引用格式：刘柳, 张聪, 吴帆, 等. 基于种群扰动遗传算法的多任务航天测控站资源池调度方法 [J]. 电子科技大学学报, 2026, 55(3): 361-373.
LIU L, ZHANG C, WU F, et al. A scheduling method for multi-task aerospace TT&C ground station resource pools based on population perturbation genetic algorithm[J]. Journal of University of Electronic Science and Technology of China, 2026, 55(3): 361-373.

基于种群扰动遗传算法的多任务航天测控站资源池调度方法

刘 柳[1,2]，张 聪[3]，吴 帆[3]，刘 田[2*]，李晶晶[4]

（1. 西安电子科技大学通信工程学院，西安 710071；2. 中国电子科技集团公司第十研究所，成都 610036；
3. 北京邮电大学电子工程学院，北京 100876；4. 电子科技大学计算机科学与工程学院，成都 611731）

摘要：现代航天测控地面站面临着测控业务种类的多样化和资源需求急剧增加的双重挑战。传统地面站基带处理资源池因其编排相对固化、缺乏灵活性，导致资源利用率和系统性能受到限制。针对上述问题，提出了一种基于种群扰动遗传算法的动态资源调度策略。该算法通过引入全局组合编码方式、精英个体保留策略、群体扰动机制、非传统的多次随机部分交叉以及单点变异与两点变异并行的变异操作，增加了调度灵活性和响应速度。通过系列仿真实验验证，结果表明，相较于传统算法，提出的基于种群扰动遗传算法能快速实现基带处理资源的动态分配，在任务处理效率上取得了显著改进，尤其在处理高负载的任务时表现优异，有效提升了地面站资源的利用率和操作效率。

关键词：航天测控地面站；任务调度；动态调度；遗传算法
中图分类号：TP301.6 **文献标志码：**A DOI: 10.12178/1001-0548.2025174

A scheduling method for multi-task aerospace TT&C ground station resource pools based on population perturbation genetic algorithm

LIU Liu[1,2], ZHANG Cong[3], WU Fan[3], LIU Tian[2*], and LI Jingjing[4]

(1. School of Telecommunications Engineering, Xidian University, Xi'an 710071, China; 2. The 10th Research Institute of China Electronics Technology Group Corporation, Chengdu 610036, China; 3. School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; 4. School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: The ground stations of modern aerospace telemetry, tracking, and command (TT&C) systems face the dual challenges of diversification of measurement and control task types and a sharp increase in resource demands. Traditional ground station baseband processing resource pools are limited in resource utilization and system performance due to their relatively solidified scheduling and lack of flexibility. To address the above problems, this paper proposes a dynamic resource scheduling strategy based on a population perturbation genetic algorithm. The algorithm introduces a global combinatorial encoding method, elitist individual preservation strategy, population perturbation mechanism, non-traditional multi-stage random partial crossover, and parallel single-point and two-point mutation operations, thereby enhancing scheduling flexibility and responsiveness. Through a series of simulation experiments, the results demonstrate that compared to traditional algorithms, the proposed population perturbation genetic algorithm can quickly achieve dynamic baseband processing resource allocation, resulting in significant improvements in task processing efficiency， it performs particularly well in handling high-load tasks, effectively enhancing the utilization and operational efficiency of ground station resources.

Key words: TT&C ground station; task scheduling; dynamic scheduling; genetic algorithm

航天测控地面站担负着各类飞行器的测控管理任务，《中国航天科技活动蓝皮书（2024 年）》显示[1]，中国在 2024 年实施 68 次发射任务，研制发射 290 余个航天器，发射次数及航天器数量刷新

收稿日期：2025−08−27
基金项目：国家自然科学基金（62293494）；成都市科技计划资助（2024-JB00-00014-GX）
作者简介：刘柳，博士生，高级工程师，主要从事测控通信方面的研究。
*通信作者 E-mail: liutian139@139.com

中国最高纪录。同时，全球航天发射进入新的高峰期，发射次数和质量均达到历史新高。随着卫星规模的不断扩大和测控任务种类的多样化，测控资源的需求急剧增加，给测控系统带来了极大的压力和挑战。当前测控地面站系统普遍采用“与特定卫星工程型号绑定”专建专用建设模式及“套件化”部署方案，引发设备功能解耦不足、资源互操作受限、冗余配置困难等问题。

为克服硬件堆叠导致的资源低效利用，适应多任务动态重组需求，国际上先后发展了三代跟踪与数据中继卫星系统[2]（tracking and data relay satellite system, TDRSS），基于测控站资源池化架构，依托通用处理平台与虚拟化运行环境，通过集中式资源编排实现设备整合与自适应配置。但现有资源池化技术存在编排固化、调度僵化缺陷，导致系统面临资源碎片化积聚与弹性衰减困境，因此资源池动态调度技术成为研究关键，动态资源分配既能提高系统吞吐量，也能满足大规模任务的计算资源需求。

目前，资源池动态调度方法一般可分为数学优化方法和智能优化算法等。数学优化方法常采用拉格朗日方法[3]、整数线性规划模型[4-5]和改进正弦余弦算法[6]。智能优化算法基于物理或生物学现象和行为建立启发元信息引导搜索进程，具有很强的通用性，主要以启发式算法为主，如蚁群算法、粒子群优化算法、遗传算法等。文献 [7] 提出基于蚁群优化的方法，通过分阶段更新信息素轨迹来避免局部最优，提高全局搜索能力。文献 [8] 对文献 [7] 的方法进行改进，生成新的信息素模型。文献 [9] 针对实时调度问题，提出了一种基于蚁群算法的分层调度算法，无需进行重新调度过程，避免了复杂的离线精细调度过程，有效减少了计算时间。文献 [10] 提出了一种改进的具有实数和二元决策变量的粒子群优化算法。文献 [11] 应用了粒子群优化算法和多目标优化决策方法的组合，分别提高信息聚合的准确性及开发选择结果的鲁棒性。文献 [12] 研究了一种对地观测卫星调度的综合方法，并提出了一种两阶段遗传退火方法来解决调度问题。文献 [13] 将综合卫星成像与数据传输调度问题表述为一个混合整数规划模型，并提出了一种改进的遗传算法。文献 [14] 针对测控数传多任务组合优化问题（如资源使用时间、通信路径选择），采用标准粒子群算法，将任务解空间映射为粒子位置参数（如带宽、时间片），定义适应值函数优化资源分配，通过全局搜索实现成本最小化或收益最大化，以此来完成卫星任务地面站资源调度。文献 [15] 提出基于种群摄动和消除策略的遗传算法来解决测控资源调度问题。文献 [16] 采用两阶段遗传算法，将卫星时间约束和资源约束转化为数学模型，第一阶段选择任务执行圈数，第二阶段选择任务弧段，逐步缩小解空间，降低计算复杂度，该方法适用于面向卫星常规测控任务的地面站资源调度。文献 [17] 提出了一种基于多自适应策略的高阶量子遗传算法来解决敏捷遥感卫星调度问题。文献 [18] 提出了一种改进型遗传算法，通过增加约束条件（如资源优先级、任务窗口时间），提高卫星测控和数据接收任务的调度效率，相比于传统算法，资源利用率提升 15%～20%。文献 [19] 采用 K 均值聚类方法，提出基于聚类的遗传算法解决卫星测距调度问题。文献 [20] 设计了基于任务序列的遗传算法染色体编码方法和基于同类任务的遗传算法交叉、变异算子，其利用遗传算法的全局搜索优化解特点，对集群的协同任务分配进行优化。文献 [21] 提出基于近端策略优化的公平集成调度算法。文献 [22] 提出基于周期触发的动态调度框架和算法，解决预置卫星任务的地面资源调度。文献 [23] 针对当前卫星地面站资源调度策略效率低、动态适应能力不足的问题，基于改进粒子群算法结合动态调整机制，综合历史数据动态优化调度方案，提升灵活性和系统纠偏能力。

现有文献在资源调度策略领域的研究多聚焦于解决天线与波束资源的可见性、可用性问题，但针对大规模处理资源的调度方法及性能分析研究较为匮乏。尤其关键的是，在前述地面设施资源池化、功能虚拟云化的新架构场景下，需将多卫星并发业务的运行完成时间作为核心考量的系统参数，而相关研究工作仍有待深入推进。为此，本文设计了一种基于改进遗传算法的动态资源调度算法对大规模资源调度模型进行求解，该方法能实现地面站资源的灵活分配，显著降低调度延迟，提升资源的利用效率。

1 地面站资源调度优化模型

1.1 问题描述

航天测控地面站采用集中式管理的方式调度异构资源池，当测控卫星数据传输到地面站后，地面站会动态调度异构资源去处理数据流任务，如图 1 所示。

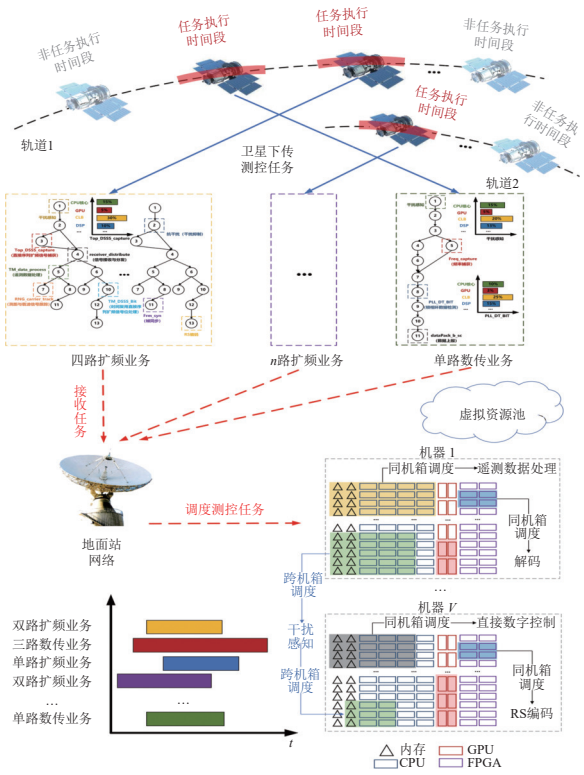


图 1　测控业务调度场景

本模型忽略卫星到地面站的数据传输时延和数据在不同的资源模块间传输的时间。以此，给出如下定义。

定义 1：卫星业务与卫星任务。卫星集合 $\boldsymbol{S}=\{s_1,s_2,\cdots,s_C\}$，其中 C 为卫星总数。对任意一颗卫星 $s_i\in\boldsymbol{S}$，其到达时间为 T_c^i，过境时长为 T_r^i。卫星集合 $\boldsymbol{S}$ 发送的业务总数为 P，用集合 $\boldsymbol{B}$ 表示 $\boldsymbol{S}$ 中包括的所有卫星业务，则有 $\boldsymbol{B}=\{\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_P\}$，其中任意 $\boldsymbol{b}_i\in\boldsymbol{B}$ 为地面站常规业务之一（测控、中低速数传、高速数传）。图 2 为一种典型测控业务模式信号处理流程图，卫星各业务可以拆分成多个卫星任务的组合，业务包含增益控制 1、数字下变频、增益控制 2、锁相环 1、锁相环 2、捕获 6 个任务，卫星任务总数为 Q，用集合 $\boldsymbol{A}=\{a_1,a_2,\cdots,a_Q\}$ 表示 $\boldsymbol{B}$ 包含的所有卫星任务。在卫星过境期间，会同时向地面发送多路卫星业务，多路业务又由多个任务组成，如图 3 所示。

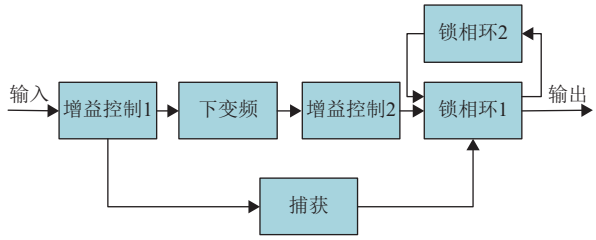


图 2　一种典型测控业务模式信号处理流程

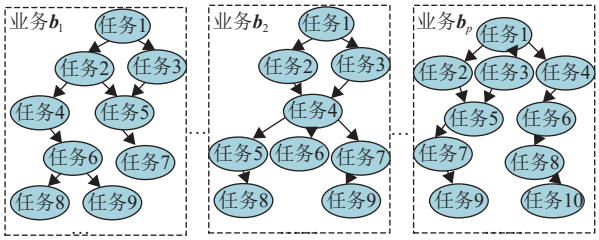


图 3　单颗卫星发送的业务集合

定义 2：异构资源池。地面站资源池是由多个处理节点组成的复杂信号处理系统，每个节点内集成了多个异构处理模块，模块内部署类型、数量各异的 FPGA、DSP、CPU 等多种处理芯片资源。跨节点、跨模块以及模块内的各类异构芯片资源均依托高速串行总线实现互连，最终构建出一个高度协同的异构计算资源池。资源池处理节点数量记为 V，用 $\boldsymbol{M}=\{m_1,m_2,\cdots,m_V\}$ 表示处理节点集合，异构资源池通过虚拟化管理技术抽象成 N 类芯片资源，对于任一处理节点 $m_i\in\boldsymbol{M}$，使用 $l_{i,j}^t$ 表示编号为 i 的处理节点中第 j 类芯片资源在 t 时刻的可用数目，同一路业务的任务放置于同一处理节点的比例的最大值为 R_t^v。

定义 3：资源启动等待时延。每个任务需要占用特定类别和数量的资源，并且资源启动结束后，任务才可以执行，这段启动时间即资源启动等待时延。将任务 $a_i\in\boldsymbol{A}$ 需要占用第 j 类资源数目记为 $r_{i,j}$，任务放置的处理节点编号为 $f_{a,i}$，记 T_x^i 为执行 a_i 所需要的资源启动等待时延。

定义 4：任务开始占用资源与完成时刻。任务 $a_i\in\boldsymbol{A}$ 的开始占用资源时刻记为 T_o^i，记 T_h^i 为此任务的资源释放时间，即该任务的完成时刻。

1.2　优化目标及约束条件

总完工时间定义为从第一路卫星业务到达至所有卫星业务执行结束所使用的时间。本文的优化目标是最小化总完工时间，从而可以提升资源池处理卫星业务的效率。优化目标的公式描述如下：

$$\min(T_m)=\min\left[\max_{a_i\in\boldsymbol{A}}\left(T_h^i\right)-\min_{s_k\in\boldsymbol{S}}\left(T_c^k\right)\right] \tag{1}$$

式中，$\max\left(T_h^i\right)$ 指 $\boldsymbol{A}$ 中最晚结束的卫星任务的资源释放时刻；$\min\left(T_c^k\right)$ 是指 C 中最早到达的卫星的到达时刻，二者的差值即总完工时间 T_m。针对优化目标，设立下述约束条件：

$$r_{i,n}\leqslant l_{f_{a,i},n}^t,\forall t\in\left[T_o^i,T_h^i\right],\ \forall a_i\in\boldsymbol{A},\ \forall n\in[1,N] \tag{2}$$

$$T_o^i \geqslant T_c^k,\ \forall a_i \in s_k \tag{3}$$

$$T_o^i + T_x^i = T_o^{i'} + T_x^{i'},\ \forall a_i,\ a_{i'} \in \boldsymbol{b}_j \tag{4}$$

$$T_h^i = T_o^i + T_x^i + T_k^r, \forall a_i \in s_k \tag{5}$$

式 (2) 表示卫星任务a_i在资源占用期间，所需的资源不能超出对应处理节点的可用空闲资源数目。式 (3) 表示任务a_i的资源启动时刻需要在所属卫星s_k到达之后。式 (4) 中$T_o^i + T_x^i$和$T_o^{i'} + T_x^{i'}$表示$\boldsymbol{b}_j$的任意两个子任务的开始执行时刻，两式相等说明$\boldsymbol{b}_j$中的所有子任务启动结束后，在同一时刻开始执行。式 (5) 表示任务的执行时长为所属卫星的过境时长。

1.3 问题特点与求解难点

本文针对航天测控资源池调度问题构建优化模型，其优化目标为最小化测控业务总完工时间，核心是求解一个以总完工时间最小化为目标的动态约束组合优化问题。该模型的复杂性与求解难点主要体现在以下方面。

1）混合整数非线性规划（mixed-integer non-linear programming, MINLP）与 NP-hard 复杂性：从数学形式上看，“多任务航天测控站资源池调度”模型是典型的 MINLP 问题。其复杂性主要源于两点，首先是整数决策变量，模型需确定每个测控业务在异构资源池中的分配方案，这涉及大量的 0 和 1 整数变量，用于决策各业务中如增益控制、捕获、锁相环等任务在某芯片资源上的分配与否。其次是非线性约束，模型中包含复杂的动态时序与同步约束。如“当且仅当所有相关资源准备就绪，测控业务才可同步开始”这类约束具有强逻辑关系，无法通过简单的线性函数描述，从而引入非线性特性。整数变量的存在与非线性约束的耦合，使得该问题成为一个已被证明的 NP-hard 问题。这意味着随着任务与设备数量的增加，可行解空间将呈现指数级增长，精确求解的计算复杂度变得难以承受。

2）非凸性、局部最优与全局最优的存在性：① 显著的非凸性与大量局部最优解：模型的非凸性主要源于两个方面。首先，0～1 决策变量的存在使得可行域在本质上是离散且非连通的，直接破坏了空间的凸性。其次，上述的逻辑与同步约束导致可行域呈现出不规则的几何形态。根据优化理论，对于此类具有组合特性的优化问题，其解空间的“邻域结构”与变量的离散组合紧密相关。微小的决策变动（如改变一个任务的分配）就可能导致目标函数值的剧烈跳跃，从而在解空间中形成大量的“峰”与“谷”，即品质不一的局部最优解[24-25]。因此，“存在大量局部最优解”是此类问题的一个内在数学属性。② 全局最优解的理论存在性：依据魏尔斯特拉斯极值定理，在非空紧集上的连续函数必能取得其最大值和最小值，其离散模拟推广更为直接：当一个函数从一个非空的有限集合映射到实数集时，该函数的值域必然是一个有限的实数集合，而任何非空的有限实数集都必然包含一个最大元和一个最小元[26-27]。因此，本模型确保存在一个全局最优解（即总完工时间最短的调度方案）。

3）求解目标的确立与算法选型依据：鉴于问题的 NP-hard 复杂性和充满局部陷阱的非凸解空间，任何旨在遍历搜索的精确算法（如分支定界法）在面对实际规模问题时都将遭遇“组合爆炸”，不具备工程实用性。因此，本文的研究目标并非追求在多项式时间内保证找到并证明该全局最优解，而是设计一种高效的元启发式算法（遗传算法），旨在有限的、可接受的计算时间内，尽可能地逼近这个理论上存在的全局最优解，即搜索到一个高质量的、工程上可行的满意解（或称近优解）。

综上所述，本调度问题的核心挑战在于其 NP-hard、非凸、多约束耦合的内在特性。这些特性决定了传统基于梯度的凸优化方法难以适用，而必须采用遗传算法这类善于跳出局部最优，能够在复杂解空间中进行全局探索的元启发式算法。

2 基于群体扰动的遗传算法的设计

针对本文的复杂地面站资源优化模型求解 NP-hard 问题非凸性特征及约束复杂性，经典的元启发式方法——遗传算法（genetic algorithm, GA）展现出良好的适用性。其核心优势在于：首先，GA 能够在庞大且非凸的解空间中进行全局性、并行的随机搜索，有效避免了因问题非凸性而使传统优化方法陷入局部最优的困境。其次，其灵活的染色体编码机制与适应度函数设计，能够自然地将模型中复杂的分配决策（离散变量）与多维约束（如时序、同步、空间布局等）融入求解过程，具备处理复杂约束的先天优势。

然而，在求解本模型这类强约束、高耦合的调度问题时，传统遗传算法也暴露出其固有缺陷，即在迭代后期常因“早熟收敛”导致种群多样性锐减，从而收敛于局部最优解，难以满足实际应用对

高质量解的严苛要求。

为突破这一瓶颈，本文提出一种基于种群扰动的改进遗传算法（genetic algorithm based on population perturbation, GA-P2）。该算法的核心创新在于，在标准遗传算子的基础上，并行引入一种由精英解引导的种群扰动策略。该策略能够在维持必要选择压力、确保算法收敛趋势的同时，对陷入停滞的种群进行动态重构，有效注入多样性。通过这种方式，GA-P2 能够实现算法全局探索（exploration）与局部开发（exploitation）能力的自适应平衡与深度协同，从而显著提升其跳出局部最优陷阱的能力，以期获得更高质量的调度方案。

与传统随机扰动策略旨在盲目跳出局部最优不同，本文提出的扰动机制是一种基于相似性的定向变异策略。它并非无差别地改变个体结构，而是保留当前迭代中优秀个体所蕴含的优良信息，再根据种群相似性特征对非优势个体实时变异。这种扰动带来了双重优势：其一，它通过相似性特征评估种群多样性，一旦多样性低于一定的阈值，算法将通过变异引入新的基因组合，赋予算法摆脱局部最优陷阱的能力；其二，它在交叉过程中将优秀策略的优良基因结构片段“注入”其他个体中，实现了优质信息的保留与传播，这不仅避免了大量无效搜索，更显著加速了算法向高质量解空间的收敛进程。为构建一个高效、鲁棒的求解框架，GA-P2 还系统地整合了与问题特性相匹配的全局组合编码方案，确保最优解不丢失的优秀策略保留机制，协同增强搜索性能的交叉与变异算子，共同构成了该算法的完整性设计。

2.1　全局组合编码方式

本文提出的 GA-P2 算法采用了一种策略编码方式，每个地面站资源优化调度策略包含测控任务部署到资源池处理节点的配置策略编码和测控任务处理顺序策略编码两部分，用元组表示为 $\left(f_{a,1},f_{a,2},\cdots,f_{a,Q},e_{b,1},e_{b,2},\cdots,e_{b,P}\right)$。其中，前 Q 个编码 $f_{a,i}$ 用于表示卫星测控任务集合 $\boldsymbol{A}$ 中所有任务放置的处理节点编号，后 P 个编码 $e_{b,j}$ 代表卫星测控业务集合 $\boldsymbol{B}$ 中 P 路卫星测控业务的处理顺序，即将 $\boldsymbol{b}_j$ 设定为 P 路卫星测控业务中第 $e_{b,j}+1$ 个被执行的业务。如图 4 所示，两种编码组合构建为一个调度策略，策略前 Q 个编码的取值范围为 $0\sim V-1$ 的随机整数，代表任务放置的处理节点编号；调度策略后 P 个编码包含 $0\sim P-1$ 的所有整数，且需满足互不相同且不遗漏的约束，否则为非法策略。初始化时，会按照上述编码规则随机生成 W 种不同的地面站资源调度策略。

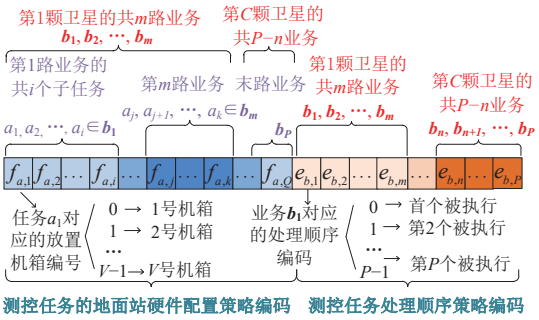


图 4　地面站资源调度策略编码

2.2　优秀策略保留机制

为减少地面站资源调度策略在迭代过程中的质量波动，本文引入优秀策略保留机制，通过归一化适应度 F 衡量调度策略的优劣，并选择优秀的资源调度策略直接保留到下一代，确保算法的稳定性。F 根据每个调度策略所对应的总体卫星测控业务全部处理完成时间计算得出，计算公式如下：

$$F=\frac{T_m^{\max}-T_m}{T_m^{\max}-T_m^{\min}} \tag{6}$$

式中，T_m 表示当前资源调度策略的总体卫星测控业务全部处理完成的时间；而 $T_m^{\max}$ 和 $T_m^{\min}$ 分别为当前种群中所有调度策略的最大和最小总体卫星测控业务全部处理完成的时间。通过上式进行归一化处理，将适应度映射到标准区间 [0, 1]，从而消除适应度的数量级差异。由式 (6) 可知，T_m 越小，F 越大，对应的资源调度策略越优秀。

在资源调度策略的选择阶段，算法选择保留适应度最高的 E 个优秀策略，不进行交叉和变异，直接选入下一代。

2.3　群体扰动机制

随着交叉和选择的进行，种群可能过早收敛到局部最优解，即种群中的不同个体的测控任务部署到资源池处理节点的配置策略编码和测控任务处理顺序策略编码具有较高的相似性。为避免陷入此困境，本研究引入群体扰动机制。该机制通过实时评估种群多样性，用于判断是否主动引入变异操作，以提高种群的多样性。

在生成新一代调度策略后，算法采用两两比对法对种群中的共 W 个策略进行相似性分析。整个种群共计两两比较 $(W(W-1))/2$ 次，每次记录 Q 个

测控任务部署到资源池处理节点的配置策略编码和 P 个处理顺序策略编码的相同位数。将第d次比较的相同位数记为G_d，以此计算衡量种群之间个体相似性的数值D：

$$D=\frac{1}{\frac{W(W+1)}{2}}\sum_{d=1}^{W(W-1)/2}\frac{G_d}{P+Q}=\sum_{d=1}^{W(W-1)/2}\frac{2G_d}{(P+Q)W(W+1)} \tag{7}$$

由式 (7) 可知，数值D越大，代表调度策略种群多样性越低，不同个体的测控任务部署到资源池处理节点的配置策略和测控任务处理顺序策略的相似性越高。若D超出$D_{\max}$算法设定的阈值时，说明种群可能趋于过早收敛，需要触发群体扰动机制，以恢复种群的多样性。具体措施为：对种群中的非优秀策略进行多次变异操作，增加个体的差异，直至D降低到设定的阈值以下。通过这种群体扰动机制，算法能够增强种群的多样性，降低陷入局部最优解的风险。

2.4 交叉和变异

GA-P2 算法通过模拟种群的交叉与变异过程，实现解空间的搜索。为保证总完工时间的正向优化，只有非优秀策略参与交叉与变异的过程，优秀策略则进入下一代以保留最优的总完工时间。同时，为了扩大搜索范围，本文采用非传统的多次随机部分交叉。

2.4.1 交叉操作

交叉操作的过程如图 5 所示，从种群中随机抽取两个个体，进行随机部分交叉。若出现冲突则需进行修正，最后拼接组合得到两个新的个体。

随机部分交叉的过程如图 5b 所示，在原调度策略的编码中，测控任务部署到资源池处理节点的配置策略编码和处理顺序策略编码分别随机选取 u 和 z 个随机长度的编码段，并对这些编码段进行交换。

对于测控任务部署到资源池处理节点的配置策略编码$\left(f_{a,1},f_{a,2},\cdots,f_{a,Q}\right)$而言，部分随机交叉可能导致大量任务被分配到同一处理节点上，导致总完工时间的负优化。如图 5b 中的个体 2，测控任务部署到资源池处理节点的配置策略编码的冲突段有大量测控任务集中于 3 号地面站处理节点。为此定义冲突 1 为：随机部分交叉后某一路业务的测控任务分配至同一地面站处理节点的比例超过算法设定的阈值$R_t^{\max}$。修正方法采用循环随机修正方法：通过迭代循环，将超出阈值的任务随机重新分配至其他可用的地面站处理节点，直到无任务占比超过设定阈值，确保资源分配的相对均衡。图 5c 展示了循环随机修正后的结果，这些任务随机更换到 5 号、4 号等地面站处理节点进行处理。对于测控任务处理顺序策略编码$(e_{b,1},e_{b,2},\cdots,e_{b,P})$，随机部分交叉后可能会出现测控任务处理顺序的重复和缺失，且重复和缺失同时出现。即不同测控任务的处理顺序相同，或某一顺位无测控任务。如图 5b 中个体 1，其测控任务处理顺序策略编码分别出现了重复的顺序编号 2。为此定义冲突 2 为：不同测控任务的处理顺序策略编码相同。为解决该问题，本研究采用匹配修正方法：识别交换片段与非交换片段间的重复位置，将非交换片段的重复编号替换为交换片段该位置交叉前的原始编号。

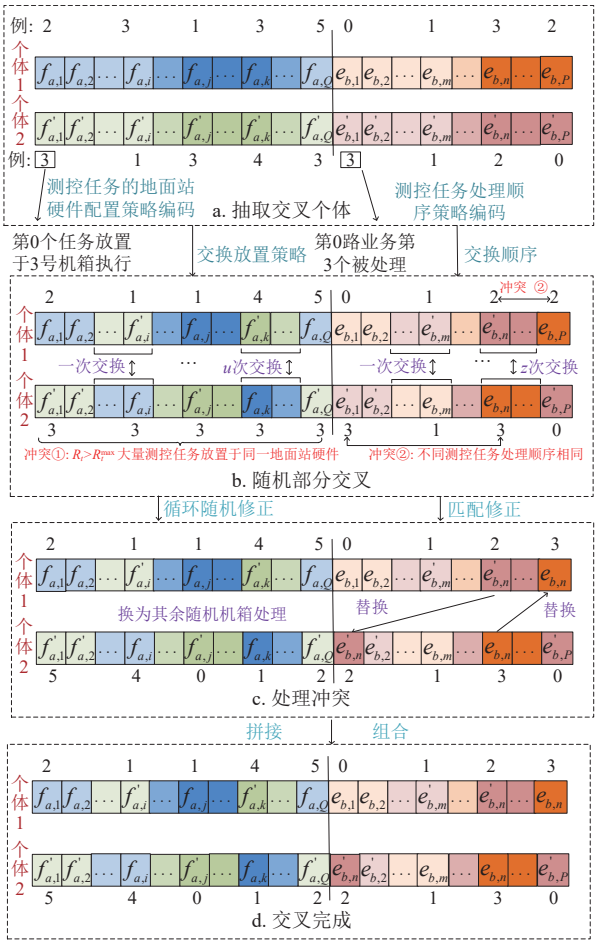


图 5 交叉过程

图 5c 展示了匹配修正的过程，个体 1 将交换片段以外的后一个编码 2，替换为前一个编码 2 交叉前对应位置的编码 3。如图 5d 所示，两部分编

码经过循环修正后，经组合拼接完成交叉操作。整个种群共计进行$(W-E)/2$次交叉操作。

2.4.2　变异操作

在卫星测控地面站资源池调度问题中，交叉操作仅能探索有限的解空间。为扩展搜索范围，本研究采用单点变异与两点变异并行方式：1）测控任务部署到资源池处理节点的配置策略编码$(f_{a,1}, f_{a,2},\cdots,f_{a,Q})$采用单点变异的方式，即随机选择测控任务$a_i$，将其随机放置到一台地面站处理节点进行处理。2）测控任务处理顺序策略编码$(e_{b,1},e_{b,2},\cdots,e_{b,P})$采用两点变异方式，即随机选择两个测控任务，交换其处理顺序。

2.5　GA-P2 整体框架

基于前文设计的全局组合编码方式、优秀策略保留机制、交叉与变异和群体扰动机制，GA-P2 算法的伪代码及整体框架如下所示。

初始化测控任务调度策略种群，随机生成每个个体的量化测控任务部署到资源池处理节点的配置策略编码和测控任务处理顺序编码

for i=0 to 最大迭代次数 do

根据总完工时间，通过式 (6) 计算个体适应度 F

按照 F 降序排序种群个体

选取前E个优秀策略，保留高质量策略个体

根据式 (7) 计算策略种群相似性数值D，量化测控任务部署到资源池处理节点的配置策略编码和测控任务处理顺序编码的差异

while D>阈值 $D_{\max}$ do

非优秀策略进行变异，随机改变测控任务部署到资源池处理节点的配置策略，交换测控任务处理顺序

根据式 (7) 重新计算 D

end while

for k =1 to $(W-E)/2$ do

非优秀策略进行多次随机部分交叉，得到新的测控任务部署到资源池处理节点的配置策略与测控任务处理顺序

while 存在冲突 do

修正随机部分交叉产生的冲突

end while

非优秀策略进行变异，随机改变测控任务部署到资源池处理节点的配置策略，交换测控任务处理顺序

end for

优秀策略与非优秀策略组成下一代种群

end for

返回最优策略及其总完工时间

3　实验测试

仿真实验生成的数据集中，卫星业务包含测控、中低速数传两种常规业务模式，其中测控业务各任务间关联关系及资源需求如图 6 和表 1 所示，中低速数传业务各任务间关联关系及资源需求如图 7 和表 2 所示。卫星数量可设置为 100、150、200、250，每种数据规模下，卫星在 1～1 000 s 内按照一维均匀分布随机到达，过境时间在 [5, 8] min 和 [8, 10] min 内的卫星各占总数的 30%，[10, 15] min 内的卫星占总数的 40%，携带 1 路业务的卫星占总数的 40%，携带 2～4 路业务的卫星各占总数的 20%，每路业务包含 10 个以上的任务，仿真业务配置情况如表 3 所示。

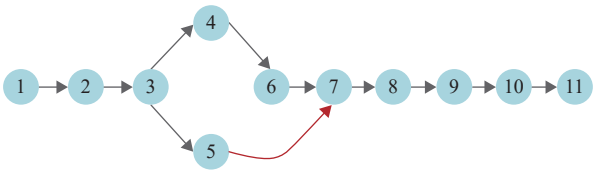


图 6　测控业务各任务间的关联关系

表 1　测控业务各任务资源需求表

任务序号	占用资源类型	占用资源情况
1	GPU	CUDA Core: 2; Execution Time: 551
2	CPU	CPU Core: 0.3; Execution Time: 562
3	CPU	CPU Core: 0.1; Execution Time: 513
4	CPU	CPU Core: 0.1; Execution Time: 513
5	FPGA	FPGA_CLB: 3694; FPGA_FF: 21877; FPGA_BRAM: 509; FPGA_DSP: 376; Execution Time: 610
6	CPU	CPU Core: 0.1; Execution Time: 513
7	CPU	CPU Core: 0.2; Execution Time: 545
8	CPU	CPU Core: 0.2; Execution Time: 545
9	FPGA	FPGA_CLB: 5545; FPGA_FF: 15162; FPGA_BRAM: 33; FPGA_DSP: 57; Execution Time: 702
10	CPU	CPU Core: 0.2; Execution Time: 545
11	CPU	CPU Core: 0.05; Execution Time: 498

地面站仿真 10 个处理节点，总资源数量通过调整各处理节点的 FPGA、CPU、GPU 个数可设置为 105、152、204，不同总资源数量下，各处理节点配置如表 4 所示。由于任务的到达时间不同，地

面设备只需要达到任务所需的资源总数的 50%～60% 以上即可视为地面站资源充足，即所有任务都可以在 1 000 s 的到达时间结束前完成部署。经测算，总资源数量 105、152、204 依次可模拟地面站资源少于、约等于、多于任务所需资源的 3 种情况。

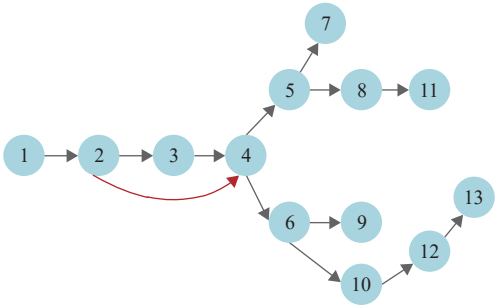


图 7 中低速数传业务各任务间的关联关系

表 2 中低速数传业务各任务资源需求表

任务序号	占用资源类型	占用资源情况
1	GPU	CUDA Core: 2; Execution Time: 551
2	CPU	CPU Core: 0.3; Execution Time: 562
3	FPGA	FPGA_CLB: 6320; FPGA_FF: 42237; FPGA_BRAM: 172; FPGA_DSP: 274; Execution Time: 781
4	CPU	CPU Core: 0.1333; Execution Time: 518
5	CPU	CPU Core: 0.2; Execution Time: 545
6	CPU	CPU Core: 0.1333; Execution Time: 518
7	CPU	CPU Core: 0.1; Execution Time: 513
8	CPU	CPU Core: 0.1333; Execution Time: 518
9	CPU	CPU Core: 0.2; Execution Time: 545
10	CPU	CPU Core: 0.25; Execution Time: 551
11	CPU	CPU Core: 0.1333; Execution Time: 518
12	CPU	CPU Core: 0.1; Execution Time: 513
13	FPGA	FPGA_CLB: 7722; FPGA_FF: 37862; FPGA_BRAM: 312; FPGA_DSP: 274; Execution Time: 816

表 3 不同卫星数量下任务数量配置情况

数量	过境时间区间/min	1路业务卫星数量	2路业务卫星数量	3路业务卫星数量	4路业务卫星数量
50	[5, 8)	6（3测控、3数传）	3（2测控、1数传）	3（1测控、2数传）	3（2测控、1数传）
	[8, 10)	6（3测控、3数传）	2（1测控、1数传）	3（1测控、2数传）	3（2测控、1数传）
	[10, 15]	7（3测控、4数传）	5（3测控、2数传）	4（2测控、2数传）	4（2测控、2数传）
100	[5, 8)	11（5测控、6数传）	6（3测控、3数传）	5（3测控、2数传）	6（3测控、3数传）
	[8, 10)	13（7测控、6数传）	6（3测控、3数传）	7（3测控、4数传）	6（3测控、3数传）
	[10, 15]	16（8测控、8数传）	8（4测控、4数传）	8（4测控、4数传）	8（4测控、4数传）
150	[5, 8)	16（8测控、8数传）	9（5测控、4数传）	9（4测控、5数传）	9（5测控、4数传）
	[8, 10)	19（9测控、10数传）	9（5测控、4数传）	8（4测控、4数传）	9（4测控、5数传）
	[10, 15]	24（12测控、12数传）	12（6测控、6数传）	13（7测控、6数传）	12（6测控、6数传）
200	[5, 8)	22（11测控、11数传）	12（6测控、6数传）	12（6测控、6数传）	12（6测控、6数传）
	[8, 10)	25（13测控、12数传）	12（6测控、6数传）	12（6测控、6数传）	12（6测控、6数传）
	[10, 15]	33（16测控、17数传）	16（8测控、8数传）	16（8测控、8数传）	16（8测控、8数传）
250	[5, 8)	30（15测控、15数传）	14（7测控、7数传）	15（8测控、7数传）	15（7测控、8数传）
	[8, 10)	29（15测控、14数传）	16（8测控、8数传）	15（7测控、8数传）	14（7测控、7数传）
	[10, 15]	41（21测控、20数传）	20（10测控、10数传）	20（10测控、10数传）	20（10测控、10数传）

3.1 对比算法描述

CFS-SWTF（completely fair scheduler-shortest waiting time first）：在所有已到达且未完成任务的卫星中，CFS[23] 按照设备的到达顺序依次执行每个设备的第 k 路业务，完成后再执行其第 k+1 路业务。SWTF 基于 SJF（shortest job first）[28]，对于新加入的卫星优先执行其前 k 路业务，将其包含的任务分配到当前等待时间最短的设备上，若存在多个设备的最短等待时间相同，则随机选择一个目标设备。

CFS-MCRP（completely fair scheduler-most compact resource placement）：基于 CFS[23] 规则确定业务的执行顺序后，为每个设备设定一个初始紧凑值（默认为 0）。每当分配一个任务时，该设备的紧凑值增加，增加数值为该任务占用设备资源比例与占用时间的乘积。任务总是分配到当前紧凑值最小的设备上；若有多个设备紧凑值相同且最小，则选择设备编号较小的设备。

FIFO-SWTF（first-in, first-out-shortest waiting time first）：FIFO[29] 按照卫星的到达时间顺序依次执行先到卫星的业务，之后按照 SWTF 的规则确定业务包含的具体任务的执行设备。

FIFO-MCRP（first-in, first-out-most compact resource placement）：在根据 FIFO[25] 执行顺序后，采用 MCRP 规则确定具体任务的执行设备。

GA（genetic algorithm）：遗传算法[15] 将业务

的执行顺序和任务的分配设备编码为染色体，并通过模拟自然选择、交叉和变异等遗传操作，在迭代过程中不断演化，逐步淘汰适应度低的个体，保留并产生适应度高的后代。

表 4　不同总资源数量下，各处理节点配置情况

总数量	处理节点号	CPU型号及数量	GPU型号及数量	FPGA型号及数量
105	处理节点1	Xeon E-2324G×4	GeForce RTX 3060Ti×4	XC7VX415T×4
	处理节点2	Xeon E-2314×4	GeForce RTX 3070×2	XC7VX485T×3
	处理节点3	Core i7-13700T×2	TESLA K40×4	VC709-XC7VLX690T×2
	处理节点4	Core i9-12900TE×2	GeForce RTX 3060Ti×4	XC7VX415T×4
	处理节点5	Xeon E5410×6	GeForce RTX 3070×2	XC7VX485T×3
	处理节点6	Xeon E5420×4	TESLA K40×4	VC709-XC7VLX690T×2
	处理节点7	Opteron 2356×3	TESLA K40×4	XC7VX415T×4
	处理节点8	Xeon E-2236×6	TESLA K40×4	XC7VX415T×4
	处理节点9	Xeon E-2288G×4	GeForce RTX 3060Ti×4	XC7VX485T×3
	处理节点10	Xeon E-2334×2	GeForce RTX 3060Ti×4	XC7VX485T×3
152	处理节点1	Core i5-13500TE×3	GeForce RTX 3060Ti×4	XC7VX415T×5
	处理节点2	Core i7-13700E×3	GeForce RTX 3070×3	XC7VX485T×4
	处理节点3	Core i7-13700T×3	TESLA K40×4	VC709-XC7VLX690T×4
	处理节点4	Core i9-12900TE×3	GeForce RTX 3060Ti×4	XC7VX415T×5
	处理节点5	Xeon E5410×10	GeForce RTX 3070×3	XC7VX485T×4
	处理节点6	Xeon E5420×10	TESLA K40×4	VC709-XC7VLX690T×4
	处理节点7	Opteron 2356×10	TESLA K40×4	XC7VX415T×5
	处理节点8	Xeon E-2236×10	TESLA K40×4	XC7VX415T×5
	处理节点9	Xeon E-2288G×10	GeForce RTX 3060Ti×4	XC7VX485T×4
	处理节点10	Xeon E5410×10	GeForce RTX 3060Ti×4	XC7VX485T×4
204	处理节点1	Core i5-13500TE×3	GeForce RTX 3060Ti×4	XC7VX415T×20
	处理节点2	Core i7-13700E×3	GeForce RTX 3070×3	XC7VX485T×8
	处理节点3	Core i7-13700T×3	TESLA K40×4	VC709-XC7VLX690T×8
	处理节点4	Core i9-12900TE×3	GeForce RTX 3060Ti×4	XC7VX415T×10
	处理节点5	Xeon E5410×10	GeForce RTX 3070×3	XC7VX485T×8
	处理节点6	Xeon E5420×10	TESLA K40×4	VC709-XC7VLX690T×8
	处理节点7	Opteron 2356×10	TESLA K40×4	XC7VX415T×10
	处理节点8	Xeon E-2236×10	TESLA K40×4	XC7VX415T×10
	处理节点9	Xeon E-2288G×10	GeForce RTX 3060Ti×4	XC7VX485T×8
	处理节点10	Xeon E5410×10	GeForce RTX 3060Ti×4	XC7VX485T×48

3.2　实验测试结果对比

本节以地面站资源调度问题为研究对象，对所提出的 GA-P2 算法以及 GA、FIFO、CFS、SWTF 和 MCRP 共 6 种算法进行了对比测试。实验环境设定为地面站资源数少于任务需求，并针对不同任务规模下的业务结束时间进行了统计与分析。

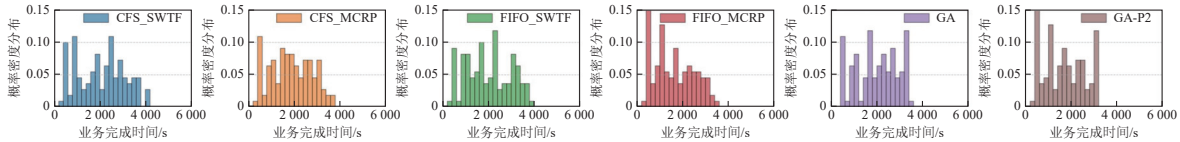
如图 8 所示，实验结果表明，在总完工时间这一关键性能指标上，与其余 5 种算法相比，GA-P2 算法的总完工时间始终保持最低，展现出显著的优越性。具体而言，其性能优于除 GA 算法外的其他算法 300 s 以上，且相较于 GA 算法也能降低 100 s 以上。

在算法稳定性方面，GA-P2、GA 和 2 种 MCRP 算法在任务规模变化时，其总完工时间的波动较小，表现出良好的鲁棒性。从概率密度分布图（图 8）可以看出，这些算法的分布曲线更为集中，峰值更高，进一步佐证了其稳定性。

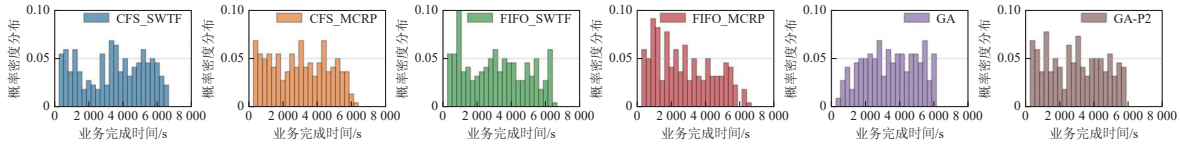
相较之下，SWTF 算法的稳定性较差，在某些特定场景下可能会产生极高的总完工时间。如在 150 颗卫星任务规模下，FIFO-SWTF 算法的总完工时间是所有算法中最长的，比最优的 GA-P2 算法高出 88.7%。

综合来看，FIFO 和 CFS 算法的性能表现依赖于具体的实验条件。若仅比较稳定性更优的 MCRP 算法，FIFO 算法的性能略优于 CFS-MCRP 算法，但二者性能均不及 GA-P2 算法。

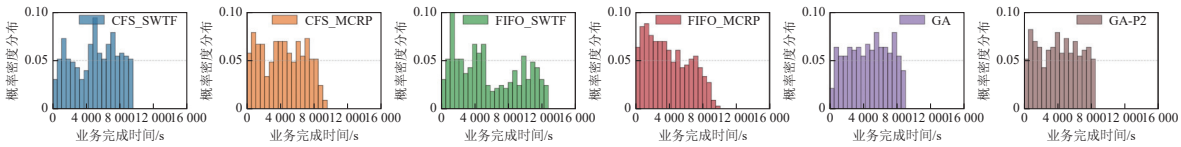
实验数据表明，MCRP 算法在整体总完工时间及稳定性方面优于 SWTF 算法，但后者在特定场景下可能表现出局部优势。如图 10 中卫星数量为 100 时，CFS-SWTF 算法的总完工时间略高于 CFS-MCRP 算法。又如在卫星数量为 200 的情况下，FIFO-SWTF 算法的总完工时间优于 FIFO-MCRP 算法。



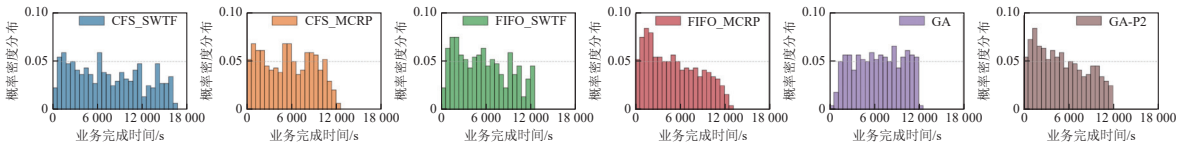
a. 地面资源过饱和，卫星数量50颗规模下，业务完成时间概率密度分布



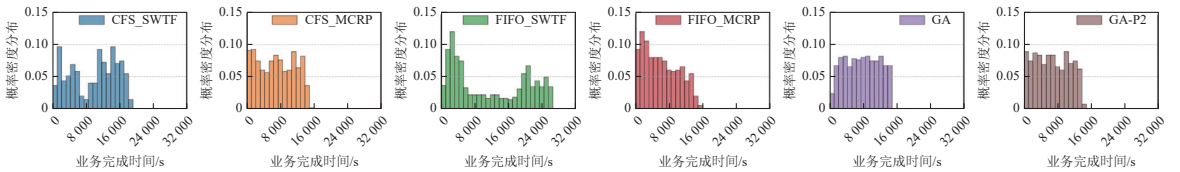
b. 地面资源过饱和，卫星数量100颗规模下，业务完成时间概率密度分布



c. 地面资源过饱和，卫星数量150颗规模下，业务完成时间概率密度分布



d. 地面资源过饱和，卫星数量200颗规模下，业务完成时间概率密度分布



e. 地面资源过饱和，卫星数量250颗规模下，业务完成时间概率密度分布

图 8　地面资源过饱和，不同规模卫星业务完成时间概率密度分布图

为验证不同算法在地面站资源变化下的鲁棒性，本文在保持 CPU、GPU 及 FPGA 型号不变的基础上，通过调整地面站资源规模，分别模拟资源饱和（资源总量与任务需求基本持平）与资源不饱和（资源总量富余、超出任务需求）两类情况，来对业务完工时间的概率密度分布进行仿真验证，仿真结果如图 9 与图 10 所示。

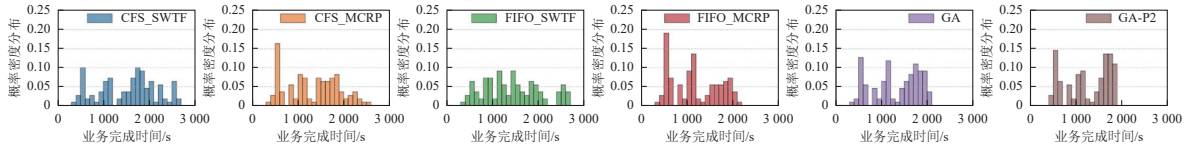
从图 9 和图 10 中可以看出，GA-P2 算法在各类资源条件与任务规模下始终维持最低总完工时间，印证了此算法的强鲁棒性。在地面站资源达到饱和阈值后，MCRP 算法和 GA 算法的总完工时间与 GA-P2 算法趋近，表明在资源充足的条件下，MCRP 算法及 GA 算法具有较高的鲁棒性和较优的总完工时间。

由于 SWTF 算法的鲁棒性较差，故通过对比 FIFO-MCRP 和 CFS-MCRP 算法来评估 FIFO 和 CFS 在业务处理顺序策略上的性能差异。从图 9 和图 10 中可以看出，在相同的资源条件和卫星任务规模下，这两种算法的总完工时间相近，即它们具有类似的优劣性和鲁棒性。

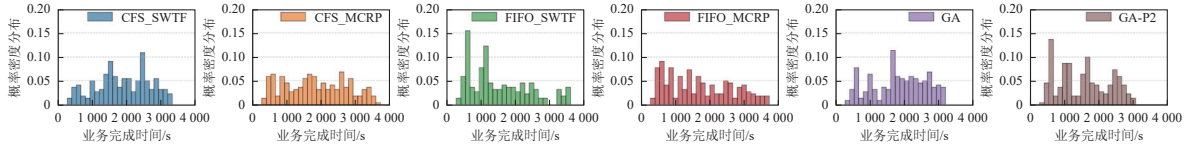
为研究影响总完工时间的因素，通过定义任务等待时间（卫星到达时刻至任务实际启动时刻的时延）并引入延迟区间分布比例指标，用于量化分析延迟分布对总完工时间的影响规律。

由于 50 颗卫星时的延迟均较小，故此处研究 100～250 颗卫星时的延迟对于总完工时间的影响。如表 5～表 8 所示，在 100 颗卫星的场景下，GA-P2 算法在短延迟区间（<3 min）的任务占比为 17.27%，GA 算法在短延迟区间（<3 min）的任务占比为 15%，CFS-MCRP 算法在短延迟区间（<3 min）的任务占比为 16.82%，三者总完工时间显著高于其他算法。

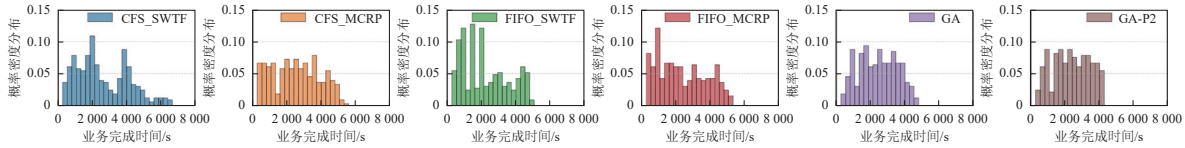
在延迟分布中，低延迟占比的提升往往对应整体完工时间缩短。经对比分析表明，GA-P2 算法体现出更优的资源调度能力。以卫星 150 颗的场景为例，FIFO-SWTF 算法的长延迟区间（>10 min）任务占比高达 90.91%，短延迟占比仅 3.03%，最终其总完工时间相较于 GA-P2 算法恶化 41%，直观印证了算法具有有效缩短长延迟任务聚集区间的效果。



a. 地面资源饱和，卫星数量50颗规模下，业务完成时间概率密度分布

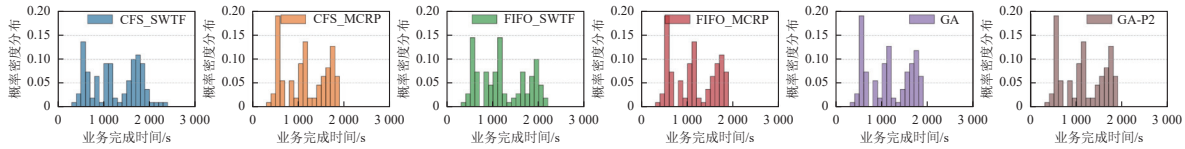


b. 地面资源饱和，卫星数量100颗规模下，业务完成时间概率密度分布

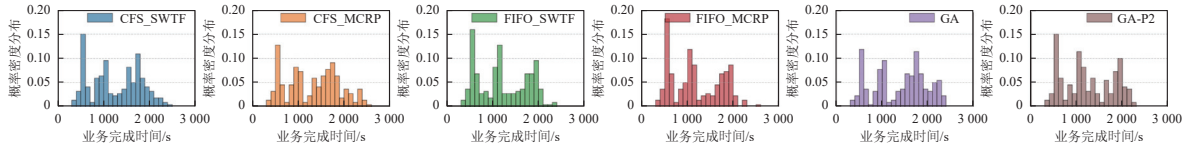


c. 地面资源饱和，卫星数量150颗规模下，业务完成时间概率密度分布

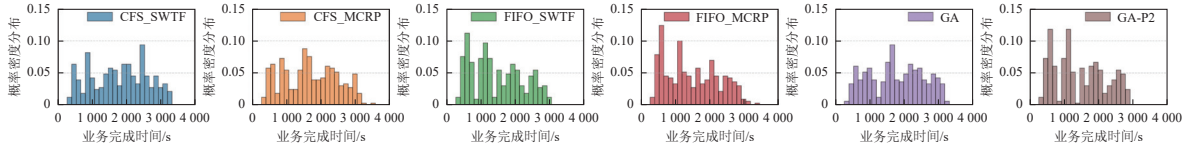
图 9　地面资源饱和，不同规模卫星业务完成时间概率密度分布图



a. 地面资源不饱和，卫星数量50颗规模下，业务完成时间概率密度分布



b. 地面资源不饱和，卫星数量100颗规模下，业务完成时间概率密度分布



c. 地面资源不饱和，卫星数量150颗规模下，业务完成时间概率密度分布

图 10　地面资源不饱和，不同规模卫星业务完成时间概率密度分布图

表 5　卫星数为 100（220 路业务 2 632 个任务）

延迟区间/min	CFS-SWTF/%	CFS-MCRP/%	FIFO-SWTF/%	FIFO-MCRP/%	GA/%	GA-P2/%
[0～3)	10.91	16.82	10.00	10.00	15.00	17.27
[3～10)	14.55	14.09	13.64	11.82	13.18	15.45
>10	74.55	69.09	76.36	78.18	71.82	67.27

表 6　卫星数为 150（330 路业务 3 948 个任务）

延迟区间/min	CFS-SWTF/%	CFS-MCRP/%	FIFO-SWTF/%	FIFO-MCRP/%	GA/%	GA-P2/%
[0～3)	3.33	7.58	3.03	7.58	7.58	8.79
[3～10)	8.18	11.51	6.06	8.18	9.70	10.30
>10	88.48	80.91	90.91	84.24	82.72	80.91

表 7　卫星数为 200（440 路业务 5 264 个任务）

延迟区间/min	CFS-SWTF/%	CFS-MCRP/%	FIFO-SWTF/%	FIFO-MCRP/%	GA/%	GA-P2/%
[0～3)	2.27	5.68	2.27	5.45	5.23	5.68
[3～10)	4.32	7.50	4.32	7.95	7.95	7.95
>10	93.41	86.82	93.41	86.59	86.82	86.37

表 8　卫星数为 250（550 路业务 6 580 个任务）

延迟区间/min	CFS-SWTF/%	CFS-MCRP/%	FIFO-SWTF/%	FIFO-MCRP/%	GA/%	GA-P2/%
[0～3)	1.82	4.72	1.82	4.73	4.55	6.18
[3～10)	1.82	6.55	1.82	6.00	5.09	7.64
>10	96.36	88.73	96.36	89.27	90.36	86.18

上述内容是地面资源不饱和、资源饱和和资源过饱和 3 种典型资源负载情况下，对不同规模卫星业务完成时间的概率密度分布仿真。下面将进一步以处理资源负载为变量，对不同调度算法下地面站执行业务的满足度进行仿真。对于航天测控业务集处理特性而言，不失一般性，仿真不同资源负载下任务满足率的预置条件中，业务数量、业务处理完成时间阈值的选择不会影响总体分布趋势，本次仿真中卫星数量为 150，业务完成时间为 3 000 s，仿真结果如图 11 所示。

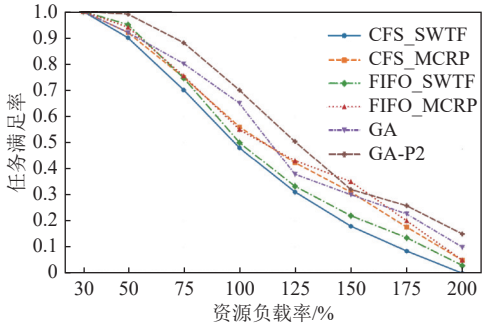


图 11 不同资源负载下任务满足率变化曲线

仿真结果表明，MCRP 算法在资源调度时因考虑了处理资源的负载均衡特性，能充分发挥计算资源的并行处理能力，在整体总完工时间及稳定性方面优于 SWTF 算法。GA 算法通过全局寻优，普遍比 MCRP、SWTF 算法优异，但在资源过饱和情况下，因过度追求快速收敛，易陷入局部最优，导致任务满足率恶化。GA-P2 算法通过加入基于种群的扰动，全局具有有效性，因此全局拥有更好的任务满足率。

4 结束语

针对航天测控地面站异构资源池的调度管理问题，本文通过建立资源调度优化模型，提出基于群体扰动的遗传调度算法（GA-P2）。实验验证表明，与对比算法相比，GA-P2 算法在总完工时间、任务延迟分布上表现更优，能快速实现多任务与资源池的动态匹配，显著提升任务处理效率，且在资源高负载时稳定性突出，有效推动了地面站资源利用率与操作效率的双重提升。

参 考 文 献

[1] 中国航天科技集团有限公司. 中国航天科技活动蓝皮书（2024 年）[M]. 北京: 中国航天科技集团有限公司, 2024.
China Aerospace Science and Technology Corporation. Blue book of China aerospace science and technology activities (2024)[M]. Beijing: China Aerospace Science and Technology Corporation, 2024.

[2] National Center for Science and Engineering Statistics (NCSES). Survey of graduate students and postdoctorates in science and engineering: Fall 2021[EB/OL]. [2025-07-01]. https://ncses.nsf.gov/pubs/nsf23326.

[3] 靳肖闪, 李军, 刘湘辉, 等. 基于拉格朗日松弛与最大分支算法的卫星成像调度算法[J]. 宇航学报, 2008(2): 694-699.
JIN X S, LI J, LIU X H, et al. Satellite imaging scheduling algorithm based on lagrangian relaxation and branch and bound algorithm[J]. Journal of Astronautics, 2008(2): 694-699.

[4] XU R, CHEN H, LIANG X, et al. Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization[J]. Expert Systems with Applications, 2016, 51: 195-206.

[5] SHE Y C, LI S, ZHAO Y B. Onboard mission planning for agile satellite using modified mixed-integer linear programming[J]. Aerospace Science and Technology, 2018, 72: 204-216.

[6] LIU Z, XIONG W, XIONG M. A large-scale scheduling method for multiple agile optical satellites[J]. CMES-Computer Modeling in Engineering & Sciences, 2023, 136(2): 1143-1163.

[7] ZHANG Z, ZHANG N, FENG Z. Multi-satellite control resource scheduling based on ant colony optimization[J]. Expert Systems with Applications, 2014, 41(6): 2816-2823.

[8] ZHANG Z, HU F, ZHANG N. Ant colony algorithm for satellite control resource scheduling problem[J]. Applied Intelligence, 2018, 48: 3295-3305.

[9] HE L, LIU X L, CHEN Y W, et al. Hierarchical scheduling for real-time agile satellite task scheduling in a dynamic environment[J]. Advances in Space Research, 2019, 63(2): 897-912.

[10] LU Z Z, SHEN X, LI D, et al. Integrated imaging mission planning modeling method for multi-type targets for super-agile earth observation satellite[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2022, 15: 4156-4169.

[11] YAN B, WANG Y J, XIA W, et al. An improved method for satellite emergency mission scheduling scheme group decision-making incorporating PSO and multimoora[J]. Advances in Space Research, 2022, 42(4): 3837-3853.

[12] ZHU W M, HU X X, XIA W, et al. A two-phase genetic annealing method for integrated earth observation satellite scheduling problems[J]. Methodologies and Application, 2019, 23: 181-196.

[13] ZHANG J, XING L. An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem[J]. Computers & Operations Research, 2022, 139: 105626.

[14] 黄磊, 徐浩. 基于粒子群算法的卫星任务地面站资源调度方法研究[J]. 中国空间科学技术, 2021, 41(2): 56-63.
HUANG L, XU H. Research on scheduling method for

satellite task ground station resources based on particle swarm optimization[J]. Chinese Space Science and Technology, 2021, 41(2): 56-63.

[15] CHEN M, WEN J, SONG Y J, et al. A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem[J]. Swarm and Evolutionary Computation, 2021, 65: 100912.

[16] 赵立, 周明, 吴峰. 面向卫星常规测控任务的地面站资源调度方法[J]. 宇航学报, 2019, 40(5): 78-85.
ZHAO L, ZHOU M, WU F. Scheduling method for ground station resources oriented to conventional satellite TT&C tasks[J]. Journal of Astronautics, 2019, 40(5): 78-85.

[17] SUN X, REN Y, YU L. Multi-adaptive strategies-based higher-order quantum genetic algorithm for agile remote sensing satellite scheduling problem[J]. Sensors, 2024, 24(15): 4938.

[18] SONG Y, OU J, WU J, et al. A cluster-based genetic optimization method for satellite range scheduling system[J]. Swarm and Evolutionary Computation, 2023, 79: 1-10.

[19] 姚敏, 王绪芝, 赵敏. 无人机群协同作战任务分配方法研究[J]. 电子科技大学学报, 2013, 42(5): 723-727.
YAO M, WANG X Z, ZHAO M. Research on task allocation method for cooperative combat operations of UAV swarms[J]. Journal of University of Electronic Science and Technology of China, 2013, 42(5): 723-727.

[20] SHI Q, LI L, FANG Z R, et al. Efficient and fair PPO-based integrated scheduling method for multiple tasks of SATech-01 satellite[J]. Chinese Journal of Aeronautics, 2024, 37(2): 417-430.

[21] LI H, LI Y J, MENG Q Q, et al. An onboard periodic rescheduling algorithm for satellite observation scheduling problem with common dynamic tasks[J]. Advances in Space Research, 2024, 73(10): 5242-5253.

[22] 孙伟, 杨洋. 一种面向卫星地面站资源调度的策略生成方法[J]. 系统工程与电子技术, 2022, 44(8): 23-30.
SUN W, YANG Y. A strategy formulation method targeted at satellite ground station resource scheduling[J]. Systems Engineering and Electronics, 2022, 44(8): 23-30.

[23] PABLA C S. Completely fair scheduler[J]. Linux Journal, 2009, 184: 4.

[24] WOLSEY L A, NEMHAUSER G L. Integer and combinatorial optimization[M]. Hoboken: John Wiley & Sons, 1999: 75-82.

[25] BIEGLER L T, GROSSMANN I E, WESTERBERG A W. Mixed integer nonlinear programming[M]. Berlin: Springer Science & Business Media, 2011: 133-135.

[26] PAPADIMITRIOU C H, STEIGLITZ K. Combinatorial optimization: Algorithms and complexity[M]. North Chelmsford: Courier Corporation, 1998: 338-344.

[27] KORTE B, VYGEN J. Combinatorial optimization: Theory and algorithms[M]. Berlin, Heidelberg: Springer, 2008: 212-218.

[28] SIAHAAN A P. Comparison analysis of CPU scheduling: FCFS, SJF and Round Robin[J]. International Journal of Engineering Development and Research, 2016, 4(3): 124-132.

[29] MORSE D, RICHARDSON G. The LIFO/FIFO decision[J]. Journal of Accounting Research, 1983(2): 106-127.

责任编辑　刘飞阳