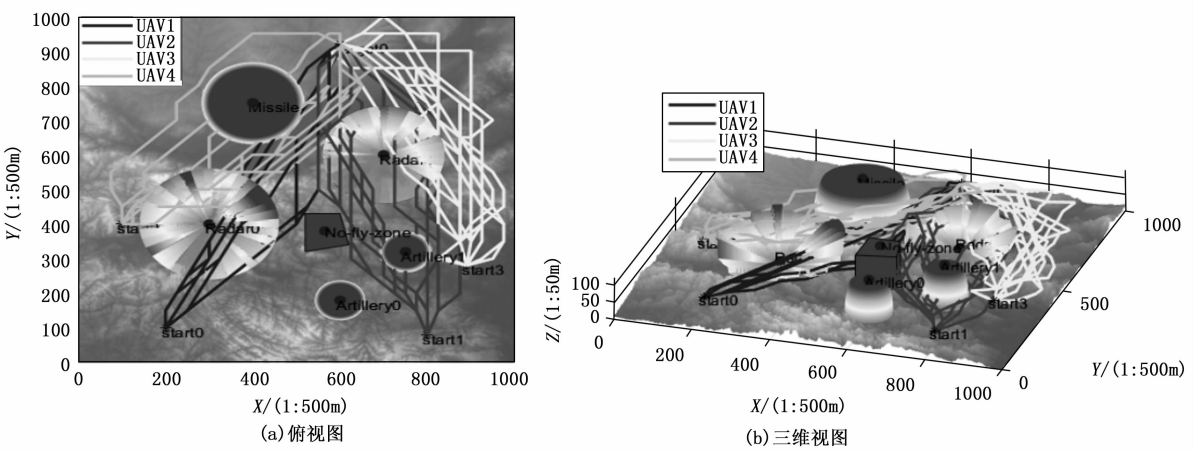
仿真结果 3 个目标的最优路径如图 5 所示。

由图 5 可知，当优化目标侧重路径长度目标时，四机基本无视雷达威胁以及火炮威胁，以最短的距离飞行至目标点；威胁代价最小的路径绕过火炮范围，并且能利用雷达盲区对雷达威胁进行规避，但代价是牺牲了路径长度；更倾向于时间协同目标时，可见无人机 2 路径有部分绕行，以满足所有无人机尽量同时到达目标点。

为对本文提出的计算资源分配策略进行分析，图 6 记录了进化不同阶段不同子问题的计算资源分配情况。实验中总迭代次数为 50，分别记录了前 25 代和后 25 代的平均计算资源分配情况。

从图 6 中看出，在进化前期，训练的策略找到了个别收敛较快的子问题，并给其分配了较多的计算资源，促进解集整体收敛，而此时其他子问题计算资源分配较少；在进化后期，为了保证解集多样性，策略对大多数子问题都分配了一定的计算资源。

为了突出强化学习在计算资源分配中应用的优势，下面将本文算法与不使用计算资源分配的协同路径规划和使用 MOEA/D-DRA 进行计算资源分配的算法进行了对比，分别进行了 50 次独立重复实验，并对结果 HV 指标以及求解时间进行比较，实验结果如表 2 所示。



(a)俯视图 (b)三维视图

图 4 四机协同路径规划

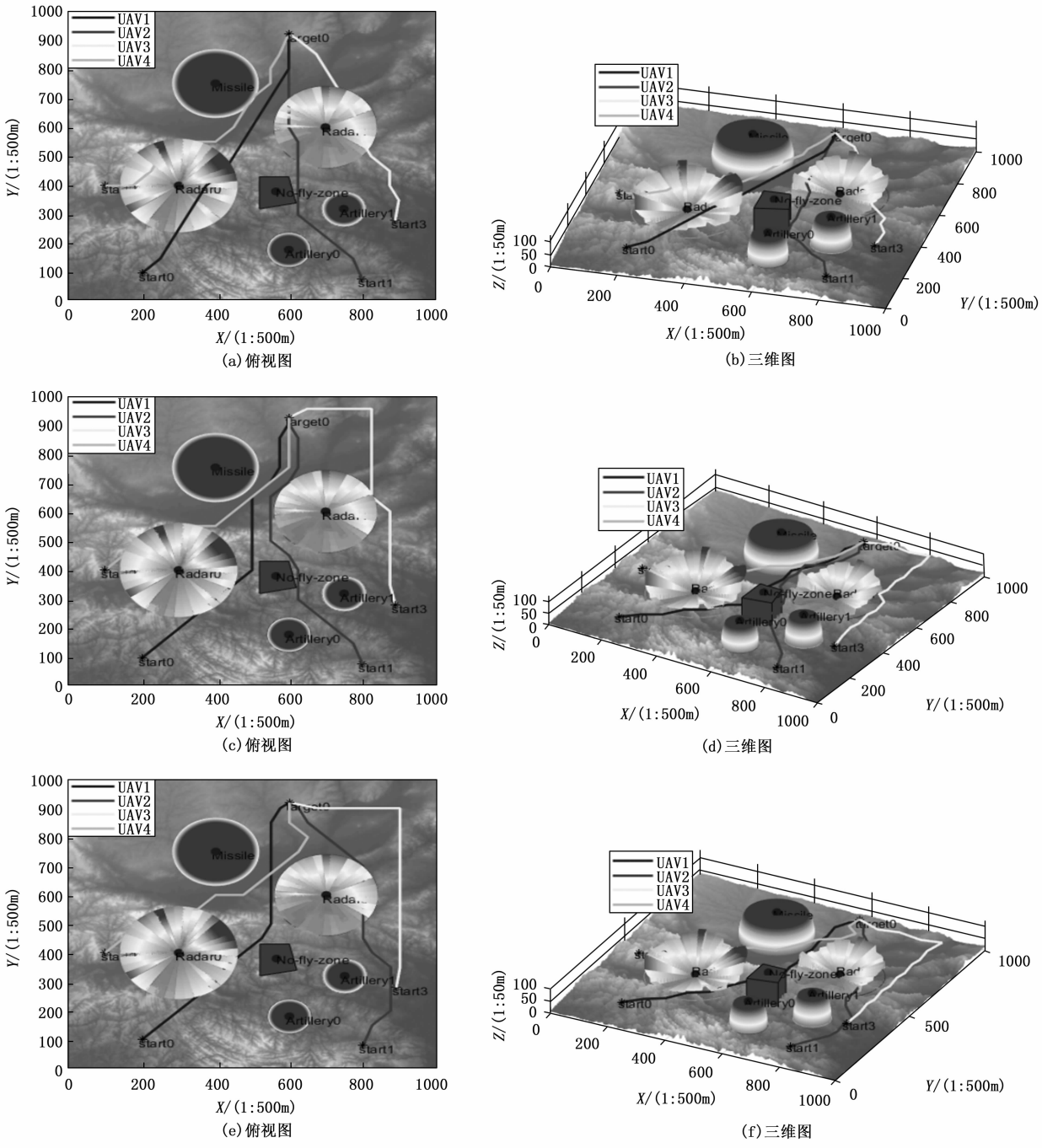


图 5　3 个目标的最优路径

表 2　50 次测试 HV 统计数据

方法	均值	标准差	最小值	最大值	运行时间/min
不使用计算资源分配	0.547 6	0.105 4	0.273 9	0.726 8	14.112 0
使用 MOEA/D-DRA 的计算资源分配	0.534 2	0.073 7	0.405 0	0.834 7	9.220 0
基于深度强化学习的计算资源分配	0.552 8	0.110 4	0.357 9	0.925 0	3.788 4

从表 2 可以看出，是否使用计算资源分配对 HV 指标影响不大，这是由于计算资源策略对最后非支配解集的分布影响较小；本文提出的方法在指标上的表现略优于 MOREA/D-DRA 方法，说明强化学习的方法能够在人工设计概率向量以外探索更多可能性，能在某些情况下有更好的表现，但也在一定程度上影响指标的标准差大小。另外，基于强化学习的计算资源在训练后运行时间相对很短，大幅度提高了算法计算效率。

4　结束语

基于分解的多目标进化算法广泛应用的背景下，本

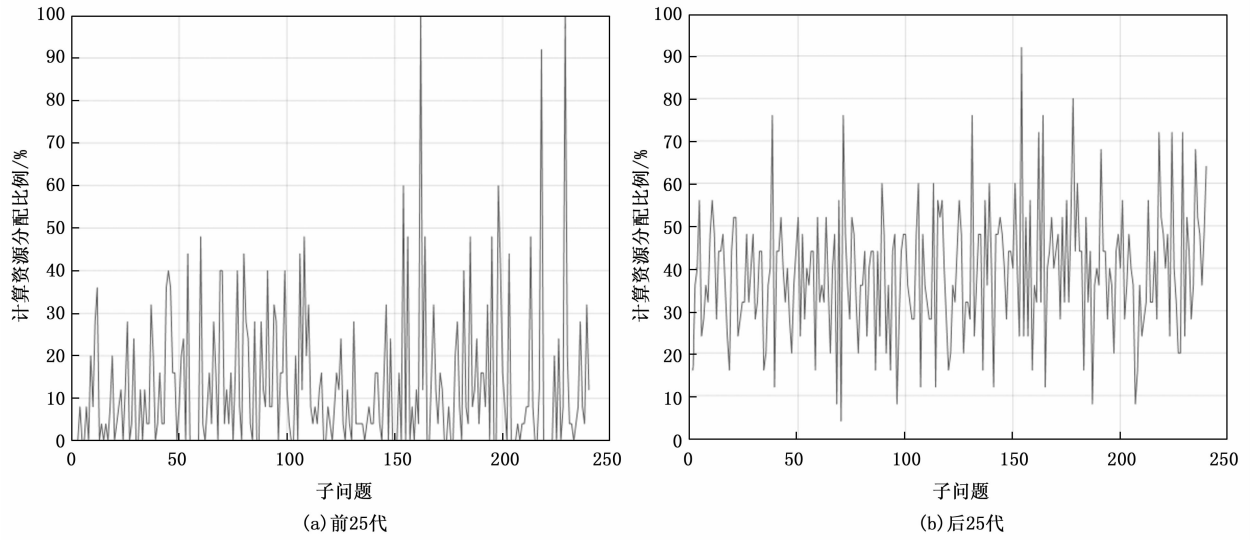


图 6 计算资源分配情况

文针对现有计算资源分配策略进行了改进，提出了一种基于深度强化学习的计算资源分配策略，并用于求解现实中的协同路径规划问题，给出了问题的建模、状态空间设计以及奖励函数设计。实验证明，本文提出的基于深度强化学习的计算资源分配策略能够在现实问题中提高计算资源分配策略性能，对强化学习方法在相关领域的应用提供了一种思路。未来进一步研究可着重提高算法泛用性，提高其在不同现实问题上的应用效果。

参考文献：

[1] HASSANALIAN M, ABDELKEFI A. Classifications, applications, and design challenges of drones: a review [J]. Progress in Aerospace Sciences, 2017, 91: 99 - 131.

[2] 高 谦，张玉良，曹 艳. 多无人机协同作战中的任务分配与路径规划算法研究 [J]. 无线互联科技，2024，21 (18)：23 - 26.

[3] 屈高敏，夏宗德，李继广，等. 基于协同进化算法多无人机协同路径规划研究 [J]. 西安航空学院学报，2024，42 (3)：1 - 9.

[4] 黄 辉，年福耿. 多无人机协同侦察路径规划研究 [J]. 舰船电子工程，2021，41 (9)：58 - 61.

[5] 高卫峰，刘玲玲，王振坤，等. 基于分解的演化多目标优化算法综述 [J]. 软件学报，2023，34 (10)：4743 - 4771.

[6] ZHANG Q F, LIU W D, LI H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances [J]. 2009 IEEE Congress on Evolutionary Computation, 2009: 203 - 208.

[7] ZHOU A M, ZHANG Q F. Are all the subproblems equally important? resource allocation in decomposition-based multi-objective evolutionary algorithms [J]. IEEE Trans. on Evolutionary Computation, 2016, 20 (1): 52 - 64.

[8] LI H, DING M, DENG J D, et al. On the use of random weights in MOEA/D [J]. 2015 IEEE EEE Congress on Evolutionary Computation, 2015: 978 - 985.

[9] EIBEN, HORVATH A E, KOWALCZYK M. Reinforcement learning for online control of evolutionary algorithms [C] //4th International Workshop on Engineering Self-Organising Applications, 2007: 151 - 160.

[10] GUO H, YINING M. Deep reinforcement learning for dynamic algorithm selection: a proof-of-principle study on differential evolution [J]. IEEE Transactions on Systems Man Cybernetics-Systems, 2024, 54 (7): 4247 - 4259.

[11] 程泽新，李东生，高 杨. 无人飞行器战场威胁空间量化建模 [J]. 电子信息对抗技术，2018，33 (5)：67 - 72.

[12] 王 旭，宋笔锋，郭晓辉. 飞行器被雷达发现概率的计算方法研究 [J]. 系统工程理论与实践，2006 (6)：130 - 134.

[13] ZHANG Q, LI H. MOEA/D: a multi-objective evolutionary algorithm based on decomposition [J]. IEEE Transactions on Evolutionary Computation, 2007, 11 (6): 712 - 731.

[14] ZHANG M, JIAO L, MA W, et al. Multi-objective evolutionary fuzzy clustering for image segmentation with MOEA/D [J]. Appl. Soft Comput. ,2016,48:621 - 637.

[15] JI J, GUO Y, GONG D, et al. WMOEA/D-based participant selection method for crowdsensing with social awareness [J]. Appl. Soft Comput. , 2020, 87: 105981.

[16] LI K, DEB K, ZHANG Q, et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition [J]. IEEE Trans. Evol. Comput. , 2015, 19 (5): 694 - 716.

[17] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints [J]. IEEE Trans. Evol. Comput. , 2014, 18 (4): 577 - 601.

(下转第 161 页)