

融入概率学习的混合差分进化算法求解绿色分布式 可重入作业车间调度

胡蓉^{1,3}, 伍星^{2,3†}, 毛剑琳¹, 钱斌¹

(1. 昆明理工大学 信息工程与自动化学院, 云南 昆明 650500; 2. 云南机电职业技术学院, 云南 昆明 650000;

3. 昆明理工大学 机电工程学院, 云南 昆明 650500)

摘要: 本文针对绿色分布式可重入作业车间调度问题 (GDRJSSP), 提出一种融入概率学习的混合差分进化算法 (HDE.PL), 以实现最大完工时间和总能耗最小。根据 GDRJSSP 的问题特点, 设计编码和解码规则, 并采用差分进化算法执行全局搜索来发现优质解区域。为能更明确地引导全局搜索方向, 设计基于贝叶斯网络结构的多维概率模型合理学习和积累优质解 (即当前种群中的较优解) 的模式信息。结合问题解的结构特征, 提出基于关键路径的 4 种邻域结构来构造局部搜索, 并设计基于非关键路径的节能策略来提升算法获取低能耗非劣解的能力。仿真实验和算法对比验证了 HDE.PL 可有效求解 GDRJSSP。

关键词: 差分进化; 绿色调度; 分布式调度; 可重入作业车间调度问题

引用格式: 胡蓉, 伍星, 毛剑琳, 等. 融入概率学习的混合差分进化算法求解绿色分布式可重入作业车间调度. 控制理论与应用, 2024, 41(3): 512–521

DOI: 10.7641/CTA.2023.20875

Hybrid differential evolution integrated with probability learning for green distributed reentrant job shop scheduling

HU Rong^{1,3}, WU Xing^{2,3†}, MAO Jian-lin¹, QIAN Bin¹

1. Faculty of Information Engineering and Automation,

Kunming University of Science and Technology, Kunming Yunnan 650500, China;

2. Yunnan Vocational College of Mechanical and Electrical Technology, Kunming Yunnan 650000, China;

3. Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming Yunnan 650500, China)

Abstract: Aiming at the green distributed reentrant job shop scheduling problem (GDRJSSP), a hybrid differential evolution incorporated with probabilistic learning (HDE.PL) is proposed to minimize the maximum completion time and the total energy consumption. According to the problem characteristics of the GDRJSSP, the rule of job allocation among factories and the encoding and decoding rules are designed, and the differential evolution algorithm is used to perform global search to find high-quality solution regions. In order to guide the global search direction more clearly, a multi-dimensional probability model based on Bayesian network structure is designed to reasonably learn and accumulate the pattern information of high-quality solutions (i.e., the better solutions in the current population). Combined with the structural characteristics of the problem solution, four neighborhoods based on the critical path are proposed to construct the local search, and an energy saving strategy based on the non-critical path is devised to enhance the ability of the algorithm to obtain low-power non-dominated solutions. Simulation experiments and algorithm comparisons verify that HDE.PL can effectively solve the GDRJSSP.

Key words: differential evolution; green scheduling; distributed scheduling; reentrant job shop scheduling problem

Citation: HU Rong, WU Xing, MAO Jianlin, et al. Hybrid differential evolution integrated with probability learning for green distributed reentrant job shop scheduling. *Control Theory & Applications*, 2024, 41(3): 512–521

收稿日期: 2022–10–08; 录用日期: 2023–12–20.

†通信作者. E-mail: xwu@kust.edu.cn; Tel.: +86 871-65939017.

本文责任编辑: 周平.

国家自然科学基金项目(62173169, 61963022), 云南省基础研究重点项目(202201AS070030)资助.

Supported by the National Natural Science Foundation of China (62173169, 61963022) and the Basic Research Key Project of Yunan Province (202201AS070030).

1 引言

绿色制造作为同时考虑环境和能耗的生产模式,是工业生产实现高质量、可持续发展的重要环节.近年,中国碳达峰、碳中和“双碳”目标的提出和推行,对我国能源及制造业在节能减排方面提出了更高要求.企业要在竞争激烈的市场中胜出,生产成本的有效控制是关键.如何对生产过程进行优化调度实现降本增效,已成为企业的生命线.此外,公司之间生产合作和企业兼并现象日益普遍,分布式制造已经成为一种重要生产模式^[1].在上述背景下,有必要深入开展绿色分布式生产调度方面的研究.

作业车间调度问题(job shop scheduling problem, JSSP)是工业生产中一类非常重要的问题.经典JSSP的研究通常局限于工件在各机器或设备上只加工1次.然而,在钢铁生产、半导体制造等战略性行业中,部分关键工件需在同一机器上处理多次(如钢卷退火和晶圆加工)^[2].这表明研究绿色分布式可重入作业车间调度问题(green distributed reentrant JSSP, GDRJSSP)的建模与求解,具有重要现实意义.在计算复杂性上, JSSP归约于GDRJSSP,而JSSP属于典型NP难问题,故GDRJSSP也为NP难问题.因此,研究该问题也具有显著理论价值.

绿色生产调度的已有研究主要聚焦于单工厂或车间调度问题,包括绿色单机调度问题^[1]、绿色并行机调度问题^[3]、绿色流水车间调度问题^[4]、绿色混合流水车间调度问题^[5]、绿色作业车间调度问题^[6]和绿色柔性作业车间调度问题^[7].在分布式工厂方面,已有研究还较为有限,主要为绿色分布式流水车间调度问题^[8]、绿色分布式装配流水车间调度问题^[9]、绿色分布式作业车间调度问题^[10].对于GDRJSSP,尚无研究报道,故亟需对其开展研究.

生产调度问题多为NP难问题,具有非凸几何结构,其主要求解算法包括运筹学算法和智能优化算法.运筹学算法(如分枝定界、列生成、分枝定价)追求获取问题最优解.然而,对于非凸优化问题,至今还没有多项式时间可获取最优解的通用算法.这使得运筹学算法需遍历或部分遍历问题解空间,从而无法在较短时间内得到较大规模问题的满意解.智能优化算法追求尽快获取问题优质解,其特点在于可通过对解空间较少区域的搜索来实现对目标值较广且较优区域的搜索^[11].这使其能在较短时间内获取问题的优质解,从而成为目前求解各类复杂调度问题(包括文献[1, 3–10]的问题)的主要算法.

差分进化(differential evolution, DE)算法将群体中个体间差异和竞争产生的群体智能用于优化搜索方向,具有较强的全局搜索能力和内在的并行搜索特性,被成功用于有效求解各种复杂优化问题^[12].DE的

研究受到众多学者的关注,已成功应用于求解交通运输、电力系统、化工制药等领域中的优化问题求解.在生产调度问题求解方面,现有的有效DE均采用混合算法框架,即利用DE对问题解空间执行全局搜索,同时根据问题特点加入合适的局部搜索,从而形成某种混合DE.譬如,Wang等^[13]提出一种离散DE求解阻塞流水车间调度问题,其优化目标为最大完工时间最小.Wu等^[14]针对分布式装配柔性作业车间调度问题,设计结合模拟退火的混合DE进行求解,以实现提前/延迟和总成本同时最小.对于GDRJSSP,目前没有基于DE的求解算法研究,故本文设计混合DE进行求解.

综上,本文提出一种融入概率学习的混合DE(hybrid DE incorporated with probabilistic learning, HDE-PL),用于求解优化目标为同时最小化总完工时间和总能耗的GDRJSSP.首先,分析问题特点,设计编码规则来确定搜索空间,并设计活动化解码规则来提升解的质量.其次,采用DE执行全局搜索,同时加入基于贝叶斯网络结构的多维概率模型合理学习和积累优质解(即当前种群中的较优解)的模式信息,提升算法驱动全局搜索较快探测或感知到解空间中优质解区域的能力.下一步,分析问题解的结构特征,设计基于关键路径的4种邻域结构来对当前非劣解集中各非劣解执行较深入的局部搜索,进而利用基于非关键路径的节能策略尝试降低各非劣解对应的总能耗.最后,将算法应用于各种规模的测试问题进行对比分析,验证了HDE-PL能够有效求解GDRJSSP.

2 问题描述

2.1 GDRJSSP描述

GDRJSSP可描述为:考虑 n 个工件 $J = \{J_1, \dots, J_n\}$,分别分配到 F 个工厂上加工,每个工厂均有 m 台加工机器 $M = \{M_1, \dots, M_m\}$,每台机器存在 s 种可调节的速度档位且对应加工速度表示为 $V = \{v_1, \dots, v_s\}$.按工艺要求每个工件需要重复加工 L 次,即工件 J_i 的第 j 道工序用 $O_{i,j}$ 来表示,工件 J_i 完成加工工序 $\{O_{i,1}, \dots, O_{i,m \times L}\}$ 的数量是 $m \times L$,相应的基本加工时间为 $p_{i,j}$.由于机器的加工速度可调整,当机器以速度 v_z 加工工序 $O_{i,j}$ 时,实际加工时间为 $p_{i,j,z} = p_{i,j}/v_z$.设工序 $O_{i,j}$ 的开工时间为 $S_{i,j}$,完工时间为 $C_{i,j}$,工厂 k 的完工时间为 C^k ,工厂 k 中机器 M_j 的开工时间为 $S^{j,k}$.当机器 M_j 以速度 v_z 加工时,单位时间能耗为 $EP_{j,z}$.当机器 M_j 上未加工工件时,则该机器处于待机状态,其单位时间能耗为 EI_j .令 $\pi = \{\pi_1, \dots, \pi_{n \times m \times L}\}$ 为整体的工序序列, $\pi^f = \{\pi_1^f, \dots, \pi_{n_f^f \times m \times L}^f\}$ 为工厂 f 中的工序序列, n_f 为分配到工厂 f 中的工件数,其中 $\sum_{f=1}^F n_f = n$.

同时,生产环境设定为:在零时刻即可加工,即工件到达时间设置为零;机器间缓冲区大小不受限制;机器可采用数个档位分别进行匀速加工,每台机器只能完成一道工序的加工;每道工序需在前一道工序加工结束后才可加工;不能打断工序在机器上的加工.优化或调度目标是确定工件在工厂上的分配情况、各机器上工件工序在加工顺序以及机器速度设置,使得生产效率指标 C_{\max} 和总能耗指标 (total energy consumption, TEC) 达到最优.基于以上描述,建立如下数学模型:

$$\min(C_{\max}, \text{TEC}), \quad (1)$$

$$\sum_{r=0, r \neq i}^n \sum_{k=1}^F x_{r,i,1,k} = 1, \forall i, \quad (2)$$

$$\sum_{r=0, r \neq i}^n x_{r,i,j,k} = \sum_{r=0, r \neq i}^n x_{r,i,1,k}, \forall i, j > 1, k, \quad (3)$$

$$\sum_{i=1, i \neq r}^n x_{r,i,j,k} \leq \sum_{i=0, i \neq r}^n x_{i,r,1,k}, \forall r > 0, j, k, \quad (4)$$

$$\sum_{i=1}^n x_{0,i,j,k} = 1, \forall j, k, \quad (5)$$

$$\sum_{k=1}^F (x_{r,i,j,k} + x_{i,r,j,k}) \leq 1, \forall r, i \neq r, j, \quad (6)$$

$$\sum_{j=1, j \neq t}^m y_{i,t,j} = 1, \forall i, t, \quad (7)$$

$$(S_{i,t} + p_{i,t,z}) y_{i,t,j} \leq S_{i,j}, \forall i, t, j \neq t, z, \quad (8)$$

$$S_{r,j} + p_{r,j,z} \leq S_{i,j} + Q(1 - x_{r,i,j,k}), \quad (9)$$

$$\forall r > 0, i \neq r, j, k, z,$$

$$S^{j,k} \leq S_{i,j} + Q(1 - x_{0,i,j,k}), \forall i, j, k, \quad (10)$$

$$\sum_{r=0, r \neq i}^n x_{r,i,1,k} C_{i,j} \leq C^k, \forall i, j, k, \quad (11)$$

$$S_{i,j} + p_{i,j,z} \leq C_{i,j}, \forall i, j, z, \quad (12)$$

$$C_{i,j} \leq C_{\max}, \forall i, j, \quad (13)$$

$$\sum_{z=1}^s h_{i,j,z} = 1, \forall i, j, \quad (14)$$

$$\text{EP}_{j,z} = \psi_j \times v_z^2, \quad (15)$$

$$\text{EI}_j = \psi_j/4, \quad (16)$$

$$\text{TEC} = \sum_{k=1}^F \sum_{j=1}^m \text{EI}_j (C^k - S^{j,k}) + \sum_{i=1}^n \sum_{j=1}^m \sum_{z=1}^s h_{i,j,z} p_{i,j,z} (\text{EP}_{j,z} - \text{EI}_j), \quad (17)$$

$$x_{r,i,j,k} \in \{0, 1\}, \quad (18)$$

$$y_{i,t,j} \in \{0, 1\}, \quad (19)$$

$$h_{i,j,z} \in \{0, 1\}, \quad (20)$$

其中:式(1)为优化目标,即要求最大完工时间和总能耗同时最小;式(2)保证每个工件都能唯一分配到一个工厂中;式(3)–(4)保证同一工件的所有操作都必须分配到同一工厂中且工件的操作满足工艺约束;每台机

器的首个加工工件由式(5)确定;工件在机器上加工顺序的唯一性由式(6)保证;式(7)–(8)限定任意时刻一个工件最多只能在一台机器上处理;式(9)限定一台机器最多只能同时处理一个工件,其中 Q 为极大的正数;式(10)确定每台机器的开始加工时间;式(11)确定每个工厂的完工时间;式(12)–(13)确定最大完工时间 C_{\max} ;式(14)限定任意时刻一个操作仅能在同一速度下进行处理;式(15)计算加工状态;式(16)计算待机状态的总能耗,其中 ψ_j 为机器 M_j 的能耗系数;式(17)计算加工过程中的总能量消耗.此外,设置二进制决策变量 $x_{r,i,j,k}$, $y_{i,t,j}$ 和 $h_{i,j,z}$.若工件 j 为工厂 k 中机器 M_j 上加工工件 r 的后继工件,置 $x_{r,i,j,k} = 1$,否则 $x_{r,i,j,k} = 0$;若工件 i 在机器 M_t 上完工后继续在机器 M_j 上加工,置 $y_{i,t,j} = 1$,否则 $y_{i,t,j} = 0$;若工件 i 的第 j 个操作 $O_{i,j}$ 以速度 v_z 处理,置 $h_{i,j,z} = 1$,否则 $h_{i,j,z} = 0$.

2.2 Pareto最优

GDRJSSP需同时考虑最小化生产效率指标及能耗指标,属于多目标优化问题.该类问题需通过支配关系来确定最优解集.关于多目标优化中的支配、非劣解、非劣解集、Pareto最优解集等概念,可参见文献[15].在求解NP难的多目标优化问题(如GDRJSSP)时,智能优化算法追求在较短时间内获取尽量逼近 Pareto最优解集的优质非劣解集.

3 融入概率学习的混合DE算法(HDE_PL)

GDRJSSP需将工件分配给不同的工厂并确定各工厂内的工件加工顺序,同时还要确定各工厂内机器的加工速度.因此,采用三元组 $(\pi, \mathbf{A}, \mathbf{V})$ 来表示 HDE_PL 的个体(即问题的可行解),其中 π 为工序序列, \mathbf{A} 为工厂分配向量, \mathbf{V} 为加工速度矩阵.

智能优化算法均是采用相应机制不断生成新个体或新解来实现搜索的.在HDE_PL的全局搜索部分,采用DE进化机制改变当前个体中的 π 来生成新个体,并利用概率学习模型采样机制生成与当前个体中的 π 和 \mathbf{A} 均不同的新个体,同时用较优新个体替换旧个体并更新非劣解集,从而逐步引导搜索前往问题解空间中存在优质解的区域.在HDE_PL的局部搜索部分,先采用多邻域变换机制尝试改进当前非劣解集中每个解(对应存在优质解的区域)中的 π ,再利用调速节能机制尝试改进这些解中的 \mathbf{V} ,从而生成更优的新个体.HDE_PL正是通过上述机制不断合理变换 π , \mathbf{A} 和 \mathbf{V} 的取值,从而实现了对GDRJSSP解空间的搜索.

3.1 编码

工序序列 π 的编码方式为基于工件的工序编码,即 $\pi = \{\pi_1, \dots, \pi_{n \times m \times L}\}$ (实例见表1最后一行).

在采用DE生成新个体时,当前个体中的 \mathbf{A} 和 \mathbf{V} 保持不变,只对 π 进行改变.然而,标准DE的个体为实数向量,只能在连续域空间内进行搜索.为使DE可求

解GDRJSSP, 利用可重入最小值排序(reentrant smallest order value, RSOV)规则^[16], 将DE中的第*i*个实数向量 $\mathbf{X}_i = [x_{i,1}, \dots, x_{i,n \times m \times L}]$ 转换为工件的工序或操作序列 $\boldsymbol{\pi}_i = [\pi_{i,1}, \dots, \pi_{i,n \times m \times L}]$. 根据RSOV规则, 首先将 $\mathbf{X}_i = [x_{i,1}, \dots, x_{i,n \times m \times L}]$ 按升序排列后得到中间序列 $\boldsymbol{\varphi}_i = [\varphi_{i,1}, \dots, \varphi_{i,n \times m \times L}]$, 再将中间序列 $\boldsymbol{\varphi}_i$ 映射为工序编码序列 $\boldsymbol{\pi}_i$. 这里 $\pi_{i,k} = \lceil (\varphi_{i,k} - 1) / (n \times m \times L) \rceil + 1$, 其中 $\lceil * \rceil$ 为取整操作, 如 $\lceil 3.54 \rceil = 3$. 表1给出 $n = 2, m = 1, L = 3$ 时 \mathbf{X}_i 到 $\boldsymbol{\pi}_i$ 的转换示例.

表1 调度解的表达

Table 1 Illustration of scheduling solution

<i>k</i>	1	2	3	4	5	6
$x_{i,k}$	1.52	3.81	2.57	0.65	2.66	0.81
$\varphi_{i,k}$	3	6	4	1	5	2
$\pi_{i,k}$	1	2	2	1	2	1

此外, 工厂分配向量 \mathbf{A} 和加工速度矩阵 \mathbf{V} 的编码为: \mathbf{A} 中第*i*个元素 a_i 表示工件*i*分配至工厂 a_i 加工, \mathbf{V} 中第*i*行第*j*列的元素 $v_{i,j}$ 表示工序 $O_{i,j}$ 的机器加工档位. 图1为 $n = 5, m = 3, L = 2, F = 2, \mathbf{A} = [1, 2, 1, 2, 2]$ 时一个调度解的甘特图.

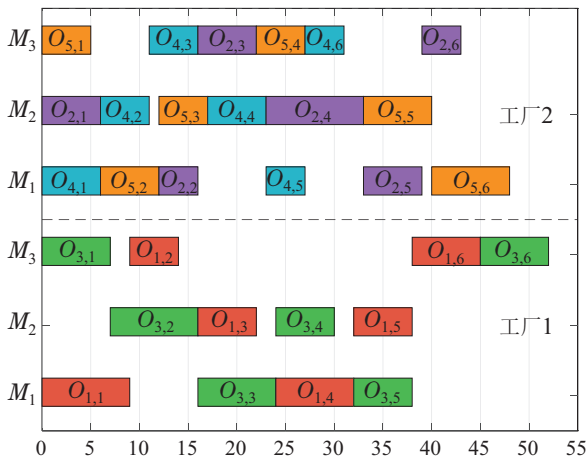


图1 GDRJSSP调度解的甘特图

Fig. 1 Gantt chart of GDRJSSP's scheduling solution

3.2 解码

传统JSSP的解码规则主要有无延迟解码、半活动化解码、活动化解码. 对于GDRJSSP的解, 本文拓展现有的单工厂活动化解码规则并用于对其解码, 以尽量缩减工序完工时间, 具体步骤如下:

步骤1 为使各工厂中工序的完工时间整体较小, 每个工件在分配时优先分配至当前完工时间最小的工厂, 依此规则确定工厂分配向量 \mathbf{A} . 根据 \mathbf{A} 将*n*个工件 $\{J_1, J_2, \dots, J_n\}$ 依次分配到不同的工厂中.

步骤2 各工厂内每台机器上的工序加工顺序确定后, 计算每个工序加工的开始时间和完工时间.

步骤2.1 根据机器的加工速度矩阵 \mathbf{V} , 计算每个工序在每台机器上的实际加工时间. 对于在机器 M_x 上以速度 v_x 加工的工序 $O_{i,j}$, 令 $p_{i,j,z}$ 为其实际加工时间, T_s 为其上一道工序 $O_{i,j-1}$ 的完工时间.

步骤2.2 依次确定每个工序的开始加工时间和完工时间. 从零时刻到当前时刻逐一检查机器 M_x 上已有工序间的时间间隔 $[S_x, E_x]$, 如满足 $\max\{T_s, S_x\} + p_{i,j,z} \leq E_x$, 则工序 $O_{i,j}$ 的开始加工时间设置为 $S_{i,j} = \max\{T_s, S_x\}$, 其完工时间设置为 $C_{i,j} = S_{i,j} + p_{i,j,z}$; 否则继续向右检查后续时间间隔. 如不存在满足条件的时间间隔, 则工序 $O_{i,j}$ 指派到机器 M_x 上的最后工序之后, 以当前时刻开始加工.

步骤3 根据式(13)(17)分别计算问题解 $\boldsymbol{\pi}$ 的最大完工时间 C_{\max} 和总能耗指标TEC.

3.3 概率学习模型

对于GDRJSSP的解 $(\boldsymbol{\pi}, \mathbf{A}, \mathbf{V})$, 工序块定义为工序序列 $\boldsymbol{\pi}$ 中连续相邻的两道工序. 显然, $\boldsymbol{\pi}$ 由不同位置上的工序块所组成. 为能有效学习优质解中 $\boldsymbol{\pi}$ 内部工序块的分布信息, 设计统计工序块出现频次的贝叶斯网络(见图2), 并以此构建工序块分布概率学习模型. 通过采样该模型生成新种群来实现与DE不同的搜索, 可丰富HDE_PL的全局搜索行为并增强其引导全局搜索到达优质解区域的能力.

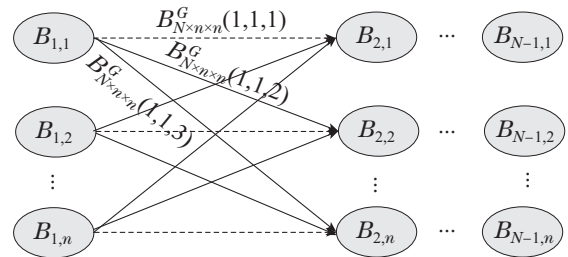


图2 N-1层贝叶斯网络(工序块频次统计模型)

Fig. 2 N-1 layer Bayesian network (statistical model of process block frequency)

3.3.1 频次统计矩阵

设第*G*代优质子种群 \mathbf{BPop}^G 中的第*i*个个体为 $\hat{\mathbf{X}}_i^G = [\hat{x}_{i,1}^G, \hat{x}_{i,2}^G, \dots, \hat{x}_{i,N}^G]$ ($N = n \times m \times L$), \mathbf{BPop}^G 中的个体数量为sps, $\mathbf{B}_{N \times n \times n}^G$ 为第*G*代时*N*层贝叶斯网络不同节点间的权重矩阵. $B_{N \times n \times n}^G(x, y, z)$ 为 $\mathbf{B}_{N \times n \times n}^G$ 中对应的权重或元素 ($x = 1, \dots, N - 1; y, z = 1, \dots, n$), 即贝叶斯网络第*x*层中节点*y*至第*x + 1*层中节点*z*之间连接弧的权重. 该权重为截止第*G*代时, 每代优势子种群中各个体第*x*位置和第*x + 1*位置出现工序块 $[y, z]$ 的累计频次. 譬如, 图2中 $B_{N \times n \times n}^G(1, 1, 2)$ 为截止第*G*代时, 每代优势子种群中各个体第1和第2位置出现工序块 $[1, 2]$ 的累计频次.

显然, 基于贝叶斯网络结构的多维矩阵 $\mathbf{B}_{N \times n \times n}^G$ 能准确统计并积累各代优质个体中工序序列 $\boldsymbol{\pi}$ 的全部

序关系信息和工序块分布信息,这克服了现有智能调度算法中常用的2维统计矩阵无法区分优质个体中不同位置工序块的不足.

同样,对于解 $(\pi, \mathbf{A}, \mathbf{V})$ 中的工厂分配向量 \mathbf{A} ,设计两层贝叶斯网络权重矩阵 $\mathbf{BF}_{n \times F}^G$ 学习优质解中 \mathbf{A} 内部工件分配至工厂的信息,其中 $\mathbf{BF}_{n \times F}^G(y, z)$ 为 $\mathbf{BF}_{n \times F}^G$ 中对应的权重或元素,即贝叶斯网络第1层中节点 y 至第2层中节点 z 之间连接弧的权重.该权重等于截止第 G 代时,每代优势子种群中各个体内工件 J_y 在工厂 z 上的累计分配频次.

3.3.2 概率学习模型

基于 $\mathbf{B}_{N \times n \times n}^G$ 和 $\mathbf{BF}_{n \times F}^G$,可分别构建工序块分布概率学习模型 $\mathbf{PB}_{N \times n \times n}^G$ 和工件分配概率学习模型 $\mathbf{PBF}_{n \times F}^G$.令 $\mathbf{PB}_{N \times n \times n}^G(x, y, z)$ 为 $\mathbf{PB}_{N \times n \times n}^G$ 中的元素, $\mathbf{PBF}_{n \times F}^G(y, z)$ 为 $\mathbf{PBF}_{n \times F}^G$ 中的元素,

$$\mathbf{PB}_{N \times n \times n}^G(x) = \begin{bmatrix} \mathbf{PB}_{N \times n \times n}^G(x, 1, 1) \cdots \mathbf{PB}_{N \times n \times n}^G(x, 1, n) \\ \vdots \\ \mathbf{PB}_{N \times n \times n}^G(x, n, 1) \cdots \mathbf{PB}_{N \times n \times n}^G(x, n, n) \end{bmatrix}_{n \times n} \quad (21)$$

为 $\{\mathbf{BPop}^1, \dots, \mathbf{BPop}^G\}$ 的个体中第 x 位置和和第 $x+1$ 位置上工序块的累计概率分布,

$$\mathbf{PBF}_{n \times F}^G = \begin{bmatrix} \mathbf{PBF}_{n \times F}^G(1, 1) \cdots \mathbf{PBF}_{n \times F}^G(1, F) \\ \vdots \\ \mathbf{PBF}_{n \times F}^G(n, 1) \cdots \mathbf{PBF}_{n \times F}^G(n, F) \end{bmatrix}_{n \times F} \quad (22)$$

为 $\{\mathbf{BPop}^1, \dots, \mathbf{BPop}^G\}$ 个体中工件分配至工厂的累计概率分布, $\mathbf{PBF}_{n \times F}^G(y)$ 为 $\mathbf{PBF}_{n \times F}^G$ 中第 y 行, $\mathbf{B}_{N \times n \times n}^G(x)$ 为 $\mathbf{B}_{N \times n \times n}^G$ 中第 x 层(类似 $\mathbf{PB}_{N \times n \times n}^G(x)$), $\mathbf{BF}_{n \times F}^G(y)$ 为 $\mathbf{BF}_{n \times F}^G$ 中第 y 行, r_1 和 r_2 分别为 $\mathbf{PB}_{N \times n \times n}^G$ 和 $\mathbf{PBF}_{n \times F}^G$ 的学习速率(见如下步骤3).同时,设定

$$\begin{aligned} T_B^G(x) &= \sum_{y=1}^n \sum_{z=1}^n B_{N \times n \times n}^G(x, y, z), \\ T_{PB}^G(x) &= \sum_{y=1}^n \sum_{z=1}^n \mathbf{PB}_{N \times n \times n}^G(x, y, z), \\ T_{BF}^G(y) &= \sum_{z=1}^F \mathbf{BF}_{n \times F}^G(y, z), \\ T_{PBF}^G(y) &= \sum_{z=1}^F \mathbf{PBF}_{n \times F}^G(y, z), \\ T_{BF}^G &= \sum_{y=1}^n \sum_{z=1}^F \mathbf{BF}_{n \times F}^G(y, z). \end{aligned}$$

根据以上定义, $\mathbf{PB}_{N \times n \times n}^G$ 和 $\mathbf{PBF}_{n \times F}^G$ 按如下更新.

步骤1 当 $G=0$ 时,进行初始化.

$\mathbf{PB}_{N \times n \times n}^0$ 的第1层(即 $\mathbf{PB}_{N \times n \times n}^0(1)$)中各元素

赋值为 $1/n$,其余层各元素赋值为 $1/n^2$,同时 $\mathbf{PBF}_{n \times F}^G$ 中各元素赋值为 $1/F$.

步骤2 当 $G=1$ 时,由式(23)–(24)更新.

$$\mathbf{PB}_{N \times n \times n}^1(x) = \begin{cases} \mathbf{B}_{N \times n \times n}^0(x)/T_B^0(x), & x=1, \\ (\mathbf{PB}_{N \times n \times n}^0(x) + \mathbf{B}_{N \times n \times n}^0(x))/(T_{PB}^0(x) + T_B^0(x)), & x=2, \dots, N-1, \end{cases} \quad (23)$$

$$\mathbf{PBF}_{n \times F}^1(y) = (\mathbf{PBF}_{n \times F}^0(y) + \mathbf{BF}_{n \times F}^1(y))/(T_{PBF}^0(y) + T_{BF}^1(y)), \quad y=1, \dots, n. \quad (24)$$

步骤3 当 $G>1$ 时,由式(25)–(26)更新.

$$\begin{aligned} \mathbf{PB}_{N \times n \times n}^G(x) &= (1-r_1) \times \mathbf{PB}_{N \times n \times n}^{G-1}(x) + r_1 \times \\ &\mathbf{B}_{N \times n \times n}^G(x)/T_B^G(x), \quad x=1, \dots, N-1, \\ \mathbf{PBF}_{n \times F}^G &= (1-r_2) \times \mathbf{PBF}_{n \times F}^{G-1} + r_2 \times \mathbf{BF}_{n \times F}^G/T_{BF}^G. \end{aligned} \quad (25)$$

步骤4 $G=G+1$,若 $G \leq \text{Max } G$ 则转步骤3.

3.4 全局搜索

如前所言,全局搜索靠DE进化机制和概率学习模型采样机制分别生成新种群来实现.DE生成新种群的过程见第2.6节的HDE_PL算法流程.概率学习模型采样生成新种群的过程如下.

通过对概率学习模型 $\mathbf{PB}_{N \times n \times n}^G$ 和 $\mathbf{PBF}_{n \times F}^G$ 进行采样,可得新的工序序列 π 、工厂分配向量 \mathbf{A} ,然后再结合初始加工速度矩阵 \mathbf{V} ,就能生成新个体或解.

对于 $\mathbf{Pop}(G)$ 的个体 i 中工序序列 $\pi_i^G = [\pi_{i,1}^G, \pi_{i,2}^G, \dots, \pi_{i,N}^G]$,其首位置工序 $\pi_{i,1}^G$ 按如下确定:

步骤1 按式(27)计算 $\text{SN}_{\text{init}}^{G-1}(y)$, $y=1, \dots, n$.

$$\text{SN}_{\text{init}}^{G-1}(y) = \sum_{z=1}^n \text{PM}_{N \times n \times n}^{G-1}(1, y, z). \quad (27)$$

步骤2 利用轮盘赌方法选择首位置上工件 J_s .随机生成实数 $\gamma \in [0, \sum_{h=1}^n \text{SN}_{\text{init}}^{G-1}(h)]$,若 $\gamma \in [0, \text{SN}_{\text{init}}^{G-1}(1)]$,则 $J_s=1$;若 $\gamma \in [\text{SN}_{\text{init}}^{G-1}(t), \text{SN}_{\text{init}}^{G-1}(t+1)]$ ($t \in \{1, \dots, n-1\}$),则 $J_s=t+1$.

步骤3 $\pi_{i,1}^G = J_s$.

然后,为进一步确定 π_i^G 中的子序列 $[\pi_{i,2}^G, \dots, \pi_{i,N}^G]$,设计工件选择函数 $\text{Select } J(\pi_i^G, x+1)$ 来得到 π_i^G 的第 $(x+1)$ 位置上的工件 $\pi_{i,x+1}^G$ ($x=1, \dots, N-1$),其具体过程如下:

步骤1 随机生成 $[0, \sum_{h=1}^n \text{PB}_{N \times n \times n}^{G-1}(x, \pi_{i,x}^G, h)]$

之间的实数 γ .

步骤2 利用轮盘赌选择工件 J_s .

若 $\gamma \in [0, PB_{N \times n \times n}^{G-1}(x, \pi_{i,x}^G, 1))$, 则 $J_s = 1$;

若 $\gamma \in [\sum_{h=1}^t PB_{N \times n \times n}^{G-1}(x, \pi_{i,x}^G, h), \sum_{h=1}^{t+1} PB_{N \times n \times n}^{G-1}(x, \pi_{i,x}^G, h))$ ($t \in \{1, \dots, n-1\}$), 则 $J_s = t + 1$.

步骤3 $\pi_{i,x+1}^G = J_s$.

对于 $\text{Pop}(G)$ 的个体 i 中工厂分配向量 $\mathbf{A}_i = [a_{i,1}, \dots, a_{i,n}]$, 在 $\text{PBF}_{n \times F}^{G-1}$ 每行上执行如上轮盘赌即可确定. 至此, 可生成新个体 i . 然后重复上述步骤, 可生成基于概率学习模型的新种群.

3.5 局部搜索

GDRJSSP属于复杂的多目标组合优化问题. 由于不同优化目标间的相互制约, 优质非劣解通常聚集在庞大解空间中接近底部的不同区域, 这增大了搜索难度. 为提升算法性能, 设计如下局部搜索对优质解区域进行更加深入细致的搜索.

GDRJSSP的关键路径为各工厂的关键路径中最长的一条. 以第3.1节中图1和本节图3为例, 工厂1和工厂2的关键路径分别为各自工厂中红线所连工序. 由于工厂1的关键路径最长, 故整个问题的关键路径为工厂1的关键路径. 结合甘特图分析GDRJSSP的解内部结构可知, 该问题任何解的最大完工时间 C_{\max} 等于其关键路径的长度, 故调整关键路径上的工序有较大概率能减小 C_{\max} . 因此, 设计基于关键路径的4种邻域搜索(对应4种邻域结构), 对每代种群中的部分个体分别执行由这4种邻域结构构成的变邻域局部搜索, 以尽量减小这些个体的 C_{\max} . 进而, 设计非关键路径的节能策略, 尝试降低执行过局部搜索的个体的总能耗TEC.

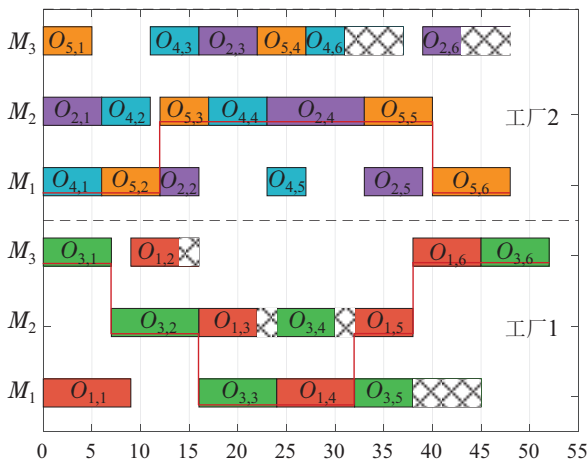


图3 关键路径和调速节能

Fig. 3 Critical path and speed regulation for saving energy

3.5.1 多邻域贪婪搜索

定义问题关键路径所属工厂为关键工厂 f_c , f_c 内分配的工序数为 n_{f_c} , f_c 内关键路径上的工序数为 n_c

(即关键工序数), 非关键工厂内的工序数为 n_{f_e} ($n_{f_e} + n_{f_c} = n \times m \times L$). 以图3为例, 其关键路径为工厂1中红线所连工序. 这些工序即是关键工序, 同时 $f_c =$ 工厂1, $n_{f_c} = 12$, $n_c = 7$, $n_{f_e} = n \times m \times L - n_{f_c} = 5 \times 3 \times 2 - 12 = 18$.

基于关键路径的4种邻域搜索如下:

1) 关键工厂内Interchange: 随机选择1个关键工序, 并将其与该工厂内所有的非关键工序逐一进行交换, 形成 $n_{f_e} - n_c$ 个邻域, 并输出最好邻域.

2) 关键工厂内Insert: 随机选择1个关键工序, 并将其分别插入到该工厂中每个非关键工序之前或之后, 形成 $n_{f_e} - n_c$ 个邻域, 并输出最好邻域.

3) 工厂间Interchange: 随机选择1个关键工序, 并将其与各非关键工厂中所有的工序逐一进行交换, 形成 n_{f_e} 个邻域, 并输出最好邻域.

4) 工厂间Insert: 随机选择1个关键工序, 并将其分别插入各非关键工厂中每个工序之前或之后, 形成 n_{f_e} 个邻域, 并输出最好邻域.

基于上述定义, 设计如下变邻域搜索.

步骤1 为当前非劣解集中的每个个体或解确定1条关键路径, 若有多条则随机确定1条.

步骤2 逐一选择当前非劣解集中的解, 并对其执行上述4种邻域搜索. 每执行完1种邻域搜索后, 判断新解(即输出邻域)是否支配旧解. 如果是, 旧解将被新解所取代, 确定新解关键路径后对新解从刚执行过的邻域搜索开始, 继续执行剩余的邻域搜索; 若两个解互不支配, 则将新解直接加入非劣解集, 继续对旧解执行剩余的邻域搜索; 若新解被旧解支配, 则继续对旧解执行剩余的邻域搜索.

步骤3 非劣解集中解的数量超过 ps 时, 则删除拥挤距离较小的解^[17], 保持非劣解集大小为 ps .

3.5.2 基于非关键路径工序降速的节能策略

对于GDRJSSP的任意解, 通过保持各工厂中关键路径上工序的加工速度不变, 对非关键路径上工序的加工速度进行降档调节, 有可能在最大完工时间 C_{\max} 不变的前提下, 实现总能耗的降低. 基于这一问题特点, 对执行变邻域搜索后的非劣解集中每个非劣解执行如下的调速节能操作:

步骤1 确定各工厂中的关键路径.

步骤2 判断各工厂的非关键路径上每道工序是否处于最低加工速度的档位. 若是则转回步骤2继续判断下一道工序; 若否则进行下一步.

步骤3 将该工序的当前加工速度降低一档, 并判断是否影响关键路径. 若是则保持档位不变, 转到步骤2继续判断下一道工序; 若否则降低一档, 转到步骤2继续判断下一道工序.

执行上述步骤对非关键路径上的所有工序完成降

档尝试后,更新该非劣解的加工速度矩阵 V .

3.6 HDE_PL算法流程

令 $\text{Pop}(G)$ 为第 G 代种群, ps 为 $\text{Pop}(G)$ 中个体数量, T_Ω 为非劣解集, $\text{CR} \in [0, 1]$ 为DE的交叉概率, $\text{rand}(0, 1)$ 为 $[0, 1]$ 上的随机数, F_s 为DE的比例因子, tmp 为DE的测试向量, tmp_k 为 tmp 中的第 k 维元素 ($k=1, \dots, n \times m \times L$), best 为DE的最优向量, best_k 为 best 中的第 k 维元素, $(\pi_{i,\text{DE}}^G, \mathbf{A}_{\text{DE}}^G, \mathbf{V}_{\text{DE}}^G)$ 为DE在第 G 代生成的个体, $(\pi_{i,\text{PL}}^G, \mathbf{A}_{i,\text{PL}}^G, \mathbf{V}_{\text{PL}}^G)$ 为概率学习模型采样在第 G 代生成的个体.

基于第3.1节至第3.5节, HDE_PL整体流程如下:

步骤1 初始化.

步骤1.1 工厂分配向量 \mathbf{A}_{DE} 和 \mathbf{V}_{DE} , 以及加工速度矩阵 \mathbf{V}_{PL} 初始化. 随机生成 \mathbf{A}_{DE} 中工厂排列, 并随机生成 \mathbf{V}_{DE} 和 \mathbf{V}_{PL} 中各元素的加工档位.

步骤1.2 DE初始化.

设置关键参数 ps , sps , F_s , CR .

设置 $\text{lower}(x_{i,k}) = 0$, $\text{upper}(x_{i,k}) = 4$, 其中 $k = 1, \dots, n \times m \times L$, $i = 1, 2, \dots, \text{ps}$. 获取每个 $\mathbf{X}_i^{G=0}$, $x_{i,k}^{G=0} = \text{lower}(x_{i,k}) + \text{rand}(0, 1) \times (\text{upper}(x_{i,k}) - \text{lower}(x_{i,k}))$, 其中 $k = 1, \dots, n \times m \times L$, $i = 1, 2, \dots, \text{ps}$.

步骤1.3 概率学习模型初始化. 根据第3.3.2节初始化 $\mathbf{PM}_{N \times n \times n}^{G=0}$ 和 $\mathbf{PF}_{n \times F}^{G=0}$.

步骤1.4 非劣解集初始化. 设置 $T_\Omega = \emptyset$.

步骤1.5 生成初始种群 $\text{Pop}(G = 0)$.

采用RSOV规则将 $\mathbf{X}_i^{G=0}$ 转化为相应的 $\pi_{i,\text{DE}}^{G=0}$, 并给每个 $\pi_{i,\text{DE}}^{G=0}$ 分配 \mathbf{A}_{DE} 和 \mathbf{V}_{DE} (见第3.1节), 构建 ps 个基于DE的初始个体 $(\pi_{i,\text{DE}}^{G=0}, \mathbf{A}_{\text{DE}}, \mathbf{V}_{\text{DE}})$.

采样 $\mathbf{PM}_{N \times n \times n}^{G=0}$ 和 $\mathbf{PF}_{n \times F}^{G=0}$ 生成 ps 对 $(\pi_{i,\text{PL}}^{G=0}, \mathbf{A}_{i,\text{PL}}^{G=0})$, 并给每对 $(\pi_{i,\text{PL}}^{G=0}, \mathbf{A}_{i,\text{PL}}^{G=0})$ 分配 \mathbf{V}_{PL} , 构建 ps 个基于概率学习模型的初始个体 $(\pi_{i,\text{PL}}^{G=0}, \mathbf{A}_{i,\text{PL}}^{G=0}, \mathbf{V}_{\text{PL}})$.

利用Pareto支配关系从上述个体中择优选择 ps 个个体, 构成 $\text{Pop}(G = 0)$, 进而用 $\text{Pop}(G = 0)$ 中的非劣解构成非劣解集 T_Ω .

步骤2 设置 $G = 1$.

步骤3 生成基于DE的种群 $i = 1$. //步骤3-8

步骤4 $\text{tmp} = \mathbf{X}_i^{G-1}$ 且 $D = 0$. 随机选择 T_Ω 中某个体为 best . 随机产生 l_1 和 l_2 满足 $l_1, l_2 \in \{1, 2, \dots, \text{ps}\}$ 且 $l_1 \neq l_2 \neq i$. 随机选择 $k(k = 1, \dots, n \times m \times L)$.

步骤5 DE变异操作和交叉操作.

步骤5.1 $\text{tmp}_k = \text{tmp}_k + F_s \times (\text{best}_k - \text{tmp}_k) + F_s \times (x_{l_1,k}^{G-1} - x_{l_2,k}^{G-1})$. 若 $\text{tmp}_k < \text{lower}(x_{i,k})$, 则 $\text{tmp}_k = 2 \times \text{lower}(x_{i,k}) - \text{tmp}_k$; 若 $\text{tmp}_k > \text{upper}(x_{i,k})$, 则 $\text{tmp}_k = 2 \times \text{upper}(x_{i,k}) - \text{tmp}_k$.

步骤5.2 $k = (k + 1) \bmod N$ 且 $D = D + 1$.

步骤5.3 若 $\text{rand}(0, 1) < \text{CR}$ 且 $D < N$, 则跳转至步骤5.1.

步骤6 DE选择操作.

如果 tmp 对应的个体支配 \mathbf{X}_i^{G-1} 对应的个体, 则 $\mathbf{X}_i^G = \text{tmp}$, 否则 $\mathbf{X}_i^G = \mathbf{X}_i^{G-1}$.

步骤7 $i = i + 1$. 若 $i \leq \text{ps}$, 则跳转至步骤4.

步骤8 基于步骤3-7生成的 ps 个 \mathbf{X}_i^G , 构建DE的 ps 个个体 $(\pi_{i,\text{DE}}^G, \mathbf{A}_{\text{DE}}, \mathbf{V}_{\text{DE}})$. //类似步骤1.5

步骤9 生成基于概率学习模型的种群.

采样 $\mathbf{PM}_{N \times n \times n}^{G-1}$ 和 $\mathbf{PF}_{n \times F}^{G-1}$ 生成 ps 对 $(\pi_{i,\text{PL}}^G, \mathbf{A}_{i,\text{PL}}^G)$ (见第3.3-第3.4节), 构建基于概率学习模型的 ps 个个体 $(\pi_{i,\text{PL}}^G, \mathbf{A}_{i,\text{PL}}^G, \mathbf{V}_{\text{PL}})$. //类似步骤1.5

步骤10 利用Pareto支配关系从步骤8-9构建的个体中择优选择 ps 个个体, 构成 $\text{Pop}(G)$, 进而用 $\text{Pop}(G)$ 更新非劣解集 T_Ω .

步骤11 从 $\text{Pop}(G)$ 中选择 sps 个个体构成 $\text{SPop}(G)$, 由式(23)-(26)更新 $\mathbf{PM}_{N \times n \times n}^G$ 和 $\mathbf{PF}_{n \times F}^G$.

步骤12 局部搜索.

在非劣解集 T_Ω 中, 先对每个非劣解执行变邻域搜索(见第2.5.1节), 再对其中的每个非劣解执行调速节能操作(见第2.5.2节). //更新了非劣解集 T_Ω

步骤13 令 $G = G + 1$. 若 $G \leq \text{Max } G$, 则跳转至步骤3.

步骤14 输出非劣解集 T_Ω .

从以上算法流程可知, 在HDE_PL的迭代搜索过程中, 非劣解集 T_Ω 在步骤10由DE和概率学习模型共同引导的全局搜索(步骤3-9)进行更新, 同时在步骤12由结合变邻域搜索和调速节能操作的局部搜索进行更新. 由于全局和局部搜索均得以加强, HDE_PL可实现对GDRJSSP解空间范围较广且具有一定深度的搜索.

4 实验设计与分析

4.1 实验设置

选择FT01, FT10, FT20, LA01, LA06, LA11, LA16, LA21, LA26, LA31这10个不同规模的JSSP标准测试算例, 扩展得到适用于GDRJSSP的测试集(拓展算例命名为标准测试算例名称前加“E”, 见表2). 在扩展测试集中, 所有工件的重入次数设置为2次, 加工阶段机器速度设置为5个可调档位(对应加工速度为 $\{1, 1.3, 1.55, 1.75, 2.10\}$), 机器能耗系数设置为 $\psi_j = 4$, 机器加工与待机能耗分别设置为 $\text{EP}_{j,z} = 4 \times v_z^2$ 和 $\text{EI}_j = 1$. 所有算法采用Delphi XE10编程, 运行环境为16 GB内存、i7处理器(3.20 GHz)的PC. 采用如下指标比较算法性能:

1) 距离指标 DI_R , 可度量非劣解集 Ω' 中的元素相对于参考集 Ω^* 的距离, 即

$$DI_R(\Omega') = \frac{1}{|\Omega^*|} \sum_{\mathbf{y} \in \Omega^*} \min\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in \Omega'\}, \quad (28)$$

其中: 解 $\mathbf{x} \in \Omega'$ 与解 $\mathbf{y} \in \Omega^*$ 在归一化优化目标空间中的距离表示为 $d(\mathbf{x}, \mathbf{y})$ ^[18]. $d(\mathbf{x}, \mathbf{y})$ 按式(29)计算:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^G ((f_i(\mathbf{x}) - f_i(\mathbf{y})) / (f_i^{\max} - f_i^{\min}))^2}, \quad (29)$$

式中: f_i 为第 i 个优化目标, f_i^{\min} 和 f_i^{\max} 分别为参考集 Ω^* 中 f_i 的最小和最大值. Ω^* 由所有比较算法得到的非劣解集合并构成.

2) 非劣解占比指标 ρ_r , 可度量由算法获得的非劣解集 Ω' 在 Ω^* 中的比例.

$DI_R(\Omega')$ 越小则表明算法获得的非劣解集更接近Pareto最优解集, 而 ρ_r 越大则表明算法性能越强.

本文将HDE_PL与非劣排序遗传算法(NSGA-II)^[17]、多目标人工蜂群算法(MOABC)^[19]、改进多目标遗传局部搜索算法(IMMOGLS)^[18]这三种高效多目标优化算法进行比较. 多目标优化算法NSGA-II在离散和连续优化领域均应用广泛. MOABC在传统人工蜂群算法中加入数种有效局部搜索算子和交叉操作

算子, 可高效求解绿色多目标分布式作业车间调度问题. IMMOGLS采用随机加权的方式处理多个优化目标, 是公认的求解多目标流水车间调度问题的高效算法. 在算法性能比较实验中, 各算法运行时间均设置为 $f \times n \times m \times 25$ ms, 并都独立重复运行20次. 评价指标 DI_R 中的参考集 Ω^* 由以上4种比较算法每次运行结束时输出的非劣解集合并得到.

4.2 仿真结果与比较

HDE_PL的关键参数为: 种群规模 ps 、优势子种群比率 φ 、学习速率 r_1 和 r_2 、变异系数 F_s 、交叉概率 CR . 采用DOE实验设计法分析参数对HDE_PL全局搜索部分性能的影响. 每个参数均选取5个不同水平值. 对于每组参数, 算法均运行10次, 终止条件为 $f \times n \times m \times 25$ ms. 以算法在不同参数组合下各规模算例的 ρ_r 平均值作为评价指标, 可得如图4所示的参数响应图, 故HDE_PL的推荐参数组合为: $ps = 30, \varphi = 0.3, r_1 = 0.2, r_2 = 0.1, F_s = 0.5, CR = 0.2$. 类似地, 可得NSGA-II的推荐参数组合为: 种群规模 $ps = 30$, 变异概率 $p_m = 0.1$, 交叉概率 $p_c = 0.8$; MOABC的推荐参数组合为: 种群规模 $ps = 30$, 放弃解极限 $L_a = 50$; IMMOGLS的推荐参数组合为: 种群规模 $ps = 30$, 精英个体数 $Ne = 10$, 局部搜索概率 $pLS = 0.8$.

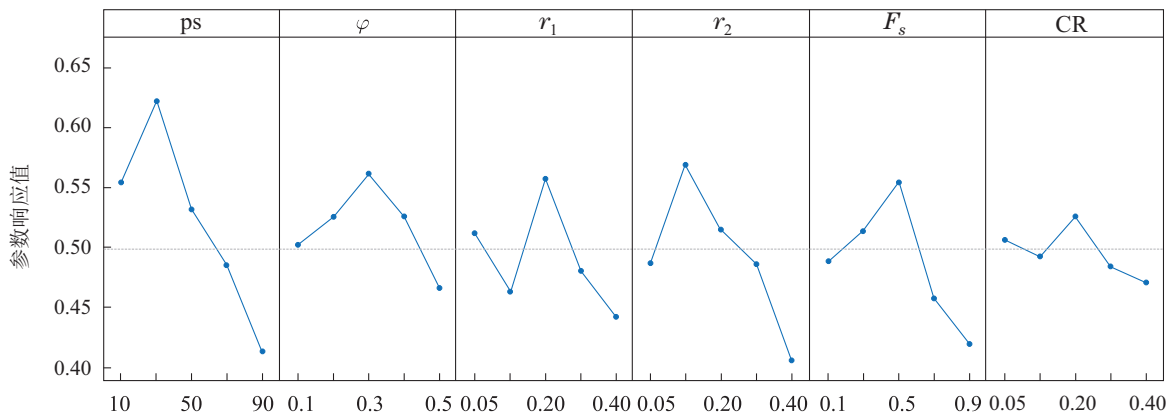


图4 HDE_PL各参数响应趋势图

Fig. 4 Response trend graph of HDE_PL's various parameters

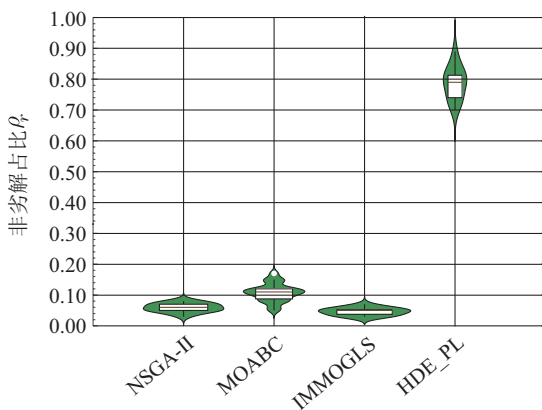
各算法在评价指标 DI_R 和 ρ_r 下的测试结果如表2所示, 其中最好值以粗体表示. 由表2可知, HDE_PL的 DI_R 值在所有问题上小于其他3种算法的 DI_R 值, 同时HDE_PL的 ρ_r 值也明显大于其他3种算法的 ρ_r 值. 这表明HDE_PL获得的非劣解集整体更接近于参考集, 且获得的优质非劣解数量更多. 为直观表现算法间的性能差异, 图5给出这4种算法在算例E.LA16($F = 2$)上的非劣解占比指标 ρ_r 对应的小提琴图(在其他算例上可得类似的图). 从图5也可看出HDE_PL在统计意义上显著占优.

相对于NSGA-II仅靠全局搜索驱动算法寻优, HDE_PL采用全局搜索和局部搜索结合的混合模式, 可以更好兼顾搜索的宽度和深度. 此外, 虽然MOABC和IM-MOGLS两者也是包含全局搜索和局部搜索的混合算法, 但是HDE_PL的全局搜索部分结合分散搜索能力强的DE和引导性强的概率学习模型, 有助于在更宽广的范围发现较多的优质解区域, 同时在局部搜索部分采用基于关键路径的变邻域搜索, 以便于执行更为集中深入的挖掘. 因此, HDE_PL具有更强的搜索引擎.

表2 NSGA-II, MOABC, IMMOGLS和HDE_PL的计算结果

Table 2 Calculation results of NSGA-II, MOABC, IMMOGLS and HDE_PL

F	算例	n, m, L	NSGA-II		MOABC		IMMOGLS		HDE_PL	
			DI _R	ρ_r	DI _R	ρ_r	DI _R	ρ_r	DI _R	ρ_r
2	E_FT06	6,6,2	0.02	0.04	0.01	0.06	0.02	0.03	0.00	0.87
	E_LA01	10,5,2	0.03	0.05	0.02	0.09	0.03	0.05	0.01	0.81
	E_LA06	15,5,2	0.06	0.06	0.04	0.08	0.07	0.05	0.02	0.81
	E_LA11	20,5,2	0.04	0.05	0.03	0.11	0.05	0.04	0.01	0.80
	E_FT10	10,10,2	0.03	0.03	0.02	0.05	0.03	0.03	0.01	0.89
	E_FT20	20,5,2	0.05	0.05	0.03	0.10	0.07	0.03	0.01	0.82
	E_LA16	10,10,2	0.03	0.04	0.02	0.08	0.05	0.04	0.01	0.84
	E_LA21	15,10,2	0.05	0.06	0.03	0.11	0.07	0.05	0.01	0.78
	E_LA26	20,10,2	0.06	0.07	0.03	0.14	0.08	0.05	0.01	0.74
4	E_LA31	30,10,2	0.07	0.07	0.05	0.15	0.09	0.06	0.02	0.72
	E_FT06	6,6,2	0.03	0.05	0.02	0.11	0.05	0.03	0.00	0.81
	E_LA01	10,5,2	0.04	0.07	0.02	0.12	0.06	0.06	0.01	0.75
	E_LA06	15,5,2	0.05	0.07	0.03	0.10	0.08	0.05	0.01	0.78
	E_LA11	20,5,2	0.05	0.08	0.03	0.12	0.07	0.07	0.01	0.73
	E_FT10	10,10,2	0.03	0.05	0.02	0.08	0.05	0.04	0.00	0.83
	E_FT20	20,5,2	0.07	0.06	0.05	0.12	0.09	0.03	0.02	0.79
	E_LA16	10,10,2	0.04	0.05	0.03	0.10	0.06	0.04	0.01	0.81
	E_LA21	15,10,2	0.05	0.08	0.04	0.12	0.07	0.06	0.02	0.74
均值	E_LA26	20,10,2	0.06	0.08	0.05	0.15	0.08	0.05	0.03	0.72
	E_LA31	30,10,2	0.09	0.07	0.07	0.17	0.11	0.06	0.04	0.70
			0.05	0.06	0.03	0.11	0.06	0.05	0.01	0.79

图5 3种算法求解E.LA16($F=2$)算例的非劣解分布图Fig. 5 Distribution of non-inferior via three algorithms for solving the E.LA16 ($F=2$)

5 结语

本文在可重入作业车间调度问题基础上,进一步考虑分布式工厂场景和节能减排需求,确立优化目标为总完工时间和总能耗同时最小,进而建立绿色分布式可重入作业车间调度问题(GDRJSSP)模型.针对GDRJSSP,提出一种融入概率学习的混合DE(HDE_PL)进行求解.在HDE_PL的全局搜索部分,设计DE与多维概率学习结合的协作搜索框架,有利于提升算法引导全局搜索到达解空间中优质解区域的能力.在

HDE_PL的局部搜索部分,为实现对优质解区域较深入的细致搜索,设计相应的多邻域局部搜索.同时,为进一步降低总能耗,设计非关键路径工序加工降速的节能策略.仿真实验验证了HDE_PL的有效性.未来工作作为设计高效HDE_PL求解GDRJSSP和物流集成的调度问题.

参考文献:

- [1] ZHOU S C, JIN M Z, DU N. Energy-efficient scheduling of a single batch processing machine with dynamic job arrival times. *Energy*, 2020, 209(118420): 1 – 16.
- [2] CHEN Tong, QIN Yuanhui, WAN Jianing, et al. Re-entrant flexible scheduling: Models, algorithms and applications. *Systems Engineering-Theory & Practice*, 2015, 35(5): 1187 – 1201. (陈通, 秦远辉, 万家宁, 等. 可重入柔性调度问题研究: 模型、算法与应用. 系统工程理论与实践, 2015, 35(5): 1187 – 1201.)
- [3] SABERI-ALIABAD H, REISI-NAFCHI M, MOSLEHI G. Energy-efficient scheduling in an unrelated parallel machine environment under time-of-use electricity tariffs. *Journal of Cleaner Production*, 2020, 249(119393): 1 – 11.
- [4] DING J Y, SONG S J, WU C. Carbon-efficient scheduling of flow shops by multi-objective optimization. *European J of Operational Research*, 2016, 248(3): 758 – 771.
- [5] ZHANG B, PAN Q K, GAO L, et al. A three-stage multi objective approach based on decomposition for an energy-efficient hybrid flow shop scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, 50(12): 4984 – 4999.
- [6] ABEDI M, CHIONG R, NOMAN N, et al. A multi-population, multi-objective memetic algorithm for energy-efficient job-shop scheduling

- with deteriorating machines. *Expert Systems with Applications*, 2020, 157(113348): 1 – 17.
- [7] LI Y B, HUANG W X, WU R, et al. An improved artificial bee colony algorithm for solving multi-objective low-carbon flexible job shop scheduling problem. *Applied Soft Computing*, 2020, 95(106554): 1 – 14.
- [8] WANG J J, WANG L. A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop. *IEEE Transactions on Systems Man & Cybernetics Systems*, 2018, 50(5): 1805 – 1819.
- [9] ZHANG Ziqi, QIAN Bin, HU Rong, et al. Multidimensional estimation of distribution algorithm for low carbon scheduling of distributed assembly permutation flow-shop. *Control and Decision*, 2022, 37(5): 1367 – 1377
(张梓琪, 钱斌, 胡蓉, 等. 基于多维EDA算法的低碳分布式装配流水线车间调度. *控制与决策*, 2022, 37(5): 1367 – 1377.)
- [10] JIANG E D, WANG L, PENG Z P. Solving energy-efficient distributed job shop scheduling via multi-objective evolutionary algorithm with decomposition. *Swarm and Evolutionary Computation*, 2020, 58(100745): 1 – 16.
- [11] HE Yujie, QIAN Bin, HU Rong. Hybrid discrete teaching-learning-based optimization algorithm for solving complex parallel machine scheduling problem. *Acta Automatica Sinica*, 2020, 46(4): 805 – 819.
(何雨洁, 钱斌, 胡蓉. 混合离散教与学算法求解复杂并行机调度问题. *自动化学报*, 2020, 46(4): 805 – 819.)
- [12] STORN R. System design by constraint adaptation and differential evolution. *IEEE Transactions on Evolutionary Computation*, 1999, 3(1): 22 – 34.
- [13] WANG L, PAN Q K, SUGANTHAN P N, et al. A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems. *Computers & Operations Research*, 2010, 37(3): 509 – 520.
- [14] WU X L, LIU X, ZHAO N. An improved differential evolution algorithm for solving a distributed assembly flexible job shop scheduling problem. *Memetic Computing*, 2019, 11: 335 – 355.
- [15] BAGCHI T P. *Multiobjective Scheduling by Genetic Algorithms*. Springer US, 2012.
- [16] QIAN B, LI Z H, HU R, et al. A hybrid differential evolution algorithm for the multi-objective reentrant job-shop scheduling problem. *The 10th IEEE International Conference on Control & Automation*. Hangzhou: IEEE, 2013: 485 – 489.
- [17] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182 – 197.
- [18] ISHIBUCHI H, YOSHIDA T, MURATA T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 204 – 223.
- [19] XIE J, GAO L, PAN Q, et al. An effective multi-objective artificial bee colony algorithm for energy efficient distributed job shop scheduling. *Procedia Manufacturing*, 2019, 39: 1194 – 1203.

作者简介:

胡蓉 副教授, 硕士生导师, 目前研究方向为智能优化调度、物流优化, E-mail: ronghu@vip.163.com;

伍星 教授, 博士生导师, 目前研究方向为机械声振信号分析、生产过程优化调度, E-mail: xwu@kust.edu.cn;

毛剑琳 教授, 博士生导师, 目前研究方向为网络资源分配和优化, E-mail: 1318524654@qq.com;

钱斌 教授, 博士生导师, 目前研究方向为优化调度理论与方法、智能优化方法, E-mail: bin.qian@vip.163.com.