

控制与决策

Control and Decision

求解区间数分布式流水线调度的混合离散果蝇优化算法

王凌, 郑洁, 王晶晶

引用本文:

王凌, 郑洁, 王晶晶. 求解区间数分布式流水线调度的混合离散果蝇优化算法[J]. *控制与决策*, 2020, 35(4): 930–936.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.1274>

您可能感兴趣的其他文章

Articles you may be interested in

考虑员工工作关系的单元装配系统构建方法

Seru production system formation considering cooperation of workers

控制与决策. 2020, 35(2): 453–460 <https://doi.org/10.13195/j.kzyjc.2018.0542>

微电子生产过程调度问题基于指标快速预报的分解算法

An indexes fast prediction based decomposition method for scheduling problem in microelectronic production process

控制与决策. 2020, 35(1): 139–146 <https://doi.org/10.13195/j.kzyjc.2018.0422>

自适应协同进化蝙蝠算法

Adaptive cooperation evolutionary bat algorithm

控制与决策. 2019, 34(8): 1626–1634 <https://doi.org/10.13195/j.kzyjc.2018.0029>

装备精确保障任务规划建模与混沌蝙蝠算法求解

Task scheduling modeling and chaotic bat algorithm solving method of equipment efficient support

控制与决策. 2018, 33(9): 1625–1630 <https://doi.org/10.13195/j.kzyjc.2017.0546>

基于滤子混合协同进化算法的无功优化

The research of reactive power flow based on filter hybrid co-evolutionary algorithms

控制与决策. 2017, 32(9): 1701–1706 <https://doi.org/10.13195/j.kzyjc.2016.0878>

一种求解多目标优化问题的进化算法混合框架

A hybrid framework of evolutionary algorithm for solving multi-objective optimization problems

控制与决策. 2017, 32(10): 1729–1738 <https://doi.org/10.13195/j.kzyjc.2016.0937>

基于网格化拉马克学习机制的差分进化算法

Differential evolution algorithm based on gridded Lamarekian learning

控制与决策. 2015(6): 1085–1091 <https://doi.org/10.13195/j.kzyjc.2014.0325>

改进型细菌觅食算法及多货叉仓库调度应用

Improved bacterial foraging algorithm and application of warehousing scheduling with multi-shuttles

控制与决策. 2015(2): 321–327 <https://doi.org/10.13195/j.kzyjc.2013.1436>

求解区间数分布式流水线调度的混合离散果蝇优化算法

王凌[†], 郑洁, 王晶晶

(清华大学自动化系, 北京 100084)

摘要: 分布式调度是制造系统领域的前沿研究, 而不确定调度问题的研究更具现实意义. 针对不确定分布式置换流水线调度问题, 采用区间数表示工序加工时间, 以最小化区间最大完工时间为目标, 利用问题特性在果蝇优化框架内提出一种混合离散果蝇优化算法. 首先, 通过改进启发式方法和随机方法混合初始化种群; 然后, 基于概率协同多搜索操作执行嗅觉搜索. 为了平衡算法的全局探索与局部开发能力, 设计基于学习机制的双种群协同搜索环节. 为了进一步提升种群性能, 针对优良解设计基于切换机制的双模式局部搜索. 基于大量算例的仿真结果与统计对比, 表明所提出算法能更有效求解区间数分布式流水线调度问题.

关键词: 分布式流水线调度; 区间数; 混合离散果蝇算法; 双种群协同

中图分类号: TP8

文献标志码: A

A hybrid discrete fruit fly optimization algorithm for distributed permutation flowshop scheduling with interval data

WANG Ling[†], ZHENG Jie, WANG Jing-jing

(Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract: Distributed scheduling is the frontier research in the field of manufacturing systems, while the study on the uncertain scheduling problems is of more realistic significance. For the uncertain distributed permutation flowshop scheduling problem, interval processing times are used and the minimization of interval maximum completion time is considered as the objective. To solve the problem, a hybrid discrete fruit fly optimization algorithm is proposed with the framework of the fruit fly optimization by using the characteristics of the problem. Firstly, an improved heuristic and the random method are used in a hybrid way to initialize the population. Then, probabilistic cooperation of multiple search operators is executed as the olfactory search. To balance the exploration and the exploitation capabilities, a learning-based bi-population cooperation scheme is designed. Meanwhile, local intensification search based on a bi-mode switch scheme is applied for the elite solutions for further improvement. Simulation results and statistical comparison with extensive instances show that the proposed algorithm can solve the distributed flowshop scheduling with interval data more effectively.

Keywords: distributed permutation flowshop scheduling; interval data; hybrid discrete fruit fly optimization algorithm; bi-population cooperation

0 引言

在全球化影响下, 大量企业逐渐转向多地域合作, 分布式制造与调度已成为一种趋势. 在企业的生产经营过程中, 往往存在各种各样的不确定因素, 导致原有的调度计划无法正常执行. 因此, 不确定环境下的生产调度问题具有很强的应用价值, 从而备受关注. 置换流水线调度问题是研究最为广泛的经典车间调度问题, 智能求解算法包括禁忌搜索^[1]、人工蜂群算法^[2]、回溯搜索算法^[3-4]等. 分布式置换

流水线调度问题 (distributed permutation flowshop scheduling problem, DPFSP)^[5] 中每个工厂都是一个置换流水车间. DPFSP 的工序加工时间通常为确定数值, 但实际生产中加工时间通常难以精确确定. 本文采用区间数表征工序加工时间的不确定性, 进而扩展产生区间数分布式置换流水线调度问题 (interval DPFSP, IDPFSP), 在工序的加工时间为区间数的情况下研究如何将工件合理分配到多个车间, 再通过合理排序来优化调度指标. 数学上, 该问题属于 NP-hard

收稿日期: 2018-09-18; 修回日期: 2018-10-18.

基金项目: 国家重点研发计划项目 (2016YFB0901900); 国家自然科学基金项目 (61873328); 国家杰出青年科学基金项目 (61525304).

责任编辑: 王伟.

[†]通讯作者. E-mail: wangling@tsinghua.edu.cn.

问题. 问题求解层面, 该问题具有大规模、强耦合、不确定等复杂性, 因此其研究具有重要的学术意义和工程应用价值.

目前, DPFSP 已有不少研究成果. Naderi 等^[5] 针对最小化最大完工时间 C_{\max} (即 makespan), 建立了 6 种混合整数规划模型, 并提出了变邻域下降算法. Gao 等^[6] 提出了一种混合遗传算法, 通过邻域搜索调整工件的工厂分配与工厂内的工件加工顺序. Liu 等^[7] 设计了一种离散仿电磁学算法, 对关键工厂设计了变邻域搜索操作. Gao 等^[8] 针对关键工厂设计了一种禁忌搜索算法, 对完工时间最长与最短的工厂执行交换操作. Wang 等^[9] 采用置换编码, 提出了基于最短完工时间工厂分配规则 (earliest completion factory, ECF) 的有效解码方法, 并设计了一种分布估计算法. Shao 等^[10] 针对分布式无等待置换流水线问题, 提出了一种改进规则产生初始解, 并设计了多种基于不同邻域搜索的贪婪迭代算法. Bargaoui 等^[11] 将化学反应优化算法应用于求解分布式置换流水线车间调度问题, 设计了迭代贪婪策略与交叉机制以改进算法. Wang 等^[12] 提出了一种混合离散布谷鸟算法 (hybrid discrete cuckoo search, HDCS), 根据问题特性设计变邻域搜索和基于关键工厂的局部搜索算子. 对于 IDPFSP, 除了考虑 DPFSP 的复杂性以外, 还需考虑区间数的比较与排序问题. 目前已有若干区间数排序方法^[13-17], 其中常用的方法是基于可能度的比较. Lei 等^[18-20] 将区间数扩展到车间作业调度问题, 分别提出了基于种群的邻域搜索算法、遗传算法和混合蛙跳算法, 并针对区间数的比较引入新的性能指标来评估算法性能. 为了解决序区间资源受限项目调度问题 (resource constrained project scheduling problem, RCPSP), Zheng 等^[21] 提出了一种有效的教学算法.

果蝇优化算法 (fruit fly optimization algorithm, FOA)^[22] 是受果蝇觅食行为启发而提出的一种新颖的群体智能优化算法. 近年来, FOA 已被用于求解诸多领域的工程优化问题, 包括预测、物流、调度、控制、故障诊断、电力系统及结构设计等^[23]. Wang 等^[24] 针对多技能 RCPSP, 提出了一种知识驱动的多目标果蝇优化算法, 设计了基于多操作协同的嗅觉搜索和视觉搜索. 针对资源约束的不相关并行机调度问题, Zheng 等^[25] 提出了一种混合整数线性规划模型, 基于问题特性设计了一种知识驱动的自适应 FOA. 针对旅行商问题, Yin 等^[26] 将 FOA 与微粒群优化相结合, 均衡全局搜索与局部挖掘能力. 尽管

FOA 在诸多调度问题的求解上展现出良好的性能, 但在 DPFSP 方面还缺乏研究. 本文将 DPFSP 扩展为 IDPFSP, 针对问题的特点在 FOA 框架内提出一种混合离散果蝇优化算法 (hybrid discrete FOA, HDFOA), 通过改进的混合初始化方法、基于概率选择的多操作协同的嗅觉搜索、贪婪迭代的视觉搜索、双种群协同搜索以及基于切换机制的双模式局部搜索, 从而取得有效的求解效果.

1 问题描述

1.1 区间数比较

定义区间数 $\tilde{x} = [x^L, x^R]$, 其中 x^L 和 x^R 分别是区间数 \tilde{x} 的左右极限. 若 $x^L = x^R$, 则 \tilde{x} 是确定数.

对于 2 个区间数 $\tilde{x} = [x^L, x^R]$, $\tilde{y} = [y^L, y^R]$, 其相加运算定义如下:

$$\tilde{x} + \tilde{y} = [x^L + y^L, x^R + y^R]. \quad (1)$$

对于区间数 \tilde{x} 和 \tilde{y} , 区间可能度记作 $P(\tilde{x} \leq \tilde{y})$ 或 $P(\tilde{x} \geq \tilde{y})$, 用以表示 $\tilde{x} \leq \tilde{y}$ 或 $\tilde{x} \geq \tilde{y}$ 的概率. Jiang 等^[17] 提出了如下的比较方法:

$$P(\tilde{x} \leq \tilde{y}) = \begin{cases} 0, & x^L \geq y^R; \\ 0.5 \times \frac{(y^R - x^L)^2}{(x^R - x^L)(y^R - y^L)}, & y^L \leq x^L < y^R \leq x^R; \\ \frac{y^L - x^L}{x^R - x^L} + 0.5 \times \frac{y^R - y^L}{x^R - x^L}, & x^L < y^L < y^R \leq x^R; \\ \frac{y^L - x^L}{x^R - x^L} + \frac{(x^R - y^L)(y^R - x^R)}{(x^R - x^L)(y^R - y^L)} + 0.5 \times \frac{(x^R - y^L)^2}{(x^R - x^L)(y^R - y^L)}, & x^L < y^L \leq x^R < y^R; \\ \frac{y^R - x^R}{y^R - y^L} + 0.5 \times \frac{x^R - x^L}{y^R - y^L}, & y^R - x^R < y^R - y^L; \\ 1, & x^R \leq y^L. \end{cases} \quad (2)$$

$$\begin{cases} \tilde{x} < \tilde{y}, & P(\tilde{x} \leq \tilde{y}) > 0.5; \\ \tilde{x} > \tilde{y}, & P(\tilde{x} \leq \tilde{y}) < 0.5; \\ \tilde{x} = \tilde{y}, & P(\tilde{x} \leq \tilde{y}) = 0.5; \\ \tilde{x} \equiv \tilde{y}, & x^L = y^L, x^R = y^R. \end{cases} \quad (3)$$

定义 $\max(\tilde{x}, \tilde{y}) = [\max(x^L, y^L), \max(x^R, y^R)]$. 显然, $\max(\tilde{x}, \tilde{y})$ 为区间数. 对于 n 个区间数 $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n$ 而言, $\max(\tilde{x}_i) = [\max(x_1^L, x_2^L, \dots, x_n^L), \max(x_1^R, x_2^R, \dots, x_n^R)]$, $i = 1, 2, \dots, n$.

1.2 IDPFSP描述

IDPFSP研究如何将 n 个工件 $J = \{J_1, J_2, \dots, J_n\}$ 分配到 f 个工厂进行加工, 其中每个工厂有相同的 m 台机器 $M = \{M_1, M_2, \dots, M_m\}$. 工件 J_i 需要按顺序依次通过 m 个操作 $\{O_{i1}, O_{i2}, \dots, O_{im}\}$ 进行加工, 操作 O_{ij} 表示工件 J_i 在机器 M_j 上的加工操作, 加工时间为区间数 $\tilde{p}_{ij} = [p_{ij}^L, p_{ij}^R]$.

假设: 所有工件相互独立且在零时刻均可加工; 机器连续可用, 即不考虑机器故障等因素; 同一时刻, 一台机器只能加工一个工件, 且一个工件不会被多个机器同时加工; 每个工件可以分配到任一工厂, 并且工厂分配一旦确定就不能改变. 同一工厂的各个机器上加工的工件序列相同, 每个工件拟进行加工的工序也相同; 工件在机器上移动的时间和机器的设置时间忽略不计 (或包含在加工时间内). 由此, 最大完工时间 \tilde{C}_{\max} 的计算方式如下:

$$\tilde{C}_{\pi^k(1),1} = \tilde{p}_{\pi^k(1),1}, \quad k = 1, 2, \dots, f; \tag{4}$$

$$\tilde{C}_{\pi^k(i),1} = \tilde{C}_{\pi^k(i-1),1} + \tilde{p}_{\pi^k(i),1}, \quad k = 1, 2, \dots, f, \quad i = 2, 3, \dots, n; \tag{5}$$

$$\tilde{C}_{\pi^k(1),j} = \tilde{C}_{\pi^k(1),j-1} + \tilde{p}_{\pi^k(1),j}, \quad k = 1, 2, \dots, f, \quad j = 2, 3, \dots, m; \tag{6}$$

$$\tilde{C}_{\pi^k(i),j} = \max(\tilde{C}_{\pi^k(i),j-1}, \tilde{C}_{\pi^k(i-1),j}) + \tilde{p}_{\pi^k(i),j}, \quad k = 1, 2, \dots, f, \quad i = 2, 3, \dots, n, \quad j = 2, 3, \dots, m; \tag{7}$$

$$\tilde{C}_{\max} = \max(\tilde{C}_{\pi^k(n_k),m}), \quad k = 1, 2, \dots, f. \tag{8}$$

其中: \tilde{C}_{\max} 为区间最大完工时间, $\tilde{C}_{i,j} = [C_{i,j}^L, C_{i,j}^R]$ 为操作 O_{ij} 的区间完工时间, π^k 为工厂 F_k 中工件加工序列 $\pi^k = \{\pi^k(1), \pi^k(2), \dots, \pi^k(n_k)\}$.

图1为IDPFSP一个调度解的甘特图, 共有4个工件, 2个工厂, 每个工厂内有2台机器. 其中, 虚线上方为工件的区间开始加工时间, 虚线下方是工件的区间完工时间. 其中 J_1 与 J_3 在 M_1 上的开始加工时间都为 $[0, 0]$, 图中对其加工时间进行加宽便于显示效果.

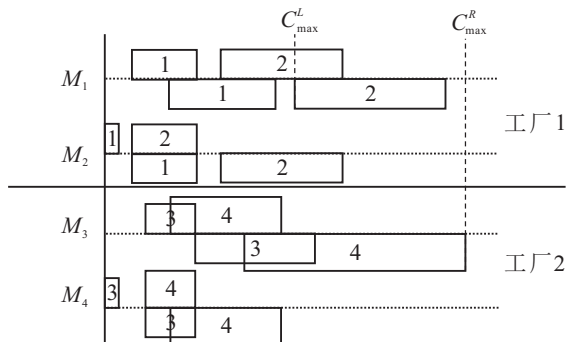


图1 IDPFSP算例的甘特图

2 混合离散果蝇优化算法

果蝇优化算法主要包括嗅觉搜索与视觉搜索两个环节. 嗅觉搜索为发散性搜索, 视觉搜索为贪婪迭代, 果蝇种群通过两个环节的不断迭代实现进化. 由于控制参数较少, 简单易行, 已在诸多领域得到成功应用. 本节针对IDPFSP特性设计一种混合离散果蝇优化算法.

2.1 编码和解码

采用 n 个工件的置换编码序列表示工件的分配和加工优先级.

为了得到可行且相对有效的调度方案, 采用ECF规则^[9]对编码序列解码, 将工件按序依次分配到各工厂中. 对于序列中的待分配工件, 分别放置到每一个工厂的末位后面, 计算当前工厂的完工时间, 将其分配到加工完成此工件最早的工厂.

2.2 种群初始化

初始种群包括 Indi_num 个个体, 其中一个个体由启发式方法 NEH2_im 构造而成, 其余个体随机生成. NEH2_im 是本文针对 IDPFSP 对 NEH2^[5] 改进而提出的, 其步骤如下.

step 1: 将所有工件按其加工总时间降序排列, 并按排列顺序依次分配工件.

step 2: 将当前待分配工件 J_{current} , 分配至每个工厂的各个位置, 保留使得当前 \tilde{C}_{\max} 最小的序列.

step 3: 选取 J_{current} 当前位置的前一个工件, 同样分配至每个工厂的各个位置, 保留使得当前 \tilde{C}_{\max} 最小的序列.

step 4: 若没有改善 \tilde{C}_{\max} , 则选取 J_{current} 当前位置的后一个工件分配至每个工厂的各个位置, 保留使得当前 \tilde{C}_{\max} 最小的序列.

step 5: 返回 step 2 直到所有工件都被分配.

NEH2_im 能构造得到质量较好的个体, 因此上述混合方式产生的初始种群, 既能保证一定的质量又能保证一定的多样性.

2.3 嗅觉搜索与视觉搜索

嗅觉搜索是FOA的核心环节, 对性能有很大影响. HDFOA的嗅觉搜索采用基于概率选择的多邻域搜索操作协同的方式实现. 3种邻域搜索操作如下.

1) 插入操作 (insert): 在置换编码序列中随机挑选 SN 个工件, 插入到 SN 个不同位置中.

2) 交换操作 (swap): 在置换编码序列中随机挑选 SN 个工件, 与 SN 个不同位置的工件进行交换.

3) 反转操作 (inverse): 在置换编码序列中随机挑选 SN 个工件, 将每个工件与随机不同位置的工件之

间的子序列倒置。

其中SN作为算法参数控制每种操作的使用深度。为有效发挥3种操作的性能,采用基于概率选择的协同方式实现嗅觉搜索。 $p_{i,g}$ 、 $p_{s,g}$ 、 $p_{v,g}$ 分别表示第g代种群选择插入、交换、反转操作的概率, $p_{i,1}$ 、 $p_{s,1}$ 、 $p_{v,1}$ 均为1/3; $C_{Si,g}$ 、 $C_{Ss,g}$ 、 $C_{Sv,g}$ 分别表示在第g代种群进化中采用插入、交换、反转改善的个体数; $p_{i,g}$ 、 $p_{s,g}$ 、 $p_{v,g}$ 按下式在种群每一代进化后更新:

$$p_{s,g} = \max\left(0.1, \frac{C_{Ss,g-1} + 1/3}{C_{Si,g-1} + C_{Ss,g-1} + C_{Sv,g-1}}\right), \quad (9)$$

$$p_{v,g} = \max\left(0.1, \frac{C_{Sv,g-1} + 1/3}{C_{Si,g-1} + C_{Ss,g-1} + C_{Sv,g-1}}\right), \quad (10)$$

$$p_{i,g} = 1 - p_{s,g} - p_{v,g}. \quad (11)$$

为防止出现分子分母为0的情况,设置各操作的最低选择概率为0.1。

2.4 双种群协同搜索

标准FOA缺少个体或种群间的协同,HDFOA中设计了基于学习机制的双种群协同搜索阶段。将种群按照性能优劣分成两个子种群,优势种群进行增强搜索进一步改善性能,劣势种群中的个体向优势种群学习。

2.4.1 增强搜索

对于完工时间最大的工厂 F_k ,称为关键工厂,设计如下4种搜索操作。

- 1) Insert_ F_k : 将 J_{i1} 插入到关键工厂内的另一随机位置中;
- 2) Swap_ F_k : 交换 J_{i1} 与 J_{i2} 的位置;
- 3) F_Insert: 将 J_{i1} 插入到 J_{i3} 的位置后;
- 4) F_Swap: 交换 J_{i1} 与 J_{i3} 的位置。

增强搜索操作顺序执行以上4种邻域操作,其中 J_{i1} 、 J_{i2} 为关键工厂中随机挑选的两个工件, J_{i3} 为加工时间最小的工厂内随机挑选的工件。若新解的 \tilde{C}_{max} 有所改善则接受新解并替换。

2.4.2 学习机制

劣势种群中的每个个体 Ff_{ri} ,将向优势种群中随机选择的个体 Ff_k 进行学习,通过信息交互,指引劣势个体的搜索方向,产生2个子代个体 Ff_{ri1} 、 Ff_{ri2} 。若最好的子代优于父代个体,则用其替换父代个体。具体实现如下。

step 1: 分别选择 Ff_k 、 Ff_{ri} 中加工时间最短工厂的工厂信息生成两个新个体 Ff_{ri1} 、 Ff_{ri2} 。

step 2: 对于 Ff_{ri} ,删除与 Ff_{ri1} 中重复的工件,得到

子序列res 1;类似地,对于 Ff_k ,删除与 Ff_{ri2} 中重复的工件,得到子序列res 2。

step 3: 将res 1中的工件按照ECF规则依次分配至 Ff_{ri1} 中,将res 2中的工件按照ECF规则依次分配至 Ff_{ri2} 中。

step 4: 选择 Ff_{ri1} 、 Ff_{ri2} 中最优个体,如果优于 Ff_{ri} ,则替换 Ff_{ri} 。

例如 $Ff_{ri} = (6, 8, 2, 7, 5, 4, 1, 3)$, $Ff_k = (8, 1, 4, 7, 6, 2, 5, 3)$,学习过程如图2所示。

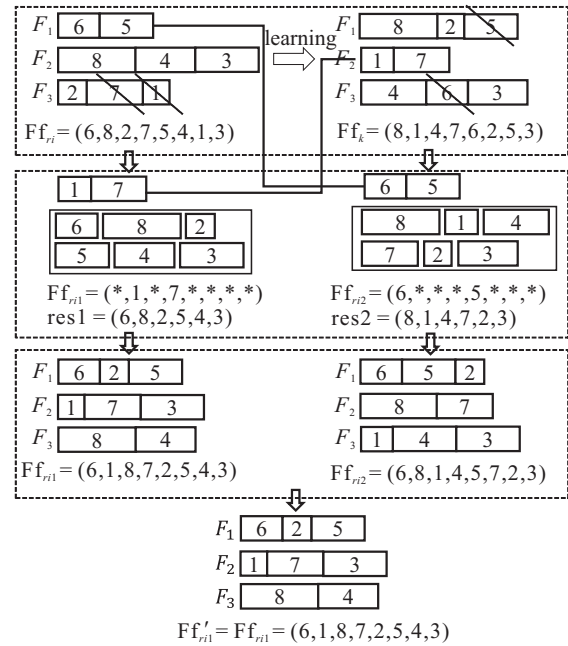


图2 学习机制示例

2.5 基于切换机制的双模式局部搜索

为了增强局部搜索能力,对最优解和次优解设计基于切换机制的局部搜索:若种群中的最优解(或次优解)连续 k_{max} 代搜索没有改善性能,则下一代采用次优解(或最优解)进行搜索。

局部搜索分为LS1和LS2两种模式,LS1采用基于关键工厂和完工时间最小工厂的多种局部搜索操作来强化细搜索,包括F_Swap、Swap_ F_k 、Insert_ F_k 、Inverse_ F_k 。前3种操作与种群协同阶段的增强搜索环节相同。Inverse_ F_k 执行如下:从关键工厂中随机选两个不相邻工件,然后对它们之间的工件序列进行逆序操作,进而择优保留。LS1模式的搜索深度利用参数ls进行控制。

LS2模式的步骤如下。

step 1: 从关键工厂中随机选出工件 J_i ,将其分配到关键工厂内的每个位置,保留使得 \tilde{C}_{max} 最小的序列与位置;

step 2: 随机选择 J_i 当前位置的前一个工件或后一个工件,同样进行全插入,保留使得当前 \tilde{C}_{max} 最小

的序列。

LS2模式的搜索深度利用自适应参数

$$\ln = \lceil \min(jn_{F_k}, 200 \times f/n) \rceil \quad (12)$$

进行控制,每次搜索保证选的工件均不同, jn_{F_k} 为 F_k 中的工件数量, $\lceil \cdot \rceil$ 表示向上取整。

在局部搜索执行过程中,首先对待搜索的个体(第1代采用最优解)进行LS1搜索,若个体的性能有所改善,则下一代执行LS2搜索,否则下一代继续执行LS1搜索。

2.6 算法流程

HDFOA的流程如图3所示,在混合初始化之后,算法通过嗅觉搜索、视觉搜索、双种群协同搜索、基于切换机制的局部搜索等多个阶段的不断循环实现种群的进化,最终获得满意解。

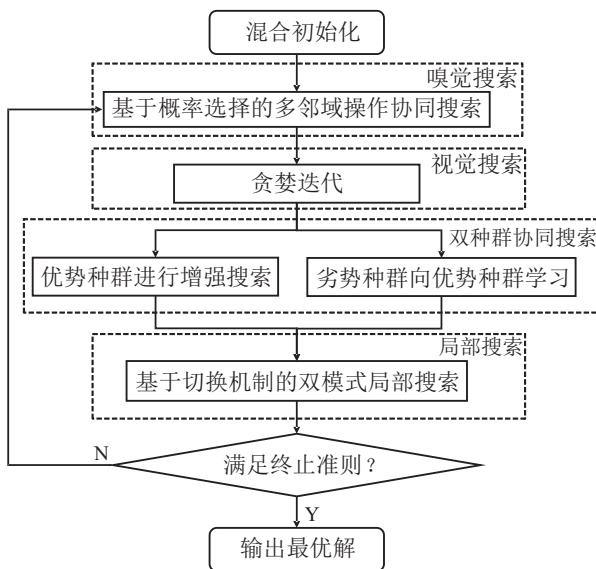


图3 HDFOA流程

3 仿真测试和比较

3.1 实验设置

HDFOA采用C++编程,运行环境为3.2GHz主频的CPU,Intel Core i5处理器.将DPFSP标准测试集的大规模算例进行扩展得到IDPFSP的测试集.采用文献[20]的方法,区间加工时间 $\tilde{p}_{ij} = [p_{ij}^L, p_{ij}^R]$,其中 p_{ij}^L 为标准算例中的加工时间, $p_{ij}^R = p_{ij}^L + \delta$, $\delta \in [0, 4]$.测试集共有 $n \times m$: $\{(20, 50, 100) \times 5\}$, $\{(20, 50, 100, 200) \times 10\}$, $\{(20, 50, 100, 200, 500) \times 20\}$ 共12种组合,每种组合产生10个不同的算例,进而考虑6种工厂数的情况,即 $f = \{2, 3, 4, 5, 6, 7\}$.所有问题的确定加工时间及其已知最优解详见<http://soa.iti.es>.

3.2 参数设置

采用实验设计方法(design of experiment, DOE)^[27]考察参数对算法性能的影响.由于IDPFSP是DPFSP

的扩展,在此采用Ta14_18算例进行DOE测试,每个参数设4个水平值,如表1所示。

表1 HDFOA参数的水平取值

参数	水平			
	1	2	3	4
Indi_num	100	150	200	250
SN	1	2	3	4
ls	100	150	200	250
k_{max}	15	20	25	30

根据正交表 $L_{16}(4^4)$ 对每组参数组合独立运行10次,以500000次总评价为终止准则,算法所得平均性能作为响应值(response value, RV),结果如表2所示.进而,计算得到各参数在每一水平值下的平均响

表2 参数正交表及RV

参数组合编号	参数				RV
	Indi_num	SN	ls	k_{max}	
1	1	1	1	1	771.7
2	1	2	2	2	766.0
3	1	3	3	3	767.5
4	1	4	4	4	769.6
5	2	1	2	3	763.4
6	2	2	1	4	762.8
7	2	3	4	1	765.8
8	2	4	3	2	763.9
9	3	1	3	4	762.0
10	3	2	4	3	763.9
11	3	3	1	2	762.0
12	3	4	2	1	766.2
13	4	1	4	2	764.4
14	4	2	3	1	762.0
15	4	3	2	4	762.9
16	4	4	1	3	769.6

表3 各参数平均RV

水平	Indi_num	SN	ls	k_{max}
1	768.7	765.4	766.5	766.4
2	764.0	763.7	764.6	764.1
3	763.5	764.5	763.9	766.1
4	764.7	767.3	765.9	764.3
极差	5.2	3.7	2.7	2.3
等级	1	2	3	4

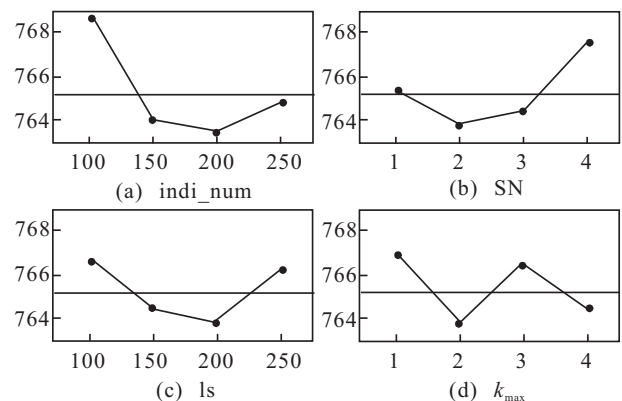


图4 各参数对算法性能影响的趋势

应值,进而确定参数对性能影响的等级,如表3所示.各参数对性能的影响趋势如图4所示.可见,Indi_num对性能的影响最大,其次是SN和ls, k_{max} 对性能的影响最小.最佳参数组合如下:Indi_num=200,SN=2,ls=200, $k_{max}=20$.

3.3 仿真结果与统计分析

采用DPFSP的混合离散布谷鸟算法(HDCS)^[12]在相同的总评价次数下进行性能比较,评价指标为相对百分偏差(relative percentage deviation,RPD)

$$RPD = (\text{alg} - \text{opt}) / \text{opt} \times 100. \quad (13)$$

其中:opt为http://soa.iti.es中的已知最优值,alg为算法所得调度解的 $\tilde{C}_{max}(\text{mid})$ (即 \tilde{C}_{max} 的中点值).RPD值越小,算法所得 \tilde{C}_{max} 的性能越优.对于每个大规模算例,分别用HDCS和HDFOA运行10次,获得 \tilde{C}_{max} 的最优、最差和平均值.对于所有720个算例,按工厂数归类,表4给出了 $\tilde{C}_{max}(\text{mid})$ 的平均RPD值.可见,在每种工厂数下HDFOA的各项统计性能均优于HDCS.

表4 RPD统计结果与对比 %

f	HDCS			HDFOA		
	最优	最差	平均	最优	最差	平均
2	4.03	5.35	4.65	3.54	4.54	4.29
3	4.41	5.94	5.14	3.90	4.94	4.69
4	4.22	5.51	4.83	3.83	4.90	4.65
5	4.37	6.06	5.21	3.88	4.90	4.67
6	4.36	5.79	5.05	3.87	4.89	4.64
7	4.20	5.56	4.83	3.79	4.80	4.56

根据式(2)统计720个算例中HDFOA所得平均 \tilde{C}_{max} 分别优于、劣于或等于HDCS的平均 \tilde{C}_{max} 的比例,如图5所示.可见,HDFOA所得解基本优于或等于HDCS,仅有极个别情况劣于HDCS.

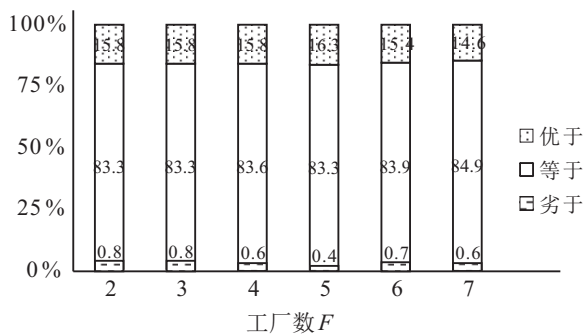


图5 HDFOA平均性能优于/劣于/等于HDCS的比例

表5给出了HDFOA与HDCS的CPU平均运行时间.显然,HDFOA比HDCS更为高效.其主要原因在于HDFOA的种群混合初始化以及对关键工厂内部

的增强性搜索,尤其是基于工厂信息的搜索时间远小于对整个解的评价时间.

表5 CPU时间对比 单位: s

问题规模(按工厂数归类)	HDCS	HDFOA
2	22.28	18.51
3	25.46	21.78
4	31.70	25.49
5	32.05	29.26
6	39.45	33.28
7	43.51	38.03
平均	32.41	27.72

为了进一步判断两种算法性能是否存在显著差异,在90%置信水平下采用非参数检验方法进行双总体t检验.若所得p值小于0.1,则表明HDFOA的性能显著优于HDCS.统计表明,在720个算例中有30.28%的算例p值小于0.1.因此,在相同的总评价次数下,HDFOA的统计性能优于HDCS,并且运行时间更短,HDFOA是求解IDPFSP的一种更为高效的算法.

4 结论

本文提出了求解区间数分布置换流水线问题的一种混合离散果蝇优化调度算法,通过大量算例仿真与数据分析验证了算法的有效性和高效性.算法创新之处在于:提出了改进的启发式规则与随机方法混合初始化种群;设计了基于概率选择机制的多操作协同环节,充分发挥各操作算子的性能;设计了双种群协同搜索,均衡算法的全局探索与局部搜索能力;设计了精英解的双模式局部搜索,进一步改进了解的性能.未来的研究工作将侧重于其他不确定问题的扩展、其他调度指标的优化、自适应算法的设计以及多个调度目标的协同优化.

参考文献(References)

- [1] Grabowski J, Wodecki M. A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion[J]. Computers & Operations Research, 2004, 31(11): 1891-1909.
- [2] Tasgetiren M F, Pan Q K, Suganthan P N, et al. A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops[J]. Information Sciences, 2011, 181(16): 3459-3475.
- [3] Lin Q, Gao L, Li X, et al. A hybrid backtracking search algorithm for permutation flow-shop scheduling problem[J]. Computers & Industrial Engineering, 2015, 85(1): 437-446.
- [4] Lu C, Gao L, Li X, et al. Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective

- backtracking search algorithm[J]. *Journal of Cleaner Production*, 2017, 144(1): 228-238.
- [5] Naderi B, Ruiz R. The distributed permutation flowshop scheduling problem[J]. *Computers & Operations Research*, 2010, 37(4): 754-768.
- [6] Gao J, Chen R. A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem[J]. *Int International Journal of Computational Intelligence Systems*, 2011, 4(4): 497-508.
- [7] Liu H, Gao L. A discrete electromagnetism-like mechanism algorithm for solving distributed permutation flowshop scheduling problem[C]. *International Conference on Manufacturing Automation*. Hong Kong: IEEE, 2011: 156-163.
- [8] Gao J, Chen R, Deng W. An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem[J]. *International Journal of Production Research*, 2013, 51(3): 641-651.
- [9] Wang S Y, Wang L, Liu M, et al. An effective estimation of distribution algorithm for solving the distributed permutation flowshop scheduling problem[J]. *International Journal of Production Economics*, 2013, 145(1): 387-396.
- [10] Shao W, Pi D, Shao Z. Optimization of makespan for the distributed no-wait flow shop scheduling problem with iterated greedy algorithms[J]. *Knowledge-Based Systems*, 2017, 137(1): 163-181.
- [11] Bargaoui H, Driss O B, Ghedira K. A novel chemical reaction optimization for the distributed permutation flowshop scheduling problem with makespan criterion[J]. *Computers & Industrial Engineering*, 2017, 111(1): 239-250.
- [12] Wang J, Wang L, Shen J. A hybrid discrete cuckoo search for distributed permutation flowshop scheduling problem[C]. *Congress on Evolutionary Computation*. Vancouver: IEEE, 2016: 2240-2246.
- [13] Moore R E. *Methods and applications of interval analysis*[M]. New York: Prentice-Hall, 1979: 9-17.
- [14] Ishibuchi H, Tanaka H. Multiobjective programming in optimization of the interval objective function[J]. *European Journal of Operational Research*, 1990, 48(2): 219-225.
- [15] Nakahara Y, Sasaki M, Gen M. On the linear programming problems with interval coefficients[J]. *Computers & Industrial Engineering*, 1992, 23(1/2/3/4): 301-304.
- [16] Nakahara Y. User oriented ranking criteria and its application to fuzzy mathematical programming problems[J]. *Fuzzy Sets & Systems*, 1998, 94(3): 275-286.
- [17] Jiang C, Han X, Liu G R, et al. A nonlinear interval number programming method for uncertain optimization problems[J]. *European Journal of Operational Research*, 2008, 188(1): 1-13.
- [18] Lei D. Population-based neighborhood search for job shop scheduling with interval processing time[M]. *Computers & Industrial Engineering*, 2011, 61(4): 1201-1208.
- [19] Lei D. Interval job shop scheduling problems[J]. *Int J of Advanced Manufacturing Technology*, 2012, 60(1/2/3/4): 291-301.
- [20] Lei D, Cao S. A novel shuffled frog-leaping algorithm for flexible job shop scheduling with interval processing time[C]. *Chinese Control Conference*. Dalian, 2017: 2708-2713.
- [21] Zheng H Y, Wang L. An effective teaching-learning-based optimisation algorithm for RCPSP with ordinal interval numbers[J]. *International Journal of Production Research*, 2015, 53(6): 1777-1790.
- [22] Pan W T. A new fruit fly optimization algorithm: taking the financial distress model as an example[J]. *Knowledge-Based Systems*, 2012, 26(2): 69-74.
- [23] 王凌, 郑晓龙. 果蝇优化算法研究进展[J]. *控制理论与应用*, 2017, 34(5): 557-563.
(Wang L, Zheng X L. Advances in fruit fly optimization algorithms[J]. *Control Theory & Applications*, 2017, 34(5): 557-563.)
- [24] Wang L, Zheng X L. A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem[J]. *Swarm & Evolutionary Computation*, 2018, 38(1): 54-63.
- [25] Zheng X L, Wang L. A two-stage adaptive fruit fly optimization algorithm for unrelated parallel machine scheduling problem with additional resource constraints[J]. *Expert Systems with Applications*, 2016, 65(1): 28-39.
- [26] Yin L J, Li X Y, Gao L, et al. A new improved fruit fly optimization algorithm for traveling salesman problem[C]. *The 8th International Conference on Advanced Computational Intelligence*. Chiang Mai: IEEE, 2016: 21-28.
- [27] Onyiah L C. Design and analysis of experiments[J]. *Handbook of Statistics*, 2001, 30(2): 241-242.

作者简介

王凌(1972—),男,教授,博士生导师,从事智能优化调度理论与方法等研究, E-mail: wangling@tsinghua.edu.cn;

郑洁(1995—),女,博士生,从事智能优化不确定调度的研究, E-mail: j-zheng18@mails.tsinghua.edu.cn;

王晶晶(1995—),女,博士生,从事智能优化不确定调度的研究, E-mail: wjj18@mails.tsinghua.edu.cn.

(责任编辑:孙艺红)