

交换突变策略改进萤火虫算法的异构并行机调度

罗冬梅, 陈玲清, 张瑀鑫, 黄兴旺

(集美大学计算机工程学院, 福建 厦门 361021)

[摘要] 序列相关设置时间的异构并行机调度问题是个 NP (non-deterministic polynomial) 问题, 在高纬度情况下难以求解。选取任务完工时间为优化目标建立数学模型, 提出一种基于交换突变策略改进的萤火虫算法, 并应用该算法进行求解, 以期在可接受的时间内提供近似最优解的可行方案。实验结果表明, 所提出的算法在处理异构并行机调度问题时具有较突出的全局搜索优势, 收敛速度较快, 搜索精度高, 测算了 36 个算例, 其中的 28 个取得最优平均解, 并且较萤火虫算法、模拟退火算法和遗传算法分别减少了 11.12%、7.36% 和 1.43% 的平均任务完工时间。

[关键词] 异构并行机调度; 萤火虫算法; 交换突变; 任务完工时间; 序相关设置时间

[中图分类号] TP 18

Improved Firefly Algorithm Based on Mutation Strategy for Unrelated Parallel Machine Scheduling Problem

LUO Dongmei, CHEN Lingqing, ZHANG Yuxin, HUANG Xingwang

(School of Computer Engineering, Jimei University, Xiamen 361021, China)

Abstract: The scheduling problem of unrelated parallel machines considering the sequence-dependent setup time is an NP(non-deterministic polynomial) problem, which is difficult to solve in high-dimensional cases. In this paper, the maximum completion time is selected as the optimization objective to establish a mathematical model, and an improved firefly algorithm based on swap mutation strategy is proposed, and the algorithm is applied to solve the problem, in order to provide a feasible solution of approximate optimal solution in an acceptable time. The experimental results show that the proposed algorithm has outstanding global search advantages in dealing with unrelated parallel machine scheduling problem with sequence-dependent setup times, with fast convergence speed and high search accuracy. The optimal average solution is obtained in 28 of the 36 cases, and the average makespan is reduced by 11.12%, 7.36% and 1.43% respectively compared with firefly algorithm (FA), simulated annealing (SA) and genetic algorithm (GA).

Keywords: unrelated parallel machines scheduling; firefly algorithm; swap mutation; makespan; sequence-dependent setup times

[收稿日期] 2021-10-28

[基金项目] 国家自然科学基金项目(62006096); 福建省自然科学基金项目(2020J01699, 2020J05146); 福建省中青年教师教育科研项目(JAT190320)

[作者简介] 罗冬梅(1977—), 女, 讲师, 主要从事计算网络方向研究。通信作者: 黄兴旺(1990—), 男, 博士, 副教授, 主要从事计算智能方向研究。E-mail: huangxw@jmu.edu.cn

0 引言

经典的异构并行机调度问题 (parallel machine scheduling problem, PMSP)^[1] 广泛存在于科学和工业生产中, 包括印刷电路板制造^[2]、半导体晶圆制造^[3]的切割操作等应用领域。在云计算^[4]、边缘计算^[5]中通常还会涉及到作业排序的等待时间、作业数据传输时间等设置时间, 并且由于设备的异构性, 相当部分类型作业还会有序相关设置时间的异构并行机调度问题 (the unrelated parallel machines scheduling problem, UPMSP)^[1]。

在 UPMSP 问题中, 优化目标通常是最小化任务完工时间。大多数问题需要使用两台或多台机器并行调度, 在某些情况下, 设置时间和处理时间同等重要, 因而需要同等考虑。Paula 等^[6]第一次将可变邻域搜索算法 (variable neighborhood search, VNS) 用于求解高维的序相关设置时间的 UPMSP 问题。Vallada 等^[7]提出一种基于局部搜索增强交叉算子的遗传算法 (genetic algorithm, GA) 来求解 UPMSP 问题, 该研究也考虑了机器和作业之间的序相关设置时间影响。Behnamian 等^[8]提出一个结合了蚁群 (ant colony optimization, ACO)、模拟退火 (simulated annealing, SA) 和 VNS 算法的混合元启发式算法, 求解同样考虑了序相关设置时间的 UPMSP 问题, 使得任务完成时间最小化。Arnaout 等的研究^[8-9]使用增强的 ACO 算法在机器数多达 10 且作业数多达 120 的用例中求解序相关设置时间的 UPMSP 问题。Ezugwu 等^[3]实现了一种改进的樽海鞘群算法 (salp swarm algorithm, SSA), 得到与序列相关的 UPMSP 算法的新的复杂度结果。牛群等^[10]基于改进的克隆选择算法实现了含调整时间的并行机调度, 其性能较遗传算法和基本克隆选择算法都有所提高。但是上述算法在高维情况下, 搜索性能有所下降。

标准萤火虫算法 (firefly algorithm, FA)^[11]具有超参数少、易实现等优点, 在连续域数值优化问题中表现出了高效的性能。由于序相关设置时间 UPMSP 问题具有非确定多项式难的性质, 求解难度大, 而 FA 采用的吸引度因子存在快速减小情况, 又容易面临提前终止搜索、达到局部最优的问题。为此, 本文提出了一种基于交换突变策略改进的萤火虫算法 (improved firefly algorithm based on mutation strategy, IFAMS) 用于求解具有序相关设置时间的 UPMSP 问题。

1 数学建模

1.1 问题描述

本文所改进的求偶学习的萤火虫算法求解序相关设置时间的 UPMSP 问题描述如下:

- 该调度问题假定在时间零点有 N 个可用的作业待分配到 M 台异构的机器上处理, 机器之间相互独立。
- 不存在抢占执行作业情况。
- 数据集中, 由 $P_{j,k}$ 表示的机器 k 处理分配到的作业 j 所需的时间, 和由 $S_{i,j,k}$ 给出的机器 k 上的作业 i 之后处理作业 j 的序相关设置时间, 都是先验和确定的。其中, $i, j = \{1, 2, \dots, N\}$, $k = \{1, 2, \dots, M\}$, 通常 $S_{i,j,k} \neq S_{j,i,k}$ 。
- 优化目标为最小化任务完工时间 C_{\max} , 因此该模型可以用 $(P_{j,k} \mid S_{i,j,k} \mid C_{\max})$ 表示。

1.2 模型构建

本文采用基于混合整数规划 (mixed integer programming, MIP) 拓展的模型^[1,3,7]来表示 $(P_{j,k} \mid S_{i,j,k} \mid C_{\max})$ 模型, 方便最小化任务完工时间 C_{\max} 的求解。为了描述该问题的数学模型, 本文给出了部分符号定义: C_j —机器上最后一个作业 j 的完成时间; $x_{i,j,k}$ —在机器 k 上处理完作业 i 后立即处理作业 j 时为 1, 否则为 0; $x_{0,j,k}$ —机器 k 首先处理作业 j 为 1, 否则为 0; V —一个充分大的正数; $A_{P_{i,j,k}}$ —校正后的机器处理时间矩阵。

本文建立了 $(P_{j,k} \mid S_{i,j,k} \mid C_{\max})$ 的数学模型为:

$$\text{Minimize } C_{\max} \quad (1)$$

约束条件有:

- 1) $\sum_k^M \sum_{i=0}^N x_{i,j,k} = 1, \forall j = 1, \dots, N;$
- 2) $\sum_{j \neq i}^N x_{0,j,k} = 1, \forall k = 1, \dots, M;$
- 3) $\sum_{i=0}^N x_{i,j,k} - \sum_{j \neq h}^N x_{h,j,k} = 0, \forall h = 1, \dots, N;$
- 4) $C_j - [C_i + \sum_{k=1}^M x_{i,j,k} (S_{i,j,k} + P_{j,k}) + V(\sum_{k=1}^M x_{i,j,k} - 1)] \geq 0, \forall i = 0, \dots, N;$
- 5) $C_j \leq C_{\max}, \forall j = 1, \dots, N;$
- 6) $C_0 = 0;$
- 7) $C_j \geq 0, \forall j = 0, \dots, N;$
- 8) $x_{i,j,k} \in \{0, 1\}, \forall i = 0, \dots, N, \forall j = 0, \dots, N, \forall k = 1, \dots, M。$

其中: 公式(1)的目标是最小化任务完工时间; 约束条件1) 确保作业只会被执行一次; 约束条件2) 表示每台机器最先被处理的作业数为1; 约束条件3) 确保每个作业最先被处理或者构成其他作业的紧后处理作业; 约束条件4) 用于计算各个作业的完工时间; 约束条件5) 将 C_{\max} 定义为必须大于任何其他作业完成时间的变量; 约束条件6) 确保虚拟作业0的完成时间为0; 约束条件7) 确保作业完成时间非负; 约束条件8) 定义了决策变量 x 的取值范围。

2 基于改进萤火虫算法的 UPMSP

2.1 标准萤火虫算法

萤火虫的荧光亮度(相互吸引度)可表示为: $\beta_{i,j}(r_{i,j}) = \beta_0 e^{-\gamma r_{i,j}^2}$ 。其中: $r_{i,j}$ 是两只萤火虫 x_i 和 x_j 之间的欧几里德距离; β_0 表示初始吸引度因子, 通常取值为1; 参数 γ 表示光吸收系数的固定值, 一般也取值为1。对于同一搜索空间随机选取的两只萤火虫 x_i 和 x_j , 它们之间的距离

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{d=1}^D (x_{i,d} - x_{j,d})^2}。$$

其中: D 代表问题的维数; d 表示处于萤火虫位置向量的维数。

搜索过程中, 萤火虫 x_j 受更亮的萤火虫 x_i 吸引, 向 x_i 移动, 其位置更新方式为: $x_j(t+1) = x_j(t) + \beta_{i,j}(x_i(t) - x_j(t)) + \alpha \epsilon$ 。其中: t 表示算法的迭代次数; α 是步长参数, 通常取值为 $[0, 1]$ 的随机数; ϵ 是 $[-0.5, 0.5]$ 之间的随机数。

在 FA 的寻优过程中, 所有萤火虫会向较亮的萤火虫移动, 最终萤火虫个体将集结在最亮的萤火虫周围, 完成寻优过程。

2.2 改进 FA 算法

仔细分析 FA 算法中的吸引度因子 β 随萤火虫距离 r 的变化可知, 当这两个萤火虫之间的距离 r 较大时, β 随 r 增加快速下降至近0, 萤火虫间的吸引度快速下降, 个体之间无法充分地进行信息交互, 此时 FA 算法面临移动提前终止、萤火虫种群进化出现停滞的情况。一旦陷入局部最优, 由于此时萤火虫不具备变异特性, 将无法跳出。

本文在 FA 框架^[11]的基础上, 提出一种改进的萤火虫算法。该算法通过引入交换突变策略, 保证算法在距离 r 较远时萤火虫个体间仍有机会进行信息交互, 从而跳出局部最优, 提高全局搜索能力。该算法既保持了 FA 原有的简单结构, 又能提高寻优精度。

选择种群中适应度函数值最小即最优的萤火虫 cbest , 应用交换突变策略进行操作, 从而引入随

机漫游, 增加种群多样性, 避免过早收敛。该操作过程的伪代码如下所示:

```

Input: cbest
    从整数 1 到  $D$  中无放回随机均匀抽取 2 个值 ( $i_1$  和  $i_2$ ) 作为突变位置
    newbest = cbest;
    /* 用 cbest 第  $i_2$  维和第  $i_1$  维的值分别对 newbest 的第  $i_1$  维和第  $i_2$  维进行交换 */
    newbest ( [  $i_1$   $i_2$  ] ) = cbest ( [  $i_2$   $i_1$  ] );
    if  $f(\text{newbest}) < f(\text{cbest})$ 
        cbest = newbest
End
Output: cbest

```

其中, cbest 表示当前最优解, D 代表问题的维数, $f(\cdot)$ 表示适应度函数。

基于上述操作, 改进的萤火虫算法 IFAMS 的完整搜索过程如下:

步骤 1) 采用均匀分布随机生成 N_p 只雄性萤火虫作为初始种群 $\{X_i | i = 1, 2, \dots, N_p\}$;

步骤 2) 计算每只雄性萤火虫的发光亮度;

步骤 3) 如果雄性萤火虫 X_j 的亮度低于雄性 X_i 的亮度, 则采用 FA 原来的移动策略更新雄性萤火虫位置, 并更新最优解;

步骤 4) 执行交换突变操作;

步骤 5) 判断算法是否达到终止迭代的条件, 若达到, 则停止迭代, 算法终止, 否则返回并继续执行步骤 3)。

2.3 IFAMS 应用于含序相关设置时间的 UPMSP

2.3.1 个体编码表示

设计适当的个体编码方案, 能够增加 IFAMS 算法处理候选可行方案与萤火虫个体之间的映射效率。基于文献 [1], 本研究设计了两阶段个体编码表示方案处理该问题。

1) 机器分配编码

第一阶段是机器分配过程, 描述了将 N 个任务分配给 M 台异构并行机, 以使所有机器的最大完成时间 C_{\max} 最小的可行解, 可用维数等于作业数的整数向量 x 表示。以 10 个作业、3 个并行机的作业分配为例, 若序列 $S_1 = [3, 1, 3, 2, 1, 2, 2, 3, 1, 1]$, 则表示机器 1 分配到的作业序号为 $\{2, 5, 9, 10\}$, 机器 2 分配到的作业序号为 $\{4, 6, 7\}$, 机器 3 分配到的作业序号为 $\{1, 3, 8\}$ 。因此, 这也要求 IFAMS 算法在迭代寻优过程中需要采用 floor(\cdot) 函数将 IFAMS 中萤火虫的位置信息进行整数离散化。

2) 序列调度编码

第二阶段则是确定每台机器上的作业序列。该序列可以矩阵的形式表示, 矩阵的长度与机器分配向量的长度相同。因此, 作业序列 S_2 可以表示为 $M \times N$ 矩阵, 表明每台机器上作业的执行顺序。以上文的例子为例, 有

$$S_2 = \begin{bmatrix} 9 & 5 & 10 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 7 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

表明机器 1 上的作业执行顺序为作业 9、作业 5、作业 10 和作业 2, 机器 2 上的作业执行顺序为作业 4、作业 7 和作业 6, 机器 3 上的作业执行顺序为作业 3、作业 1 和作业 8。作业 2 后面的零值表示作业 2 是机器 1 最后处理的作业。类似的, 作业 6 和作业 8 分别是机器 2 和机器 3 最后处理的作业。根据 S_2 和 $x_{i,j,k}$ 的定义, 可以很容易地得到 $x_{i,j,k}$ 相应的值, 从而可以根据约束条件 4) 计算任务完工时间。具体而言, 设置一个充分大的正数 ($V = \infty$), 当作业 j 在作业 i 之后执行时, $\sum_{k=1}^M x_{i,j,k} = 1$, 因此

此 $V(\sum_{k=1}^M x_{i,j,k} - 1) = 0$, 此时有 $C_j = C_i + S_{i,j,k} + P_{j,k}$ 。

为方便求解任务完工时间, 本文以校正后的处理时间矩阵 A_{p_k} 替代两个独立阶段的时间 (机器处理时间 P 和序相关设置时间 T) 进行处理。 A_{p_k} 可以表示为上述两个阶段耗时的线性组合, 数学式为: $A_p = \theta_1 \times T + \theta_2 \times P$ 。

为简单起见, 通常可取 $\theta_1 = \theta_2 = 1$, 因而有: $\forall j = 1, 2, \dots, N, \forall k = 1, 2, \dots, M; A_{p_{i,j,k}} = S_{i,j,k} + P_{j,k}, \forall i = 1, 2, \dots, N$ 。代入式 (1) 中, 可得序相关设置时间的 UPMSP 问题的优化目标函数 (即适应度函数) 为:

$$\text{Minimize } C_{\max} = \max_{1 \leq j \leq N} \{C_j\} = \max_{\substack{k=1, \dots, M \\ i=1, \dots, N}} \sum_{j=1}^N A_{p_{i,j,k}}$$

2.3.2 算法应用过程

在初始化阶段, IFAMS 算法首先将 N 个计划内的作业随机分配给 M 台可用的处理机器, 生成由 N_p 只雄性萤火虫个体构成的初始雄性萤火虫种群 (用矩阵 X 表示)。每只雄性萤火虫个体对应于 $M \times N$ 矩阵中编码后选定的特定机器上的作业序列。外部档案 A 中的雌性萤火虫种群可借由雄性萤火虫种群初始化完成。

具体应用步骤如下:

- 步骤 1) 初始化。根据序相关设置时间 UPMSP 的数学模型和给定的机器处理时间、序相关设置时间的范围, 初始化改进求偶学习萤火虫算法的各项参数。
- 步骤 2) 种群进化。IFAMS 的种群进化过程如 2.2 节所述。
- 步骤 3) 当进化满足停止条件时, 输出最优雌性萤火虫的相关信息作为近似最优调度方案。

3 仿真实验

为了验证 IFAMS 算法在求解序相关设置时间的 UPMSP 问题上的有效性, 本文采用大量的测试用例进行了验证, 所有实验都是通过 Matlab 2020a 编程实现, 在 8.0 G RAM 3.6 GHz CPU 的 Windows 环境下运行。基于各个算法迭代生成过程的次数, 将 IFAMS 算法在 ($P_{j,k} \mid S_{i,j,k} \mid C_{\max}$) 问题中的性能表现与文献中现有的一些元启发式算法进行了比较, 包括 GA、SA 及标准的 FA 算法。为了避免测试结果的随机性, 每种算法重复测试 15 次。考虑到需要在可接受时间范围内获取可行解, 因而选取最大迭代次数为 1000 次, $N_p = 50$, 其他所有参数直接来自文献 [12]。考虑到求解任务调度方案的时间相较于执行时间, 比值较小, 故而本文主要测试算法的求解精度, 通过均值 Mean 和方差 SD 的大小来衡量。

3.1 测试用例

文献 [13] 的测试数据集已应用于许多相关研究中^[3,7,10,12-14]。该数据集包含两个类别, 即处理时间和设置时间, 其数据都由离散的均匀分布 $U[50, 100]$ 随机生成。

本研究生成的数据集包含 N 个作业和 M 个机器, 有 6 种情况。每种情况中的机器数量分别为 2、4、6、8、10、12, 作业数量范围则分别在 20、40、60、80、100、120 之间。

3.2 实验结果和性能比较

根据测试用例, 本研究构造得到 36 组算例, 这些算例分别在各个算法中运行, 所得结果如表 2 所示。其中, 每种算例情况的最优值以黑色加粗标注, 结果保留 2 位小数。由表 2 的数据分析可知:

- 1) 从精度角度而言, IFAMS 得到最优 Mean 值的次数远远多于 FA、GA 和 SA, 说明本文提出的 IFAMS 算法非常适用于求解序相关设置时间的异构并行机调度问题, 表明在 FA 框架基础上引入交换突变策略的有效性。
- 2) IFAMS 算法得到的结果与其他 3 种算法相比, 能够在 36 个算例中的 28 个取得最优 Mean 值, 并且随着算例维数的增大, IFAMS 取得最优 Mean 值的频次逐渐增加, 表明 IFAMS 在处理较大规模的寻优问题时其精度方面的性能优势显著。此外, 在 SD 值方面, IFAMS 也较 FA 取得了较大提升, 其寻优稳定性仅次于 SA。

表 2 IFAMS 算法与 4 种对比算法的性能比较
Tab. 2 Performance comparison between IFAMS algorithm and four comparison algorithms

<i>M</i>	<i>N</i>	GA		SA		FA		IFAMS	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
2	20	1206.53	7.24	1200.47	7.74	1234.27	11.97	1208.33	12.06
	40	2477.47	14.83	2534.20	9.62	2616.00	28.78	2480.07	12.79
	60	3692.93	31.04	3859.27	17.04	3920.53	40.40	3695.20	24.26
	80	4987.20	27.57	5248.20	19.79	5272.60	53.94	4962.93	29.05
	100	6148.80	27.11	6515.33	26.41	6578.00	81.13	6098.40	20.47
	120	7456.27	40.98	7955.40	28.00	8035.27	129.70	7427.47	35.02
4	20	577.00	17.85	568.67	5.91	604.20	16.25	562.93	6.44
	40	1199.53	27.64	1230.40	10.85	1310.93	33.55	1167.93	22.46
	60	1804.07	25.40	1919.87	12.45	2024.40	46.94	1790.80	20.26
	80	2420.20	26.03	2597.60	14.81	2727.73	32.33	2386.93	12.03
	100	3072.47	27.36	3302.53	17.20	3407.60	71.59	3029.73	30.29
	120	3717.80	28.94	4010.87	20.92	4133.67	62.03	3647.93	28.21
6	20	393.73	11.34	391.40	4.27	432.40	12.51	394.73	8.92
	40	793.60	18.99	835.53	7.08	905.27	29.29	788.73	17.46
	60	1219.73	22.05	1292.93	13.11	1375.67	30.05	1191.40	17.90
	80	1629.93	37.37	1745.20	9.05	1836.67	41.11	1577.13	23.81
	100	2070.80	26.20	2224.27	20.72	2319.60	38.07	2015.00	13.15
	120	2504.27	29.61	2707.73	16.52	2808.80	48.20	2438.07	23.80
8	20	296.07	6.41	295.40	4.94	340.87	12.40	298.40	8.23
	40	578.40	20.03	615.93	6.93	682.47	34.40	575.53	24.16
	60	922.40	23.26	980.00	9.04	1061.40	26.73	891.73	15.80
	80	1244.27	23.97	1337.27	9.69	1403.13	37.56	1213.47	30.22
	100	1556.47	24.28	1683.00	10.49	1767.53	37.85	1522.13	32.95
	120	1886.73	24.61	2054.93	10.79	122.27	32.87	1826.67	22.84
10	20	196.53	8.68	203.13	2.77	271.53	33.83	199.13	29.79
	40	455.73	15.75	485.13	6.08	571.00	46.91	447.27	19.37
	60	711.53	20.64	776.20	8.33	872.33	45.16	711.67	23.16
	80	1003.40	33.43	1073.87	9.86	1195.20	79.36	967.87	21.23
	100	1253.00	20.43	1363.20	10.44	1437.93	45.01	1208.33	19.43
	120	1523.33	25.36	1668.67	13.00	1735.07	56.69	1471.47	20.39
12	20	189.80	8.19	192.60	2.92	222.87	9.07	188.27	9.98
	40	416.53	13.57	446.80	4.84	503.27	30.66	420.33	14.14
	60	612.07	32.30	658.20	8.66	756.53	46.55	599.60	34.69
	80	831.67	20.43	909.13	8.16	1025.20	89.77	813.33	30.16
	100	1054.87	24.08	1162.47	12.92	1280.33	77.01	1035.73	31.80
	120	1286.40	26.36	1406.53	12.27	1512.73	67.66	1233.87	19.19
平均值 Mean value		1760.9	22.76	1873.7	11.49	1952.9	44.93	1735.8	21.28

图1和图2展示了机器数分别为2和10的情况下,这4个算法在不同作业数的测试用例上的平均收敛曲线。分析图1和图2可知,IFAMS表现出的收敛速度和FA相似,优于GA和SA。

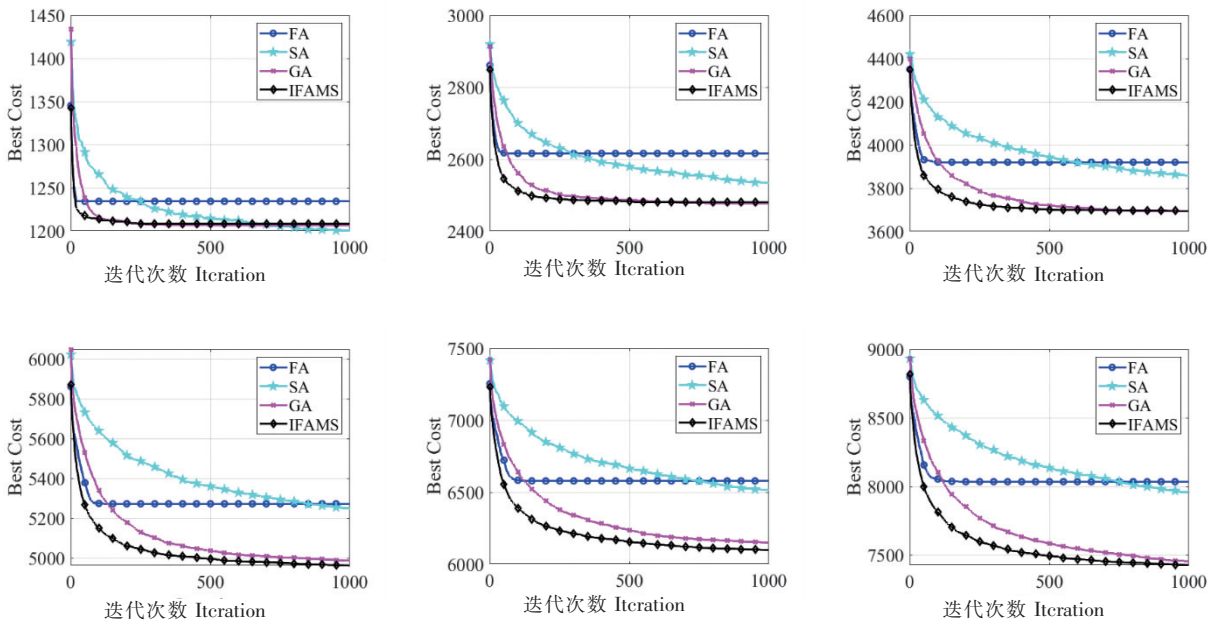


图1 机器数 $M=2$ 时各个算法的平均收敛曲线

Fig.1 Average convergence curve of each algorithm when the number of machines $M=2$

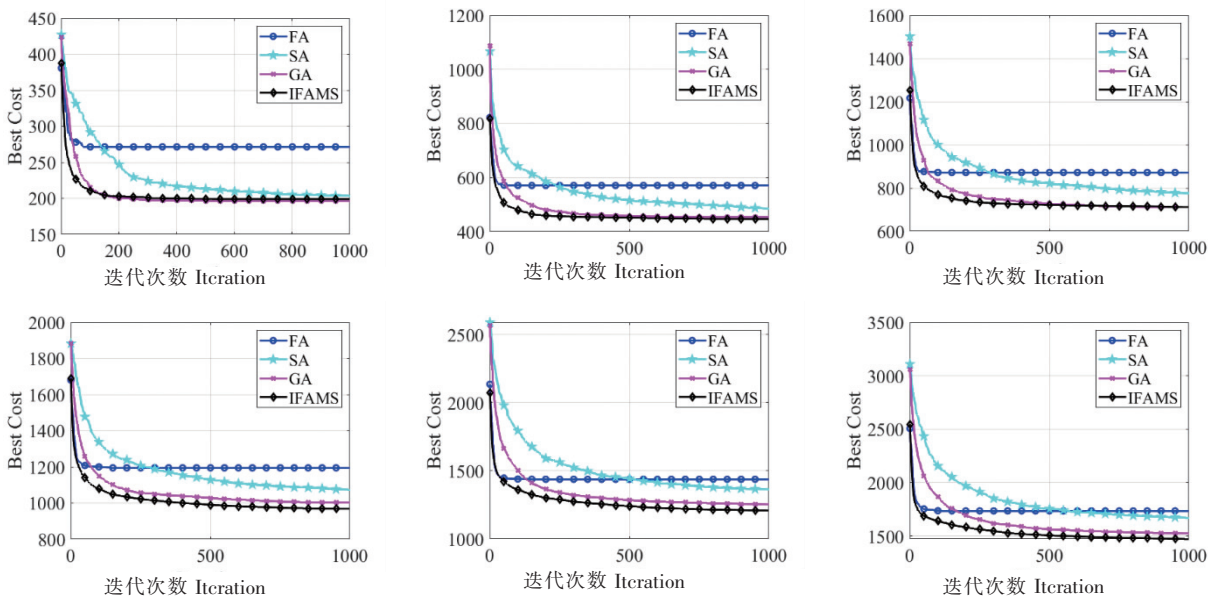


图2 机器数 $M=10$ 时各个算法的平均收敛曲线

Fig.2 Average convergence curve of each algorithm when the number of machines $M=10$

4 结论

本文针对序相关设置时间UPMSP问题,提出了一种改进的求偶学习萤火虫算法IFAMS,以最小化任务完工时间为调度优化目标,求解序相关设置时间UPMSP问题。该算法引入交叉算子,在不引入额外参数的前提下,增强了萤火虫距离较远时的搜索表现,使算法能够更加充分地利用种群社会信息,增强跳出局部最优的能力,提升算法全局搜索性能。通过大量算例测试,结果表明本文所提出的

调度算法在处理序相关设置时间 UPMS P 问题时是有效的,具有较高的收敛速度,同时精度方面相较于 FA 具有较大提升,相较 GA 和 SA 也有明显的优势。

边缘计算等应用场景的调度问题是调度研究的热点,但目前还未有较为统一的模型标准,未来将继续借鉴 UPMS P 模型,在考虑到带宽、不同网络服务提供商资费等更多约束的条件下进行研究,并提出高效算法进行求解。

[参 考 文 献]

- [1] EZUGWU A E. Enhanced symbiotic organisms search algorithm for unrelated parallel machines manufacturing scheduling with setup times [J]. Knowledge Based Systems, 2019, 172: 15-32.
- [2] CHEN C L, CHEN C L. Hybrid metaheuristics for unrelated parallel machine scheduling with sequence-dependent setup times [J]. The International Journal of Advanced Manufacturing Technology, 2008, 43(1): 161-169.
- [3] EZUGWU E S, AKUTSAH F. An improved firefly algorithm for the unrelated parallel machines scheduling problem with sequence-dependent setup times [J]. IEEE Access, 2018, 6(99): 54459-54478.
- [4] 李兵, 付新玥, 高翔, 等. 基于蚁群算法的云计算并行机资源调度研究 [J]. 华中科技大学学报 (自然科学版), 2012, 40(增刊 1): 225-229.
- [5] HUANG X, LI C, CHEN H, et al. Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies [J]. Cluster Computing, 2020, 23(2): 1137-1147.
- [6] PAULA M, RAVETTI M G, MATEUS G R, et al. Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search [J]. IMA Journal of Management Mathematics, 2007, 18(2): 101-115.
- [7] VALLADA E, RUIZ R. A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times [J]. European Journal of Operational Research, 2011, 211(3): 612-622.
- [8] BEHNAMIAN J, ZANDIEH M, GHOMI S M T F. Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm [J]. Expert Systems with Applications, 2009, 36(6): 9637-9644.
- [9] ARNAOUT J P, RABADI G, MUSA R. A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times [J]. Journal of Intelligent Manufacturing, 2010, 21(6): 693-701.
- [10] 牛群, 周台金, 王小海, 等. 基于改进克隆选择算法的含调整时间并行机调度 [J]. 东南大学学报 (自然科学版), 2012, 42(S1): 163-167.
- [11] YANG X S, HE X. Firefly algorithm: recent advances and applications [J]. International Journal of Swarm Intelligence, 2013, 1(1): 36-50.
- [12] AAEA B, AQ C, MAE D. Enhanced salp swarm algorithm based on firefly algorithm for unrelated parallel machine scheduling with setup times [J]. Applied Mathematical Modelling, 2021, 94: 285-305.
- [13] KIM D W, KIM K H, JANG W, et al. Unrelated parallel machine scheduling with setup times using simulated annealing [J]. Robotics and Computer-Integrated Manufacturing, 2002, 18(3/4): 223-231.
- [14] 宋强. 求解异构并行机调度问题的混合多目标教学优化算法 [J]. 控制理论与应用, 2020, 37(10): 2242-2256.

(责任编辑 朱雪莲 英文审校 黄振坤)