

基于混合启发式遗传优化算法的异构任务调度研究

李明¹ 李中武¹ 邢亮¹ 张文豪¹ 王蝶¹ 李振彦¹

LI Ming LI Zhongwu XING Liang ZHANG Wenhao WANG Die LI Zhenyan

摘要

针对异构计算环境中传统遗传算法易陷入早熟收敛和局部最优的困境,导致全局搜索能力不足的问题,文章设计了一种遗传优化算法。该算法采用双染色体编码机制,兼顾任务拓扑约束与资源映射灵活性;运用 HEFT 启发式规则结合随机扰动策略构造高质量初始种群;针对双染色体结构特征,分别设计自适应遗传算子,任务序列染色体采用顺序交叉确保依赖合法性,资源分配染色体采用两点交叉维护基因的多样性;同时算法中引入自适应变异概率调节机制,实现种群演化的动态控制;最后,嵌入周期性邻域搜索对精英解微调,提升局部优化效率。在 CPU-GPU 组成的异构多核处理器平台上,采用随机生成的任务图测试集对算法进行测试,结果表明:与 HEFT 和蚁群算法相比,所采用的遗传优化算法明显降低了任务的调度长度,提升了算法的收敛速度,有效优化了异构计算平台上的任务调度性能。

关键词

遗传算法;任务调度;异构计算;自适应交叉变异;双染色体编码

doi: 10.3969/j.issn.1672-9528.2026.01.003

0 引言

随着计算密集型应用的迅猛发展,异构计算平台逐渐成为现代高性能计算体系的核心架构。这类平台通过整合 CPU、GPU、FPGA 等多元化处理器资源,能够充分利用硬件协同优势,但资源特性差异与任务需求多样性,使任务调度成为影响系统效能的关键因素。

异构计算平台任务调度是典型的 NP-hard 问题^[1],现有求解方法各有局限。比如,贪心算法容易陷入局部最优;而启发式方法虽能利用领域知识提升求解效果,但其强依赖性导致方法的普适性不足。遗传算法作为全局优化方法,因并行搜索、自适应调整等特性,在异构调度中展现出潜力,但传统遗传算法仍存在初始种群质量参差不齐、交叉变异算子未充分适配异构资源特性、收敛速度慢且易早熟等缺陷,难以达到实际应用要求。为解决上述问题,本研究构建了一种面向异构计算环境的遗传优化算法。该算法结合异构资源的性能特征与任务需求,采用启发式规则生成初始种群,提高初始解的质量与多样性;设计自适应的交叉变异算子体系,实现对种群进化方向的动态调整;最后,引入局部搜索策略,对遗传算法输出的最优解进行微调,增强局部优化能力,进一步提升调度性能。本文研究内容组织如下:首先回顾遗传算法在异构计算系统任务调度领域的研究现状;随后阐述任务调度问题的数学建模方法;接着详细介绍所提出的遗传优化算法设计方案,重点描述双重染色

体编码策略及自适应交叉变异机制;然后给出算法的分析测试结果;最后给出全文的总结。

1 相关工作

异构计算环境中的任务调度研究引起了广泛关注,众多研究学者以提高处理器性能、资源利用率和系统效率等为目标对此进行了大量研究^[2-6]。从 CPU 到 GPU、ASIC 和 FPGA,研究人员对具有多种计算单元的异构计算系统进行了深入的研究,这些系统整合了具有不同指令集的处理器的架构,促进了多个内核之间的平滑协作以高效处理各种计算任务^[7]。在众多异构计算架构中,CPU-GPU 异构架构因其出色的性价比而成为高性能计算集群的主要选择,然而找到一种能够在合理时间内管理这种计算环境的高效算法仍然极具挑战。

Topcuoglu 等^[8]研究了两种任务调度算法。其中异构最早完成时间算法(HEFT)通过选择优先级最高的任务并将其分配给能够最小化任务完成时间的处理器来确定整个任务集的调度顺序。关键路径算法(CPOP)基于关键路径分析确定任务优先级,该方法显示出巨大的潜力,但由于其巨大的搜索空间导致其往往无法始终寻找到最小完工时间。在混合任务调度方法中,Akbari 等^[9]结合布谷鸟优化算法和遗传算法创建了 HACG-TS 算法,该方法通过寻找任务的最短完成时间和提高资源利用率实现了多目标的任务调度,并在一定程度上缓解了遗传算法易陷入局部优化的问题。王晴等^[10]发现遗传算法收敛的速度较慢,开展了混合粒子群的遗传优化算法,并用于云计算平台的任务调查研究;刘露等^[11]将粒子群优化(PSO)的核心机制整合到遗传算法(GA)的关键遗传算子

1. 中国航空工业集团公司西安航空计算技术研究所
陕西西安 710000

操作中,有效提高了混合算法的全局优化性能和收敛效率。

实验结果表明,该混合算法的收敛性能优于传统的 GA 或 PSO 算法。从上述研究工作中可以看出,目前异构计算平台上基于进化算法的任务调度方法仍然存在收敛速度慢和易陷入局部最优值问题。因此本文结合实际场景中的任务依赖关系和硬件异构特性,以最大完工时间最小化为总体目标,对用于异构任务调度的遗传算法进行改进研究。

2 问题定义

在异构计算环境中,静态调度的应用程序图通常用有向无环图(DAG)表示为 $G = (T, E)$ 。其中, T 表示任务集; E 表示任务之间依赖关系的边集。顶点 T_i 代表任务集 $T = \{t_1, t_2, \dots, t_i\}$ 中的某个任务,而 $e(i)$ 表示任务 T_i 和 T_j 之间的一条边,带权重的边分别表示如果两个任务被调度到不同处理器上时它们之间的依赖关系和通信成本。在有向无环图中,如果存在一条从任务 T_i 到任务 T_j 的边,即 T_i 被视为 T_j 的前驱任务,而 T_j 是 T_i 的后继任务。

图 1 展示了一个由 8 个节点组成的有向无环图模型示例,在这个 DAG 模型中,每个节点代表一个任务,每条边代表任务之间的依赖关系和通信延迟。任务间的通信时间集合用 $C = \{\dots c_{i,j} \dots\}$ 表示, $c_{i,j}$ 表示两个相互依赖的任务间的通信开销。每个任务在处理器上的执行时间集合则被表示 $W = \{\dots W_{t_i,p_j} \dots\}$, W_{t_i,p_j} 表示任务 T_i 在处理器 p_j 上的计算开销。

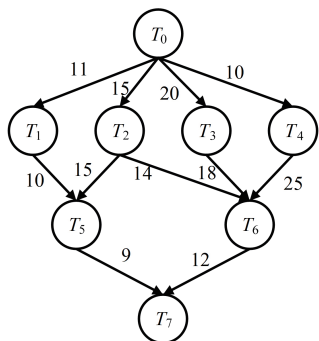


图 1 有向无环图示例

本文将异构计算环境下的任务调度问题描述为:将 n 个存在依赖关系的任务分配到 m 个异构处理器上,目标是在保证任务拓扑顺序的前提下,求得任务调度序列的最优解,保证按照该序列执行完整个任务集的完成时间最短。本文的任务调度目标函数用公式表示为:

$$\min \{\text{makespan}\} = \min \{\max f(t_i)\} \quad (1)$$

式中: makespan 为整个任务集的完成时间,也称为任务集的调度长度。任务 t_i 的完成时间 $f(t_i)$ 由公式计算为:

$$f(t_i) = s(t_i) + e(t_i, p_j) \quad (2)$$

式中: $s(t_i)$ 是任务 t_i 的所有前驱任务的完成时间(包含任务 t_i 的所有前驱节点); $e(t_i, p_j)$ 是任务 t_i 在分配的处理器 p_j 上的执行时间; p_j 是调度决策中分配给任务 t_i 的处理器。

3 基于遗传算法的任务调度策略

任务调度算法是本文的研究重点,针对遗传算法易出现早熟收敛、局部最优等问题,本文基于混合启发式遗传优化算法的任务调度模型从编码方式、遗传算子设计、局部优化三方面对遗传算法进行改进。

3.1 编码

在遗传算法中,编码是将问题的解映射为染色体或个体的过程,编码方式的选择直接影响算法的性能和解的质量。

考虑到异构平台调度需同时处理任务执行顺序与资源分配映射两个核心问题,传统单染色体编码难以兼顾二者的约束与协同。因此,本文采用双染色体编码策略,第一条染色体为任务序列染色体,采用排列编码,长度等于任务总数,每个基因位对应任务 ID,基因顺序表示任务在异构资源上的执行顺序,任务的执行顺序满足任务依赖关系,即父任务必须位于子任务之前。第二条染色体表示资源分配染色体,该染色体采用整数编码,长度与任务序列染色体一致,每个基因位对应任务分配的处理器 ID,处理器 ID 与异构资源池中的处理器资源一一映射。图 2 为随机任务图所对应的双染色体编码的示意图。

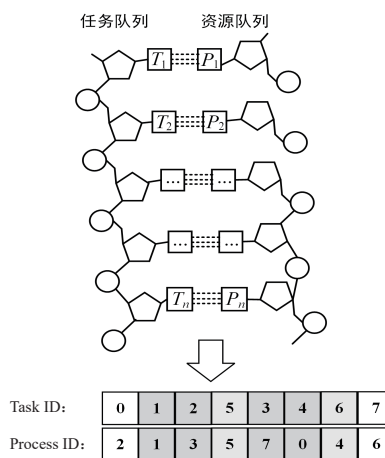


图 2 双染色体的编码示意图

双染色体编码既保证了任务执行的依赖关系,又实现了任务与资源的灵活匹配,为后续遗传操作提供了清晰的调整空间。

3.2 初始种群构建与适应度函数设计

初始种群的设置对于遗传算法的收敛效率和解的质量具有重要影响。传统随机初始化易生成大量不可行解或低适应度解,因此本文结合启发式算法 HEFT 与随机扰动生成初始种群,步骤如下:

(1) 采用异构调度中经典的 HEFT 算法,计算任务的最长路径长度与平均执行时间,获得任务的优先级排序;

(2) 按照任务的优先级顺序开始进行任务调度,对于当前任务,考虑该资源上已有任务队列的结束时间、该任务

在该资源上的具体计算时间、该任务与其父任务之间的通信延迟（如果父任务在不同资源上执行）等因素，计算任务最早完成时间 EFT，选择能使当前任务获得最早完成时间的那个资源进行分配；

(3) 多次重复步骤 (1~2)，直到所有任务都被调度完毕，得到完整的调度方案，生成 10%~20% 的初始解；

(4) 从生成的 HEFT 核心解中随机选择一个解作为进行扰动的“源解”；

(5) 对选择的“源解”进行随机调整（交换任务序列中的两个非依赖任务位置、将某任务分配给相邻资源），得到一个不同于源解的新调度方案；

(6) 多次重复步骤 (4~5)，生成剩余 80%~90% 的初始解；

(7) 将步骤 (3) 生成的所有 HEFT 核心解和步骤 (6) 生成的所有随机扰动解合并在一起，形成遗传算法的初始种群。

获得初始种群后，需要计算种群的适应度函数。适应度函数是遗传算法中优化问题的目标函数，它的作用是对种群里的所有个体进行评价，对各种遗传算子起到控制作用。为降低算法的整体复杂度，其在设计时应尽可能简洁。本文中，适应度函数为整个任务集的完成时间。

3.3 遗传算子

3.3.1 选择算子

选择算子通过筛选种群中的优质个体实现进化导向，其目的是将优良特性传给后代，即优秀个体可直接遗传至下一代，或作为父本生成新个体。选择算子的算法有很多，本文采用了锦标赛选择与精英保留策略。每次随机选取 k 个个体，将适应度高的个体保留至下一代，维持种群的多样性；然后将当前种群中适应度前 5% 的个体直接复制到下一代，避免优秀解丢失。

3.3.2 交叉算子

交叉与变异算子在遗传算法中扮演着关键角色，其核心作用是为了防止算法结果过快收敛，避免无法找到全局最优解。具体来说，当通过选择算子确定了下一代的种群个体后，并不会立即让这些个体进入下一代进化过程。相反，为了确保能够实现全局最优解，会以当前这一代种群个体为基础，按照初始设定的交叉概率和变异概率进行优化操作。只有当所有个体都完成了交叉和变异这两项运算后，新的种群个体才正式形成，并开启新一轮的进化循环。

考虑到本文所采用的随机任务图对资源类型的选择同时受任务计算需求和通信开销影响，任务与资源间呈现弱耦合特征，因此针对本文所用的双链染色体结构，两条染色体间采用相互独立的交叉和变异策略。即两条染色体可采用同步交叉（变异）模式，即同时按各自特定方式执行操作；也可采用异步交叉（变异）模式，即一条染色体执行操作时另一条保持不变。这种灵活的变异机制有效保障了基因变异的丰

富性和优秀基因的遗传。

基于上述的交叉变异原则，首先分别设计两条染色体满足约束的交叉方式。对于任务序列染色体，本文采用顺序交叉方法（OX），交叉过程如下：

(1) 在双亲染色体的相同位置随机选定交叉的起止位置（如图 3 箭头所示）；

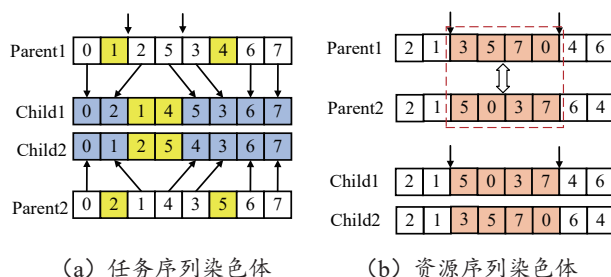


图 3 双染色体的交叉过程示意图

(2) 构建子代 1，首先将父代 2 中被选基因按原序复制至子代 1 对应位置；然后将父代 1 的其他基因按顺序填入子代 1 中的剩余位置；

(3) 构建子代 2，即首先将父代 1 中被选基因按原序复制至子代 2 对应位置；然后将父代 2 其他基因按顺序填入子代 2 中的剩余位置。

对于资源分配染色体，采用随机双点交叉方法，在双亲染色体上随机选择两个交叉点，交换两点间的任务分配基因片段。图 3 展示了完整的染色体交叉过程。

3.3.3 变异算子

变异是遗传算法中模拟基因突变的随机操作，通过修改个体基因序列片段引入新遗传信息，其核心作用是维持种群多样性、避免早熟收敛，并探索搜索空间的未访问区域。本文采用自适应的变异概率，依据个体适应度动态调整变异概率，即适应度高的个体赋予较低变异概率以保留优良特性；适应度低的个体则提高变异概率以引入新基因。

对于图 1 中的 DAG 应用示例模型，对任务序列染色体 {0 1 2 5 3 4 6 7} 应用插入变异，假如选择子任务节点 5 作为一个变异点，根据子任务调度优先队列中的依赖关系，其最后一个前驱子任务节点 1 和 2 决定了插入起点，其第 1 个后继子任务节点 7 决定了其插入的终点，因此子任务 5 的合法的移动范围为 2~7 之间，若将 6 作为变异基因的插入位置，则生成的有效新序列为 {0 1 2 3 4 6 5 7}，如图 4 (a) 所示。

双染色体的变异同样遵循独立设计的原则，为避免对资源分配染色体引入较大的变化，文中采用交换变异的方法，随机选择两个任务，将其资源类型进行交换，如图 4 (b) 所示。

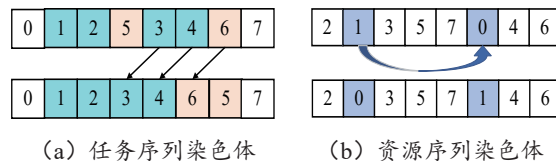


图 4 双染色体的变异过程示意图

3.4 局部搜索

在遗传算法中,局部搜索是一种通过“微调”个体基因以挖掘局部最优解的启发式策略,其核心目标是增强算法的利用能力,即在遗传算法通过选择、交叉、变异实现全局探索的基础上,对种群中的潜在优秀个体进行邻域内的精细搜索,快速收敛至更优的局部解,从而平衡遗传算法“广域探索”与“局部求精”的矛盾。本文中在每一代进化后,对适应度前 10% 的个体进行邻域搜索,调整任务顺序与资源分配,寻找更优解。具体地,每次将满足依赖关系的两个相邻任务的位置进行交换,计算 makespan 变化,若缩短则保留;或者每次仅修改一个任务的分配节点,将某任务从当前资源转移到其他可用资源,尝试将其分配到所有其他允许的节点上,保持其他任务的分配顺序不变,计算适应度变化,若提升则保留。

4 实验结果分析

为了验证混合启发式遗传优化算法(HGA)在异构计算平台任务调度中的性能,实验选取随机生成的有向无环图任务集(分别包含 50、100、150、200、250 个任务)作为测试用例,在由 CPU-GPU 组成的异构计算平台上对算法进行测试。选取最早完成时间算法(HEFT)和蚁群优化(ACO)作为对比算法,采用 makespan(最大完成时间)及收敛速度作为评价指标进行对比分析。

图 5 展示了不同任务规模下各算法的平均调度长度(即最小完成时间)。由图 5 可知,相较于 HEFT 和 ACO, HGA 的平均执行时间更短,尤其是任务数量增多时,这种差异越发明, HGA 的性能明显优于 HEFT 和 ACO。该结果充分说明, HGA 具有更快的收敛速度和更强的寻优能力,算法的整体性能更为优异。

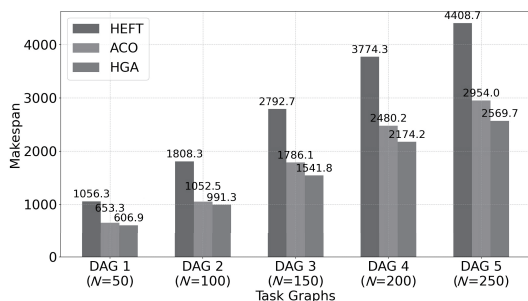


图 5 不同算法的最优调度长度对比

5 结语

本文针对异构计算平台任务调度中传统遗传算法易陷入早熟收敛、局部最优,且收敛效率不佳等问题,提出了一种改进遗传算法,通过双链染色体编码优化、自适应交叉变异概率调整及融合邻域搜索的局部搜索策略,算法的全局搜索能力与收敛性能均得到了提升。实验结果表明,与异构最早完成时间算法及蚁群优化算法相比,改进后的算法显著降低

了任务的调度长度,加快了收敛速度,实现了异构环境下任务调度的效率有效提升。本研究验证了在 CPU-GPU 异构任务调度场景下混合启发式遗传优化算法的有效性,为高性能计算、云计算等场景下的任务优化分配提供了可行方法。

参考文献:

- [1] LAZAREV A A, ARKHIPOV D I. Estimation of the absolute error and polynomial solvability for a classical NP-Hard scheduling problem[J]. Doklady mathematics, 2018,97 (3): 262-265.
- [2] RAVI P S, TUNCEL L. Approximation ratio of LD algorithm for multi-processor scheduling and the coffman-sethi conjecture[J].Information processing letters, 2020,159/160:105959.
- [3] 程小辉,童辉辉,康燕萍.基于麻雀搜索算法的异构多核处理器任务调度[J].计算机应用与软件,2023,40(4):211-216.
- [4] 罗乐,王春华,张多利,等.一种多核系统改进型列表调度算法[J].电子科技,2020,33(6):52-57.
- [5] 马学森,谈杰,陈树友,等.云计算多目标任务调度的优化粒子群算法研究[J].电子测量与仪器学报,2020,34(8): 133-143.
- [6] KHAN M S A,SANTHOSH R. Task scheduling in cloud computing using hybrid optimization algorithm[J]. Soft computing, 2022, 26(23): 13069-13079.
- [7] SULAIMAN M, HALIM Z, LEBBAH M, et al.An evolutionary computing-based efficient hybrid task scheduling approach for heterogeneous computing environment[J/OL]. Journal of grid computing,2021[2025-12-22].https://link.springer.com/article/10.1007/s10723-021-09552-4.
- [8] TOPCUOGLU H, HARIRI S, WU M Y. Performance-effective and low-complexity task scheduling for heterogeneous computing[J]. IEEE transactions on parallel and distributed systems, 2002, 13(3):260-274.
- [9] AKBARI M. Hybrid approach based on cuckoo optimization algorithm and genetic algorithm for task scheduling[J]. Evolutionary intelligence, 2021,14(4):1931-1947.
- [10] 王晴,付学良,李宏慧,等.基于云计算任务调度的遗传粒子群优化算法[J].计算机科学与应用,2018,8(9): 1334-1340.
- [11] 刘露,陈赞,刘世劼,等.一种新型粒子群改进遗传算法[J].微型机与应用,2017,36(23): 17-20.

【作者简介】

李明(1994—),女,河南驻马店人,博士研究生,工程师,研究方向:人工智能、进化计算以及异构调度等。

(收稿日期:2025-08-21 修回日期:2026-01-06)