增强分布估计算法求解低碳分布式流水线调度

杨晓林, 胡　蓉[†], 钱　斌, 吴丽萍

(昆明理工大学 信息工程与自动化学院, 云南 昆明 650500)

摘要: 针对低碳分布式流水线调度问题(DFSP–LC), 提出了一种基于序关系的增强分布估计算法(OEEDA), 用于最小化最大完成时间和总碳排放量. 在 OEEDA 的第 1 阶段, 利用基于贝叶斯统计推断的分布估计算法(BEDA)在问题解空间进行一定时间的搜索, 用于发现优质解并将其保存于非劣解集中. 在 OEEDA 的第 2 阶段, 提出了基于序关系的四维矩阵(OFDM)对优质解的序关系(即工件块结构及其位置信息)进行有效学习和积累, 进而设计了在解中固定部分块结构的采样机制, 可更加明确地指导算法的全局搜索方向. 同时, 引入基于解、工厂间、工厂内的3种不同Insert融合的搜索方式, 对2个阶段全局搜索得到的优质解区域进行较为细致的局部搜索. 最后, 通过仿真实验和算法对比验证了OEEDA的有效性.

关键词: 碳排放; 流水线调度; 序关系; 四维矩阵

引用格式: 杨晓林, 胡蓉, 钱斌, 等. 增强分布估计算法求解低碳分布式流水线调度. 控制理论与应用, 2019, 36(5): 803 – 815

DOI: 10.7641/CTA.2018.70968

Enhanced estimation of distribution algorithm for low carbon scheduling of distributed flow shop problem

YANG Xiao-lin, HU Rong[†], QIAN Bin, WU Li-ping

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming Yunnan 650500, China)

Abstract: An enhanced estimation of distribution algorithm based on ordered relationship (OEEDA) is presented to minimize the makespan and total carbon emission for a low carbon scheduling of distributed flow shop problem (DFSP–LC). In the first stage of OEEDA, an estimation of distribution algorithm based on Bayesian statistical inference (BEDA) is utilized to perform the global search in the problem's solution space for a certain period of time, with the purpose of finding good solutions and storing them in the non-dominated set. In the second stage of OEEDA, a four-dimensional matrix based on ordered relationship (OFDM) is proposed to effectively learn and accumulate the excellent solutions'information of ordered relationship, i.e., the information of job blocks and their corresponding positions. Then, a sampling scheme that fixes some blocks in the solution is designed to guide the global search direction more clearly. Moreover, a search method based on three kinds of Insert operator, i.e., solution-based Insert, inter-factory Insert, and intra-factory Insert, is introduced to execute a more thorough local search from the promising regions obtained by the above two stages' global search. Finally, simulations and comparisons show the efficiency of the proposed OEEDA.

Key words: carbon emission; flow shop scheduling; ordered relationship; four-dimensional matrix

Citation: YANG Xiaolin, HU Rong, QIAN Bin, et al. Enhanced estimation of distribution algorithm for low carbon scheduling of distributed flow shop problem. *Control Theory & Applications*, 2019, 36(5): 803 – 815

1　引言

中国制造2025的提出, 使得智能生产将取代大部分的传统工厂生产模式. 智能生产要求企业在绿色发展、资源分配、调度决策和数据分析等方面能发挥出显著作用. 在全球化的背景下, 随着公司之间生产合作和企业兼并现象的日益普遍, 分布式制造已经成为一种常见的生产模式[1]. 分布式制造可以对资源进行合理分配, 在满足生产方案要求的情况下, 以最小的

收稿日期: 2017−12−29; 录用日期: 2018−06−05.
[†]通信作者. E-mail: ronghu@vip.163.com; Tel.: +86 13508719500.
本文责任编委: 薛安克.
国家自然科学基金项目(51665025), 云南省应用基础研究计划项目(2015FB136), 云南省教育厅科学研究基金项目(2017ZZX149)资助.
Supported by the National Natural Science Foundation of China (51665025), the Applied Basic Research Program of Yunnan Province (2015FB136) and the Scientific Research Foundation Project of Yunnan Education Department (2017ZZX149).

成本实现最大化产出. 但是, 分布式生产在获得经济效益的同时也存在巨大的环境压力和节能压力, 如二氧化碳排放所引起的温室效应, 以及大量的能源消耗[2]. 因此, 研究低碳分布式流水车间调度问题 (low carbon scheduling of distributed flow shop problem, DFSP–LC)具有很强的工业背景. 在计算复杂度上, DFSP属于非确定性多项式(non-deterministic polynomial hard, NP–hard) 问题, 同时 DFSP 又规约为 (reduce to) DFSP–LC, 可得出DFSP–LC也为NP–hard问题. 综上, 开展DFSP–LC有效求解算法的研究具有重要工程和学术意义.

近年来, 在低碳车间调度研究方面, Gilles等[3]提出了基于工件间隔时间的能源效率调度规则, 求解优化目标为最小化总完工时间和能源消耗的单机调度问题. Yan等[4]提出了基于多级节能的遗传算法, 求解优化目标为最小化最大完工时间和能源消耗的柔性流水车间调度问题. Tang等[5]提出了改进的粒子群优化算法, 求解优化目标为最小化最大完工时间和能源消耗的柔性流水车间调度问题. Zhang等[6]提出了多目标遗传算法, 求解优化目标为最小化总加权延迟时间和总能耗的作业车间调度问题. Ding等[7]提出了基于非支配解结构特性的迭代贪心算法, 求解优化目标为最小化最大完工时间和碳排放总量的流水车间调度问题. Ai 等[8]提出了新型蛙跳算法, 采用基于种群和记忆的种群划分策略, 求解优化目标为最小化碳排放量的柔性作业车间调度. Lu等[9]提出了基于机器设置时间和工件运输时间的混合多目标回溯搜索算法, 求解优化目标为最小化最大完工时间和能源消耗的流水车间调度问题. Zheng等[10]提出了基于帕累托解的分布估计算法, 并利用活动模式列表编码, 求解优化目标为最小化最大完工时间和碳排放总量的多目标多模式资源约束调度问题. Dai等[11]提出了改进的遗传模拟退火算法, 求解优化目标为最小化最大完工时间和总能耗的柔性流水车间调度问题. 在分布式车间调度研究方面, Naderi等[12]提出了分散搜索算法, 求解优化目标为最小化最大完工时间的的分布式流水车间调度问题, 并针对该问题已知的720个优质解进行了进一步的优化. Gao等[13]提出了基于工厂分配规则[14]和NEH的启发式算法,求解优化目标为最小化最大完工时间的分布式流水车间调度问题. Xu等[15]提出了混合免疫算法, 利用个体性能对种群进行划分的策略, 求解优化目标为最小化最大完工时间的分布式流水车间调度问题. Rifai等[16]提出了多目标自适应大邻域搜索算法, 用于求解优化目标为最小化最大完工时间、总成本及平均延迟时间的分布式可重入流水车间调度问题. 在低碳分布式流水车间调度研究方面, Deng等[17]提出了竞争文化基因算法, 并通过对分配规则的设计和机器速度的调整, 求解优化目标为最小化最大完工时间和碳排放总量的分布式流水车间调度问题. 由文献调研可知, 低碳分布式流水车间调度问题的研究仍十分有限. 因此, 设计求解该类重要问题的有效算法具有重要意义.

分布估计算法(estimation of distribution algorithm, EDA)在求解车间调度问题的研究中得到了广泛应用. Baluja等[18]在求解旅行商问题时提出了基于种群的增量学习算法, 利用二维变量独立概率模型进行种群更新并且指导算法的搜索方向. Mühlenbein等[19]在1996年提出了EDA的概念, 在此之后, EDA得到了快速发展. 不同于传统进化算法中的交叉和变异操作, EDA是一种基于统计学原理的随机优化方法, 通过对优质解的采样和最佳搜索区域的预测, 可以使EDA在全局搜索中发挥出更显著的作用并且加快算法的收敛速度. Larraanaga等[20]指出, EDA的高效性和搜索能力使其在连续和组合优化领域均有广泛应用. Jarboui等[21]利用二维概率矩阵学习优质解的工件块结构, 进而设计了EDA求解优化目标为最小化总流经时间的流水车间调度问题. Salhi等[22]基于二维概率矩阵, 提出了带引导变异操作的EDA求解复杂的混合流水线车间调度问题. Wang等[23]针对最小化最大完工时间的柔性流水车间调度问题, 建立了0–1混合整数线性规划模型, 对小规模问题采用数学规划方法求解; 同时对大规模问题设计了基于二维概率矩阵的EDA进行求解. Wang等[24]针对具有相同并行机的混合流水车间调度问题, 建立了描述问题解空间分布的二维概率模型, 进而设计了一种改进EDA, 优化目标为最小化最大完工时间. Qian等[25]针对可重入流水车间调度, 提出了基于coupla函数的混合EDA, 优化目标为最小化最大完工时间.

EDA近几年在求解调度问题上得到了快速的发展, 但仍然存在不足之处. 譬如: Rawat等[26]利用遗传算法求解流水车间调度问题时, 算法的交叉操作容易破坏优质个体中的相似工件块结构, 会导致算法难以在优质解区域进行搜索. 对于EDA, 如果其采样生成新个体时破坏了优质个体的工件块结构, 同样会使得算法性能不佳, 为提高算法性能, Chen等[27]针对带序相关和设置时间的可重入作业车间调度问题提出了基于贝叶斯统计推断的分布估计算法(estimation of distribution algorithm based on Bayesian statistical inference, BEDA), 可在一定程度上避免EDA生成种群时工件块结构被破坏的问题. 但是, 已有的EDA调度算法基本上都采用二维概率矩阵, 难以同时保留和学习优质解中的工件块结构及其位置信息, 算法性能较有限. 因此, 本文提出了一种基于序关系的增强分布估计算法(enhanced estimation of distribution algorithm based on ordered relationship, OEEDA) 用于求解优化目标为最小化最大完成时间和总碳排放量的

DFSP–LC. 在所提算法的第1阶段, 利用BEDA发现并保存优质解于非劣解集中; 进而在算法第2阶段, 设计了基于序关系的四维矩阵(four-dimensional matrix based on ordered relationship, OFDM)对优质解的序关系(即工件块结构及其位置信息)进行有效学习和积累, 并提出了在解中固定部分块结构的采样机制, 可更加明确地引导算法在优质解区域进行搜索. 同时, 引入多种Insert融合的搜索方式, 对两阶段全局搜索得到的优质解区域进行较为细致的局部搜索, 使得算法全局和局部搜索之间达到较好平衡. 最后, 通过仿真实验和算法对比验证了OEEDA为求解DFSP–LC的有效算法.

2　问题描述

调度问题根据求解方法可分别建立2类模型, 即排列模型和数学规划模型. 如果采用智能优化方法, 则需建立排列模型. 该模型由各工件在各台机器上的开始加工时间或完工时间计算公式组成, 并不显式出现约束, 其约束是隐式包含在问题解的编码中. 排列模型的问题解是基于工件或操作进行编码的, 即解中每一位的取值是1至工件数或1至操作数. 如果采用割平面、列生成等传统运筹学方法, 则需建立数学规划模型. 调度问题的数学规划模型一般为0–1整数或混合整数规划模型, 其问题解中每一位(即决策变量)的取值为0或1, 同时决策变量的约束由多个不等式约束或等式约束显式给出. 不少中外重要期刊的文献在用智能算法求解各类调度问题时, 也会给出数学规划模型, 这对理解问题有一定意义. 当然, 这些文献并未直接求解数学规划模型. 因此, 本节只给出如下排列模型.

DFSP–LC包含f个工厂和n个工件, 其中每个工厂k分配n_k个工件, 每个工厂k均为流水线车间, 每个工厂内有m台机器, 每台机器均有s种加工速度组成($S=\{v_1,v_2,\cdots,v_s\}$), n个工件需分配到不同的工厂进行加工操作, $O_{i,j}$表示工件i在机器j上的操作, 该操作在加工过程中不可中断, 每个操作$O_{i,j}$都有相应的标准加工时间$p_{i,j}$. 每台机器的每个加工操作速度选定后, 在该加工操作完成前不可更改. 当机器j以速度$V_{i,j}$加工操作$O_{i,j}$, 此时加工操作$O_{i,j}$的加工时间变为$P_{i,j}/V_{i,j}$, 单位时间能源消耗为$PP_{j,v}$. 当机器处于空闲状态时, 机器将处于待机模式, 单位时间能源消耗为SP_j. 规定对于同一个操作$O_{i,j}$, 如果机器速度增大则加工时间相应减少, 能耗相应增加. Deng等[17]给出了假设: 机器速度由v增加到u, 则需满足$p_{i,j}/u<p_{i,j}/v$, $PP_{j,u}\cdot p_{i,j}/u>PP_{j,v}\cdot p_{i,j}/v$. π^k表示工厂k的工件加工序列, $\pi^k(i)$表示π^k中第i个位置的工件. 所有工厂的整体调度方案集合为$\Pi=(\pi^1,\pi^2,\cdots,\pi^f;V)$. 每个工厂的加工完工时间为$C(k)$, 碳排放量为$CE(k)$.

DFSP–LC与PFSP有相同的约束条件, 如每台机器在同一时刻至多加工一个工件, 每个工件在同一个时刻至多在一台机器上加工, 加工过程中不允许中断和工件抢占, 各工件必须按照工艺路线以指定的次序在机器上加工等. DFSP–LC分为3个子问题: 工件对工厂分配、机器速度选择、调度. 工件对工厂分配即利用分配规则对所有工厂进行工件的分配, 机器速度选择即确定机器加工时的速度, 调度即确定每个工厂内的所有工件在机器上的加工顺序. DFSP–LC的目标函数为最大完工时间$C_{\max}$和碳排放总量TCE, 具体定义如下所示:

$$C_{\pi^k(1),1}=\frac{p_{\pi^k(1),1}}{V_{\pi^k(1),1}}, \tag{1}$$

$$C_{\pi^k(i),1}=C_{\pi^k(i-1),1}+\frac{p_{\pi^k(i),1}}{V_{\pi^k(i),1}}, \tag{2}$$

$$C_{\pi^k(1),j}=C_{\pi^k(1),j-1}+\frac{p_{\pi^k(1),j}}{V_{\pi^k(1),j}}, \tag{3}$$

$$C_{\pi^k(i),j}=\max\left\{C_{\pi^k(i-1),j},C_{\pi^k(i),j-1}\right\}+\frac{p_{\pi^k(i),j}}{V_{\pi^k(i),j}}, \tag{4}$$

$$C(k)=C_{\pi^k(n_k),m}, \tag{5}$$

$$C_{\max}=\max\left\{C(1),C(2),\cdots,C(f)\right\}, \tag{6}$$

$$CE(k)=\varepsilon\cdot\int_0^{C(k)}(\sum_{v=1}^{s}\sum_{j=1}^{m}pp_{jv}\cdot X_{kj}(t)+\sum_{j=1}^{m}SP_j\cdot(1-X_{kj}(t)))\mathrm{d}t, \tag{7}$$

$$TCE=\sum_{k=1}^{f}CE(k), \tag{8}$$

其中: $X_{kj}(t)$为一个表示工厂k中的机器j在t时刻处于加工状态, 为0表示处于待机状态. ε为能耗与碳排放量之间的转换系数, 通常取0.7559. 图1所示为两个工厂的甘特图和能源消耗曲线. 由图1中的数据可以确定$C_{\max}=C(2)$, $TCE=(CE(1)+CE(2))$.

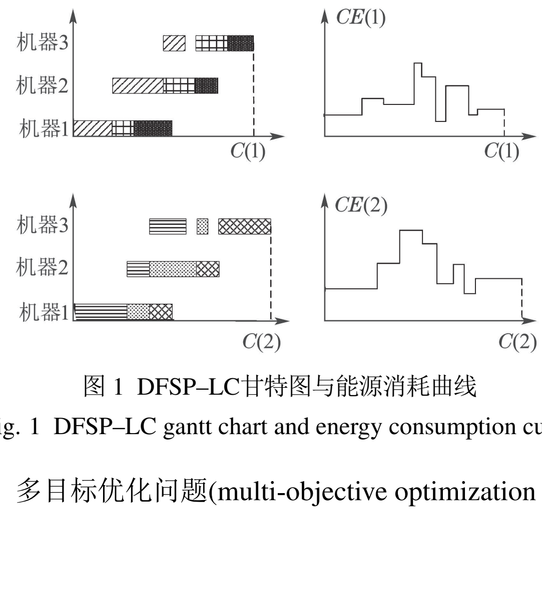


图 1　DFSP–LC甘特图与能源消耗曲线

Fig. 1　DFSP–LC gantt chart and energy consumption curve

多目标优化问题(multi-objective optimization problem, MOP)即求解多个目标函数的优化问题, 不同解的目标函数值可能会相互支配. 如果问题中有ω个目标函数, 则该优化问题可描述如下:

$$\begin{cases} \min\ f_1(x), f_2(x), \cdots, f_\omega(x), \\ \text{s.t.}\ x \in X, \end{cases} \tag{9}$$

$f_1(x), f_2(x), \cdots, f_\omega(x)$是MOP中需要优化的$\omega$个目标函数, x是一个决策变量即一个调度方案, X是该问题可行解的集合.

对于解x_1和x_2, 如果解x_1支配解x_2, 则需满足

$$\begin{cases} \forall i \in \{1, 2, \cdots, \omega\}:\ f_i(x_1) \leqslant f_i(x_2), \\ \exists j \in \{1, 2, \cdots, \omega\}:\ f_j(x_1) < f_j(x_2). \end{cases} \tag{10}$$

MOP中, 如果一个解不受搜索空间内的任何解支配则称该解非劣解. MOP中的所有非劣解构成非劣解集. 对于DFSP–LC, 第1个目标函数为$C_{\max}$(见式(6)), 第2个目标函数为TCE(见式(8)).

3 OEEDA求解DFSP–LC

3.1 OEEDA算法流程

步骤 1 分别初始化种群和OFDM的参数, 生成初始非劣解集;

步骤 2 对非劣解集中的解进行局部搜索, 更新非劣解集;

步骤 3 根据非劣解集更新BEDA, 采样生成新种群, 对当前种群中的非劣解进行局部搜索, 更新非劣解集;

步骤 4 判断算法是否满足OFDM重启条件, 若是则执行步骤5; 否则转至步骤3;

步骤 5 利用非劣解集更新OFDM, 选出B_n个块结构, 根据第3.5节生成新种群, 对当前种群中的非劣解进行局部搜索, 更新非劣解集;

步骤 6 判断终止条件是否满足? 若是则输出结果; 否则转至步骤5.

3.2 编码与解码

对于 DFSP–LC, 本文采用基于工件的编码方式, 令 $\delta = \{\delta(1), \delta(2), \cdots, \delta(n)\}$ 为该问题的一个解. 本文采用最迟完工工厂规则(latest completion factory, LCF)[14]作为解码方式, 将工件分配到各工厂. 同时, 提出了一种将各工厂加工子序映射到问题解的逆向解码规则(reverse LCF, RLCF). 该规则可以确保在执行工厂内和工厂间的Insert局部搜索后得到的各工厂子序能映射到一个唯一的新解, 从而使得局部搜索发现的优质解或个体序关系得以保留, 有利于概率矩阵学习和更新优质解信息. LCF与RLCF的伪代码如下所示:

Procedure LCF rule.

For $k = 1$ to F

$\pi^k(1) = \delta(k)$;

$n_k = 1$;

For $k = F + 1$ to n

Find the factory f that can process job $\delta(k-1)$ with the lowest current $C_{\max}$;

$n_f = n_f + 1$;

$\pi^f(n_f) = \delta(k)$.

Procedure RLCF rule.

For $k = 1$ to F

$\delta(k) = \pi^k(1)$;

$n_k = 1$;

For $k = F + 1$ to n

Find the factory f that can process job $\delta(k-1)$ with the lowest current $C_{\max}$;

$\delta(k) = \pi^f(n_f + 1)$;

$n_f = n_f + 1$.

下面通过一个实例阐述LCF工件分配过程. 设有5个工件需分配到2个工厂进行加工, 各工厂均包含3台相同的机器. 工件加工时间表如表1所示, $\delta=\{2, 3, 1, 4, 5\}$. 根据LCF, 首先将工件2和工件3分别分配至f_1与f_2, 由于$C_{\max}(f_1) < C_{\max}(f_2)$, 因此将工件1分配至$f_1$, 此时, $C_{\max}(f_1) > C_{\max}(f_2)$, 则将工件4分配至$f_2$, 对于工件5, 由于$C_{\max}(f_1) > C_{\max}(f_2)$, 所以将工件5分配至$f_2$. 得到$\pi^1 = \{2, 1\}$, $\pi^2 = \{3, 4, 5\}$, 如图2所示.

表 1 加工时间

Table 1 Processing times

机器序号	工件序号				
	n_1	n_2	n_3	n_4	n_5
m_1	50	20	30	20	40
m_2	20	10	10	30	20
m_3	30	40	50	10	40

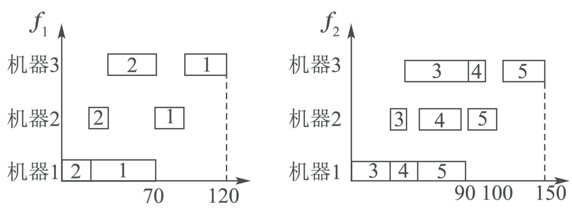


图 2 工件分配甘特图

Fig. 2 Gantt chart of job assignment

3.3 初始化种群

为保持初始种群的多样性和分散性, 本文在初始化种群时采用随机生成的方法. 在得到初始种群后评价每个解的最大完工时间和碳排放量, 进而确定非劣解并构建初始非劣解集.

3.4　贝叶斯概率模型

OEEDA的第一阶段利用贝叶斯概率模型学习非劣解中的结构信息, 并采用BEDA对解空间进行较广范围的全局搜索. 贝叶斯概率模型的结构如图3所示. 其中, 贝叶斯网络的层数为n, 每层的结点数均为n, 结点$N_{i,j}$表示优质解中第i个位置的工件为j. 从结点$N_{i,j}$到结点$N_{i+1,k}$的定向弧权重为优质解中符合该子序的数量.

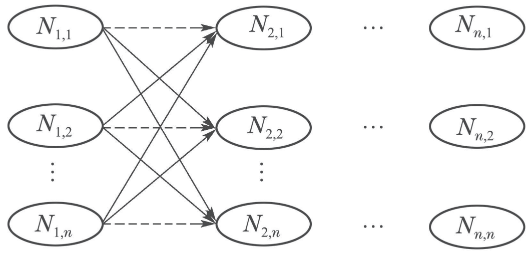


图 3　贝叶斯网络图

Fig. 3　Bayesian network diagram

在初始化贝叶斯概率模型时, 为了防止算法过早收敛与陷入局部最优, 定向弧权重设置为$1/n$. 该权重在根据优质解的工件序信息进行更新之后进行归一化. 下面通过一个实例阐明概率模型的更新过程. 设优质解的个数为20, $n=10$, 并且所有优质解的前3个位置的工件信息都符合图4的贝叶斯网络. 通过将贝叶斯网络中结点间的权重作为概率模型的更新指标, 相邻结点之间的选择概率计算过程如下所示:

$$P(N_{1,1})=\frac{7}{20},\ P(N_{1,2})=\frac{9+3}{20},\ P(N_{1,3})=\frac{1}{20},$$

$$P\left(N_{2,3}|\,N_{1,1}\right)=\frac{7}{20},\ P\left(N_{2,1}|\,N_{1,2}\right)=\frac{9}{20},$$

$$P\left(N_{2,3}|\,N_{1,2}\right)=\frac{3}{20},\ P\left(N_{2,1}|\,N_{1,3}\right)=\frac{1}{20},$$

$$P\left(N_{3,2}|\,N_{2,1}\right)=\frac{1}{20},\ P\left(N_{3,3}|\,N_{2,1}\right)=\frac{9}{20},$$

$$P\left(N_{3,1}|\,N_{2,3}\right)=\frac{3}{20},\ P\left(N_{3,2}|\,N_{2,3}\right)=\frac{7}{20}.$$

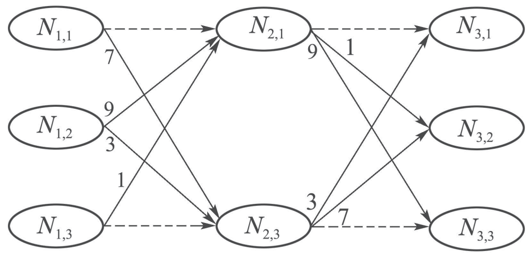


图 4　非劣解部分结构网络图

Fig. 4　Network diagram of the part structure of non-dominated solutions

本节利用非劣解集构建贝叶斯网络并初始化概率模型. 针对概率模型进行采样生成新个体, 一般选用轮盘赌策略进行采样, 但针对离散工件排序采样, 轮盘赌策略往往会重复选中同一个工件而产生不可行解, 导致浪费了大量的时间. 鉴于此, 本文改进了概率模型的更新机制, 提出了一种伴随位置实时更新部分概率矩阵的方法, 该方法能够有效保证轮盘赌策略不会选择已有的工件, 进而提高算法的执行效率.

首先利用贝叶斯网络图更新概率模型中的第1个位置的概率值并对该位置的概率值进行归一化处理, 通过轮盘赌方法选择第1个位置的工件. 基于当前选择的工件更新下一个位置的概率值, 进行归一化处理. 为了保证解的可行性, 如果更新概率时遇到已选择的工件则不再对该工件进行统计, 且概率矩阵中该工件所对应的余下位置上的概率值全部置为零. 当概率矩阵更新完待选位置上工件的概率后, 再次利用轮盘赌方法确定该位置的工件. 直至最后一个位置的工件序号被确定. 利用图4中的贝叶斯网络生成新种群中3/4的个体, 其余个体随机生成. 由贝叶斯网络生成的个体或解中部分概率更新过程如图5所示.

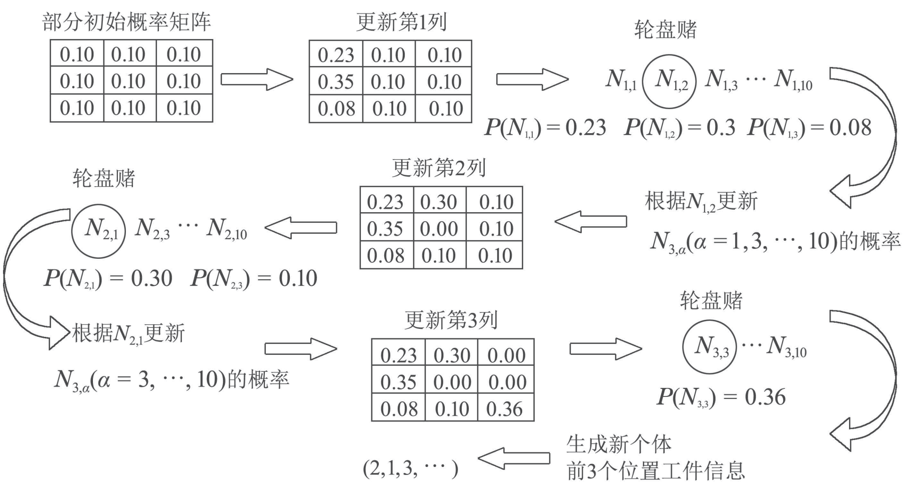


图 5　部分更新过程

Fig. 5　Part of the update process

3.5 基于序关系的四维矩阵

在贝叶斯网络对非劣解的采样和种群更新过程中, 优质解在结构上将表现出一定的相似性, 主要体现为相似的序关系, 即工件组成的相似块结构, 以及各个块在解中的位置顺序. 包含这些序关系的解将以更高的概率收敛到近似最优解. 但贝叶斯网络对序关系的学习能力有限, 难以保留和学习优质解中的工件块结构及其位置信息, 使得算法在搜索近似最优解的过程中降低鲁棒性. OEEDA第2阶段提出的OFDM可以有效提取种群中优质解在四维空间内的组合块结构信息及位置信息, 从而提升了算法在搜索过程中的鲁棒性, 有效地引导算法的搜索方向. 相同的块结构出现在解的不同位置将会对该解的目标值产生较大的影响. 因此, 该OFDM的第1个维度记录了该块结构在解中的位置信息, 剩余3个维度分别记录了该位置块结构上毗邻的3个工件序号, 以保证块结构的合理有效性. 第4节验证了所提算法在求解DFSP–LC时的效果要优于单独使用BEDA或OFDM.

首先, 定义Best_Pop(gen)为算法第gen代的优质种群或非劣解集, 种群规模为Best_popsize. OFDM的构造是通过对优质种群的所有个体的序关系采样.

$$\begin{aligned}\text{Best\_Pop(gen)}=\{&\text{Best\_}\pi^{\text{gen},1},\text{Best\_}\pi^{\text{gen},2},\cdots,\\&\text{Best\_}\pi^{\text{gen,Best\_popsize}}\},\\ \text{Best\_}\pi^{\text{gen},\omega}=[&\text{Best\_}{j_1}^{\text{gen},\omega}\ \ \text{Best\_}{j_2}^{\text{gen},\omega}\ \ \cdots\\&\text{Best\_}{j_n}^{\text{gen},\omega}],\end{aligned}$$

$\omega=1,\cdots,\text{Best_popsize}$为Best_Pop(gen)中的第$\omega$个个体. 令$\text{BM}^{\text{gen}}_{l\times n\times n\times n}$为第gen代基于位置$l(l=n-2)$的四维矩阵. 其中: $\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,y,z)(p=1,2,\cdots,l;\ x,y,z=1,2,\cdots,n)$为$\text{BM}^{\text{gen}}_{l\times n\times n\times n}$中的下标为$(p,x,y,z)$的元素. $\text{BM}^{\text{gen}}_{l\times n\times n\times n}$具体定义如下式所示:

$$\text{OneS\_BM}^{\text{gen},\omega}_{l\times n\times n\times n}(p,x,y,z)=\begin{cases}1,\ x=\text{Best\_}j^{\text{gen},w}_{p_1},\ y=\text{Best\_}j^{\text{gen},w}_{p_2},\\ \quad z=\text{Best\_}j^{\text{gen},w}_{p_3},\\ 0,\ 其他,\end{cases}\tag{11}$$
$$p=1,2,\cdots,l;\ x,y,z=1,2,\cdots,n,$$

$$\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,y,z)=\sum_{\omega=1}^{\text{Best\_popsize}}\text{OneS\_BM}^{\text{gen},\omega}_{l\times n\times n\times n}(p,x,y,z),\tag{12}$$
$$p=1,2,\cdots,l;\ x,y,z=1,2,\cdots,n,$$

$$\begin{aligned}&\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,y,1)=\\&[\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,1,1)\ \ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,2,1)\\&\cdots\ \ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,n,1)]_{l\times n\times n},\end{aligned}\tag{13}$$
$$p=1,2,\cdots,l;\ x=1,2,\cdots,n,$$

$$\begin{aligned}&\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,y)=\\&[\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,y,1)\ \ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,y,2)\\&\cdots\ \ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,y,n)]_{l\times n\times n},\end{aligned}\tag{14}$$
$$p=1,2,\cdots,l;\ x,y=1,2,\cdots n,$$

$$\begin{aligned}&\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x)=\\&[\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,1)\ \ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,2)\\&\cdots\ \ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,x,n)]_{n\times n},\end{aligned}\tag{15}$$
$$p=1,2,\cdots,l;\ x=1,2,\cdots,n,$$

$$\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p)=\begin{bmatrix}\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,1)\\ \vdots\\ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,n)\end{bmatrix}_{n\times n}=\begin{bmatrix}\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,1,1) & \cdots & \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,1,n)\\ \vdots & & \vdots\\ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,n,1) & \cdots & \text{BM}^{\text{gen}}_{l\times n\times n\times n}(p,n,n)\end{bmatrix}_{n\times n},\tag{16}$$

$$\begin{aligned}\text{BM}^{\text{gen}}_{l\times n\times n\times n}=[&\text{BM}^{\text{gen}}_{l\times n\times n\times n}(1)\ \ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(2)\\&\cdots\ \ \text{BM}^{\text{gen}}_{l\times n\times n\times n}(l)].\end{aligned}\tag{17}$$

$\text{OneS_BM}^{\text{gen},\omega}_{l\times n\times n\times n}(p,x,y,z)$在四维矩阵中记录了Best_Pop(gen)中第$\omega$个个体的第$p_1$个位置的工件$\text{Best_}j^{\text{gen},\omega}_{p_1}$为$x$, 第$p_2$个位置的工件$\text{Best_}j^{\text{gen},\omega}_{p_2}$为$y$, 第$p_3$个位置的工件$\text{Best_}j^{\text{gen},\omega}_{p_3}$为$z$的块结构出现的次数. 即表示为Best_Pop(gen)中第ω个个体基于位置p的

$$\left[x=\text{Best\_}j^{\text{gen},w}_{p_1}\ \ y=\text{Best\_}j^{\text{gen},w}_{p_2}\ \ z=\text{Best\_}j^{\text{gen},w}_{p_3}\right]$$

工件块结构. $\text{BM}^{\text{gen}}_{l\times n\times n\times n}(p)$记录了Best_Pop(gen)中所有个体的第$p$个位置的工件块结构信息. 基于序关系的四维矩阵的构建如图6所示.

$\text{BM}^{\text{gen}}_{l\times n\times n\times n}$的构建基于由BEDA获得的优质解. 当算法运行时间占比满足OFDM启动比率B_α时, 则终止BEDA, 启动OFDM在之后的每代中学习和保留当代各非劣解的工件块结构和位置信息, 并用于生产新个体以实现对问题解空间的搜索. 对于不同

的问题规模n, OEEDA将利用OFDM确定B_n个块结构分别对新种群内3/4的个体进行更新, 保证这些个体都具有相同的优质块结构信息. 块结构的选取是从四维矩阵内具有最大值的元素开始, 依次递减选取B_n个不重叠、工件不冲突的块结构. 也就是说, 这3/4的新个体中, 每个个体依据OFDM确定$3 \times B_n$个位置上的工件, 其余位置的工件由剩余工件随机确定. 这可确保搜索在具有明确引导性的同时也保持了一定的分散性. 为了避免算法陷入局部最优, 新种群中剩余1/4的个体随机生成.

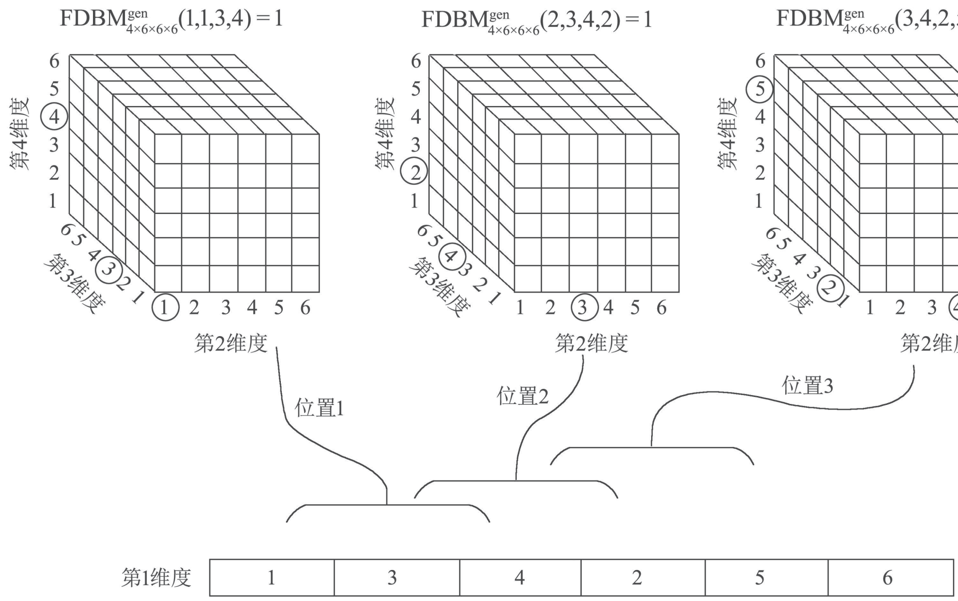


图 6 基于序关系的四维矩阵

Fig. 6 Four-dimensional matrix based on ordinal relationship

3.6 基于Interchange和Insert的局部搜索

为增加算法的搜索能力, 需对优质解的近邻区域进行较细致的局部搜索. 由于交换操作Interchange和插入操作Insert构成的解空间中解之间的距离是各种常用搜索操作或邻域中距离最小的, 有利于对解空间进行细致搜索. 因此, 对非支配解解集中的解执行基于这两种邻域的局部搜索. 具体来说, 首先对做局部搜索的个体执行1 次Interchange操作, 实现扰动. 随后对该个体依次执行3类搜索深度或次数均为40的Insert操作. 在每次执行Insert操作后, 如果新个体支配原个体则将其替换, 并用新个体作为当前个体执行后续操作, 最终输出当前个体. 各操作具体定义如下:

Interchange: 随机选择两个不同位置p_1和p_2, 交换个体p_1和p_2位置的工件实现扰动.

Solution based insert: 随机选择两个不同位置p_1和$p_2 (p_2 > p_1)$, 将p_2位置的工件前插在p_1位置.

Inter factory insert: 在最迟完工时间对应的工厂内选择加工时间最大的工件, 并将该工件插入到随机选取的其他工厂的随机位置p_1.

Intra factory insert: 在最迟完工时间对应的工厂内随机选择 2 个不同位置p_1和$p_2 (p_2 > p_1)$, 将p_2位置的工件前插在p_1位置.

4 仿真实验结果与分析

4.1 参数设置

本文所有测试问题* 的加工时间均采用100以内随机正整数按照问题规模生成. 各工厂内机器的速度档位设定为

$$S = \{1, 1.1, 1.2, 1.3, 1.4\},$$

$$PP_{j,v} = 4 \times v^2,\ SP_j = 1.$$

采用Ishibuchi等[28]提出的统计对比方法对非劣或非支配解集S_j进行评价. 该评价指标描述如下:

$$\mathrm{R\_NDS}\,(S_j) = \frac{|S_j - \{x \in S_j | \exists y \in S : y \prec x\}|}{|S_j|}, \tag{18}$$

S表示K个算法的非劣解集的集合,

$$S = S_1 \cup \cdots \cup S_K.$$

$y \prec x$表示解x被y支配. $|S_j|$表示非劣解集S_j中非劣解的个数. $\mathrm{R_NDS}\,(S_j) = 1$表示$S_j$中的解不被$S$中的任何解支配, 即比率 $\mathrm{R\_NDS}\,(S_j)$ 越大, 非劣解集

*测试问题及测试实例可在https://pan.baidu.com/s/1qkgsMUcmeNlPxelHW8wCCw下载, 若链接失效请联系通讯作者.

S_j的表现越好.

OEEDA有3个关键的参数: 种群规模NP、OFDM启动时间比率B_α、块结构个数B_n. 本文针对中等规模$f=3$采用实验设计方法(design of experiment, DOE)[29]探讨参数对算法性能的影响. 各参数均选取4个水平值, 如表2所示, 从而建立规模为$L_{16}(4^3)$的正交实验表. 算法在每组参数下均独立运行中等规模$f=3$中的11个问题规模各20次, 以$1.5\times n$秒作为各规模的终止条件, 以算法在不同参数组合下各规模的R_NDS平均值AVG作为评价指标.

表 2 参数水平

Table 2 Parameter values

参数	水平			
	1	2	3	4
NP	20	40	60	80
B_α	0.2	0.4	0.6	0.8
B_n	$\frac{n}{3}$	$\frac{n}{3-2}$	$\frac{n}{3-4}$	$\frac{n}{3-6}$

表 3 正交表和AVG

Table 3 Orthogonal array and AVG values

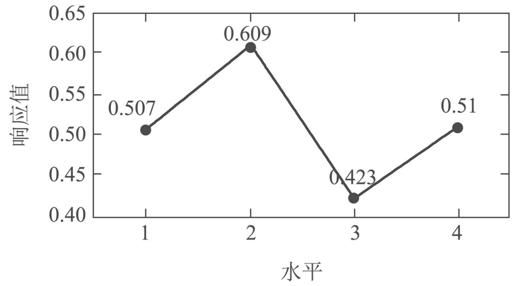
参数组合编号	水平			AVG × 10
	NP	B_α	B_n	
1	1	1	1	0.437
2	1	2	2	0.640
3	1	3	3	0.478
4	1	4	4	0.475
5	2	1	2	0.476
6	2	2	3	0.552
7	2	3	4	0.750
8	2	4	1	0.658
9	3	1	3	0.325
10	3	2	4	0.413
11	3	3	1	0.383
12	3	4	2	0.571
13	4	1	4	0.440
14	4	2	1	0.438
15	4	3	2	0.767
16	4	4	3	0.396

表 4 各参数响应值

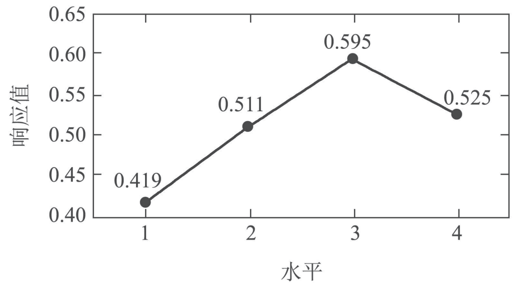
Table 4 Response values of parameters

水平	参数		
	NP	B_α	B_n
1	0.507	0.419	0.479
2	0.609	0.511	0.613
3	0.423	0.595	0.438
4	0.510	0.525	0.519

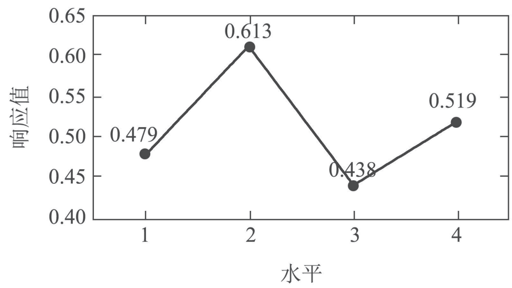
正交表和所得评价指标如表3所示. 根据正交表得到各参数的响应值如表4所示, 各参数对算法性能影响的参数响应图如图7所示.



(a) NP对算法性能的影响



(b) B_α对算法性能的影响



(c) B_n对算法性能的影响

图 7 参数响应图

Fig. 7 Response diagram of parameters

由实验结果可以发现不同的参数组合会对算法性能产生较大的影响, 为了发挥算法的最佳性能, 根据16种参数组合下的总共11个问题规模, 对每种组合下的各问题均单独运行算法20次, 共计3520次, 用于进行参数实验分析. 基于实验分析结果, 本文设定了最佳的参数组合, 即NP$=40$, $B_\alpha=0.6$, $B_n=n/3-2$. 因此, 后文将基于此参数设置开展进一步的性能测试和算法比较研究.

4.2 性能指标

在这一部分, 为了评价各算法的搜索能力, 采用了2个主要的评价指标来对每个算法的非劣解集S_j进行评价.

第1个指标如式(18)所示, R_NDS (S_j)越大, 非劣解集S_j的表现越好. 第2个指标为非劣解集S_j中

的解不被S中的任意解支配的个数. 第2个评价指标描述如下:

$$\text{NDS\_NUM}(S_j)=|S_j-\{x\in S_j|\,\exists y\in S: y\prec x\}|,\tag{19}$$

NDS_NUM (S_j)的值越大, 非劣解集S_j的表现越好. 显然, 第1个指标为占优比, 评价的是算法得到非支配解的整体质量, 第2个指标为占优个数, 评价的是算法获取非支配解的能力. 这2个指标间存在正相关关系, 相关系数为1/$|S_j|$.

4.3 实验结果

为验证所提算法全局搜索的有效性, 本文首先对不加局部搜索的OEEDA(OEEDA without local search, OEEDA–nols)、单独使用基于序关系四维矩阵的分布估计算法(OFDM based algorithm, OFDMA)、单独使用基于贝叶斯统计推断的分布估计算法(EDA based on Bayesian statistical inference, BEDA)在全局搜索下进行了性能测试, 测试结果如表5所示.

根据表5可知, OEEDA–nols的性能优于BEDA, BEDA性能优于OFDMA. 这与设计OFDMA的出发点一致, 本文提出的基于序关系的四维矩阵就是针对优质解的序关系进行有效学习和积累, 单独使用OFDMA时, 由于初始种群的随机性导致个体主要分布在问题解空间的较差区域, OFDMA无法对序关系进行有效学习. OEEDA–nols由于利用了BEDA对问题解空间进行了一定时间的搜索, 在获得优质解的基础上采用OFDMA进行序关系的学习, 有效地指导算法的搜索方向, 从而提高算法的最终性能.

表 5 算法性能验证($f=3$)

Table 5 Verification of algorithm performance ($f=3$)

$n\times m$	OFDMA		BEDA		OEEDA–nols	
	R_N	N_N	R_N	N_N	R_N	N_N
20×5	0.09	0.30	0.44	1.00	**0.64**	**2.20**
30×5	0.00	0.00	0.51	1.40	**0.62**	**2.30**
30×10	0.05	0.20	0.33	0.95	**0.73**	**1.85**
50×5	0.00	0.00	**0.62**	**1.85**	0.50	1.55
50×10	0.08	0.20	**0.58**	**1.15**	0.43	1.00
50×20	0.10	0.25	0.42	0.65	**0.63**	**1.45**
70×5	0.13	0.45	0.34	0.80	**0.64**	**2.10**
70×10	0.12	0.40	0.38	0.95	**0.76**	**2.15**
70×20	0.20	0.50	**0.45**	**0.75**	0.38	0.70
100×10	0.10	0.35	0.45	0.55	**0.63**	**1.30**
100×20	0.10	0.35	0.40	0.65	**0.65**	**1.25**
平均值	0.09	0.27	0.45	0.97	**0.63**	**1.61**

为验证OEEDA的有效性, 本文将其与BEDA, OFDMA, OEEDA–nols, 文献[28]中的算法IMMOGLS, 文献[30]中的算法GGA进行比较. 其中, IMMOGLS是一种基于随机权的有效多目标遗传进化算法, 其性能优于著名的多目标算法SPEA和NSGAII. GGA 算法是对著名多目标算法SPEAII的改进和提升. 不同问题规模的实验结果如表6所示, 该结果表面加入局部搜索后OEEDA的性能得以进一步提升, 同时OEEDA 几乎在所有问题上占优. 这验证了OEEDA求解不同规模问题的有效性.

表 6 算法比较结果(多目标)

Table 6 Comparisons of algorithms (multi-objective)

$n\times m\times f$	BEDA		OFDMA		IMMOGLS[28]		GGA[30]		OEEDA_nols		OEEDA	
	R_N	N_N	R_N	N_N	R_N	N_N	R_N	N_N	R_N	N_N	R_N	N_N
$20\times5\times2$	0.00	0.00	0.00	0.00	0.00	0.00	0.30	1.10	0.02	0.05	**0.90**	**2.40**
$20\times5\times3$	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.80	0.05	0.10	**0.58**	**1.35**
$20\times5\times4$	0.00	0.00	0.00	0.00	0.00	0.00	**0.55**	**1.30**	0.28	0.55	0.47	1.15
$30\times5\times2$	0.05	0.20	0.00	0.00	0.00	0.00	0.23	1.10	0.07	0.20	**0.90**	**2.60**
$30\times5\times3$	0.00	0.00	0.00	0.00	0.00	0.00	0.26	0.90	0.01	0.05	**0.83**	**2.25**
$30\times5\times4$	0.14	0.40	0.00	0.00	0.00	0.00	0.32	0.95	0.03	0.10	**0.72**	**1.85**
$30\times10\times2$	0.00	0.00	0.00	0.00	0.00	0.00	0.17	0.45	0.05	0.10	**0.88**	**1.60**
$30\times10\times3$	0.05	0.05	0.00	0.00	0.00	0.00	0.13	0.30	0.00	0.00	**0.95**	**2.20**
$30\times10\times4$	0.08	0.30	0.00	0.00	0.00	0.00	0.22	0.50	0.03	0.05	**0.84**	**2.20**
$50\times5\times2$	0.05	0.15	0.00	0.00	0.00	0.00	0.12	0.35	0.05	0.35	**0.90**	**2.30**

(转下页)

(接上页)

$50\times5\times3$	0.05	0.05	0.00	0.00	0.00	0.00	0.11	0.30	0.02	0.05	**1.00**	**2.60**
$50\times5\times4$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.35	0.90	**0.80**	**1.90**
$50\times10\times2$	0.05	0.05	0.02	0.10	0.00	0.00	0.15	0.25	0.05	0.15	**0.85**	**1.80**
$50\times10\times3$	0.05	0.05	0.00	0.00	0.00	0.00	0.02	0.05	0.05	0.20	**0.93**	**1.90**
$50\times10\times4$	0.10	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.30	**0.83**	**1.60**
$50\times20\times2$	0.05	0.05	0.00	0.00	0.00	0.00	0.01	0.05	0.00	0.00	**0.95**	**1.65**
$50\times20\times3$	0.05	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.10	**0.95**	**1.80**
$50\times20\times4$	0.03	0.05	0.00	0.00	0.00	0.00	0.02	0.05	0.10	0.15	**0.88**	**1.60**
$70\times5\times2$	0.10	0.25	0.00	0.00	0.00	0.00	0.42	1.65	0.00	0.00	**0.76**	**1.95**
$70\times5\times3$	0.00	0.00	0.00	0.00	0.00	0.00	0.27	1.10	0.13	0.20	**0.77**	**1.25**
$70\times5\times4$	0.00	0.00	0.00	0.00	0.00	0.00	0.45	0.70	0.00	0.00	**0.63**	**1.15**
$70\times10\times2$	0.03	0.10	0.00	0.00	0.00	0.00	0.07	0.10	0.08	0.20	**0.91**	**1.90**
$70\times10\times3$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	**0.90**	**1.60**
$70\times10\times4$	0.08	0.10	0.00	0.00	0.00	0.00	0.03	0.05	0.13	0.50	**0.90**	**1.60**
$70\times20\times2$	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.15	0.00	0.00	**0.93**	**1.40**
$70\times20\times3$	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.15	**0.95**	**1.55**
$70\times20\times4$	0.05	0.05	0.00	0.00	0.00	0.00	0.05	0.05	0.00	0.00	**0.93**	**1.55**
$100\times10\times2$	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.50	0.03	0.10	**0.83**	**1.25**
$100\times10\times3$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**	**1.60**
$100\times10\times4$	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.05	0.05	0.10	**0.97**	**1.55**
$100\times20\times2$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.15	**0.92**	**1.35**
$100\times20\times3$	0.05	0.05	0.00	0.00	0.00	0.00	0.05	0.05	0.10	0.30	**0.95**	**1.35**
$100\times20\times4$	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05	0.00	0.00	**1.00**	**1.45**
平均值	0.03	0.07	0.00	0.00	0.00	0.00	0.14	0.40	0.06	0.15	**0.78**	**1.64**

为验证OEEDA求解单目标分布式流水线调度问题的有效性, 本文将其与文献[31]中的有效EDA (effective EDA, EEDA)以及文献[15]中的混合免疫(hybrid immune algorithm, HIA)进行比较. EEDA利用二维概率模型学习相似工件块结构, 并结合学习速率对概率模型进行更新, 其性能优于著名的VND(b)和NEH(2)算法; HIA是对经典IA算法的改进, 通过将特定的交叉、变异和疫苗接种等操作应用于种群更新, 显著提升了算法的效果, 实验结果如表7所示. 实验结果表明OEEDA性能完全优于EEDA, 同时虽然OEEDA在部分小规模问题上的平均值劣于HIA. 但随着问题规模扩大, 其性能优于HIA. 具体原因在于EEDA采用二维概率矩阵, 只能保留相似工件块结构信息, 未能保留其对应的位置信息, 而HIA采用的交叉等操作难以避免工件块结构的破坏. 这使得EEDA和HIA生成新种群时引导性有所欠缺. OEEDA采用四维概率矩阵可较好避免这些问题, 有效提升了算法搜索能力. 因此, OEEDA可有效求解不同规模的多目标和单目标问题.

4.4 案例分析

为进一步验证所提算法的有效性, 将OEEDA用于求解云南某钢铁公司的钢材加工成形调度问题. 如图8所示, 该公司下属2个同类型加工厂, 分别位于昆明和玉溪, 每个工厂具有相同的加工环境, 加工原材料钢坯按照形状可分为板坯、方坯、矩形坯及圆坯, 加工过程依次为挤压、拉拔、车削、铣削、刨削和镗削共6个环节, 其中前2个环节为塑性加工, 后4个环节为切削加工. 各环节的加工机器可通过调整档位来设定加工速度. 近年来, 该公司积极推进节能减排, 在其生产调度过程中不仅考虑优化经济指标(最大完工时间), 也同时考虑优化环境指标(总碳排放量). 显然, 此钢材加工成形调度问题为典型的DFSP–LC. 目前的调度方案是由公司里有经验的调度员基于工件编号由小到大的排序, 使用某种规则确定, 最常用的规则为LCF规则.

本节采用该公司下属2个工厂加工30个钢坯(即工件)的实际生产数据作为测试实例用OEEDA运行10 s求解, 同时请工厂有经验的调度员在2 min内给出人工调度方案. OEEDA获得了调度方案S_1至S_7, 如表8和图9所示. 调度员利用LCF得到了调度方案T_1和T_2, 如表8所示. 由表8可知, S_1至S_7明显优于T_1和T_2.

表 7 算法比较结果(最大完工时间)

Table 7 Comparisons of algorithms (makespan)

$n \times m \times f$	EEDA[31]			HIA[15]			OEEDA		
	最优	最差	平均值	最优	最差	平均值	最优	最差	平均值
$20 \times 5 \times 2$	733	772	752.50	**663**	**683**	**673.00**	666	701	683.50
$20 \times 5 \times 3$	580	609	594.50	**513**	**536**	**524.50**	**513**	540	526.5
$20 \times 5 \times 4$	495	521	508.00	**451**	**470**	**460.50**	456	493	474.5
$30 \times 5 \times 2$	1038	1081	1059.50	982	**1002**	**992.00**	**976**	1016	996.00
$30 \times 5 \times 3$	789	815	802.00	**726**	**748**	**737.00**	735	759	747.00
$30 \times 5 \times 4$	658	687	672.50	**599**	**621**	**610.00**	614	636	625.00
$30 \times 10 \times 2$	1508	1572	1540.00	1414	1469	1441.50	**1401**	**1457**	**1429.00**
$30 \times 10 \times 3$	1212	1250	1231.00	1145	**1163**	**1154.00**	**1133**	1190	1161.50
$30 \times 10 \times 4$	1051	1087	1069.00	**987**	**1024**	**1005.50**	1007	1045	1026.00
$50 \times 5 \times 2$	1686	1742	1714.00	1557	1618	1587.50	**1549**	**1614**	**1581.50**
$50 \times 5 \times 3$	1239	1264	1251.50	1136	1174	1155.00	**1128**	**1163**	**1145.50**
$50 \times 5 \times 4$	995	1036	1015.50	**911**	960	935.50	915	**953**	**934.00**
$50 \times 10 \times 2$	2168	2219	2193.50	2046	2100	2073.00	**1989**	**2073**	**2031.00**
$50 \times 10 \times 3$	1646	1694	1670.00	1581	1610	1595.50	**1532**	**1601**	**1566.50**
$50 \times 10 \times 4$	1400	1438	1419.00	**1290**	1354	1322.00	**1290**	**1342**	**1316.00**
$50 \times 20 \times 2$	2883	2967	2925.00	2802	**2908**	2855.00	**2705**	2926	**2815.50**
$50 \times 20 \times 3$	2344	2400	2372.00	2243	2308	2275.50	**2208**	**2257**	**2232.50**
$50 \times 20 \times 4$	2058	2104	2081.00	1950	2020	1985.00	**1949**	**2010**	**1979.50**
$70 \times 5 \times 2$	2135	2192	2163.50	2015	2083	2049.00	**1985**	**2073**	**2029.00**
$70 \times 5 \times 3$	1546	1579	1562.50	1434	1491	1462.50	**1422**	**1457**	**1439.50**
$70 \times 5 \times 4$	1232	1272	1252.00	1131	1184	1157.50	**1123**	**1175**	**1149.00**
$70 \times 10 \times 2$	2743	2789	2766.00	2579	2705	2642.00	**2502**	**2644**	**2573.00**
$70 \times 10 \times 3$	2045	2117	2081.00	1910	1976	1943.00	**1888**	**1936**	**1912.00**
$70 \times 10 \times 4$	1712	1764	1738.00	1584	**1644**	1614.00	**1566**	1650	**1608.00**
$70 \times 20 \times 2$	3484	3564	3524.00	3404	**3539**	3471.50	**3249**	3558	**3403.50**
$70 \times 20 \times 3$	2753	2807	2780.00	2628	2818	2723.00	**2572**	**2665**	**2618.50**
$70 \times 20 \times 4$	2356	2417	2386.50	2273	2371	2322.00	**2224**	**2305**	**2264.50**
$100 \times 10 \times 2$	3655	3759	3707.00	3545	3661	3603.00	**3409**	**3651**	**3530.00**
$100 \times 10 \times 3$	2692	2870	2781.00	2546	2655	2600.50	**2544**	**2571**	**2557.50**
$100 \times 10 \times 4$	2208	2257	2232.50	2108	2154	2131.00	**2072**	**2142**	**2107.00**
$100 \times 20 \times 2$	4606	4723	4664.50	4526	**4639**	4582.50	**4321**	4687	**4504.00**
$100 \times 20 \times 3$	3532	3642	3587.00	3415	3541	3478.00	**3336**	**3497**	**3416.50**
$100 \times 20 \times 4$	3008	3059	3033.50	2908	2999	2953.50	**2821**	**2939**	**2880.00**
平均值	1945.15	2002.09	1973.58	1848.55	1916.00	1882.27	**1812.12**	**1900.79**	**1856.45**

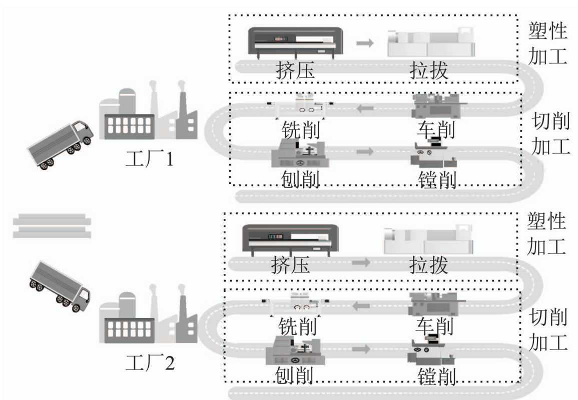


图 8 金属成形工艺流程图

Fig. 8 The process flow chart of metal forming

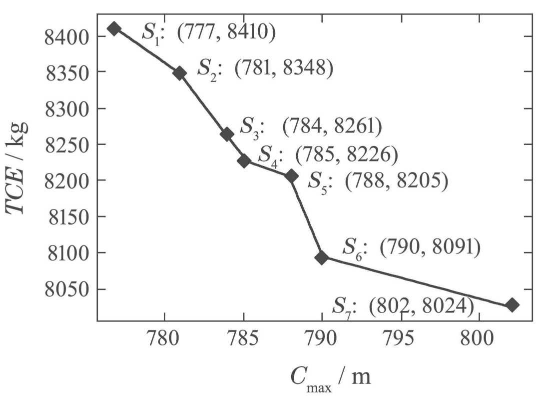


图 9 OEEDA目标值对比图

Fig. 9 Comparison chart of the target values of OEEDA

表 8 调度方案

Table 8 Scheduling scheme

调度方案	工件调度排序																	$C_{\max}$	TCE
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
T_1-f_1	17	27	22	30	3	6	23	8	18	28	24	16	12	20	2	19	—	850	9248
T_1-f_2	13	7	15	1	11	9	26	14	5	4	29	10	25	21	—	—	—		
T_2-f_1	8	5	20	23	28	25	3	16	26	7	10	1	17	15	12	2	—	843	9340
T_2-f_2	21	27	30	18	24	29	9	6	11	19	14	13	22	4	—	—	—		
S_1-f_1	8	12	21	18	1	3	22	6	11	24	27	4	26	—	—	—	—	777	8410
S_1-f_2	16	29	30	5	23	17	7	9	28	15	25	14	20	13	19	2	10		
S_2-f_1	13	22	1	29	10	11	16	24	6	17	4	20	5	21	28	—	—	781	8348
S_2-f_2	8	9	25	18	15	30	7	27	12	23	2	3	14	26	19	—	—		
S_3-f_1	22	8	29	6	3	9	15	1	13	17	24	12	11	28	26	19	2	784	8261
S_3-f_2	16	27	5	25	23	30	18	4	7	20	21	10	14	—	—	—	—		
S_4-f_1	30	22	5	18	25	11	16	23	8	20	4	3	21	10	—	—	—	785	8226
S_4-f_2	26	24	29	1	19	7	17	9	14	6	13	28	27	15	12	2	—		
S_5-f_1	13	24	14	1	11	27	5	9	15	17	8	25	12	20	10	16	21	788	8205
S_5-f_2	28	23	6	18	2	7	30	19	22	29	3	4	26	—	—	—	—		
S_6-f_1	13	8	9	24	30	7	25	18	29	22	5	14	3	20	19	10	—	790	8091
S_6-f_2	23	1	4	28	11	16	17	2	27	15	21	6	26	12	—	—	—		
S_7-f_1	17	24	14	1	11	27	5	9	15	13	8	25	12	20	3	16	21	802	8024
S_7-f_2	28	23	6	18	2	7	30	19	22	29	4	26	10	—	—	—	—		

5 结论

本文提出了一种基于序关系的增强EDA(enhanced EDA based on ordered relationship, OEEDA), 用于求解低碳分布式流水线调度问题(low carbon scheduling of distributed flow shop problem, DFSP–LC). OEEDA分为2阶段, 可分别实现对问题解空间较广和较细范围的全局搜索, 同时每个阶段也融入了较为细致的局部搜索. 具体来说, 在OEEDA前一阶段, 利用基于贝叶斯统计推断的EDA(EDA based on Bayesian statistical inference, BEDA)对解空间进行较广范围的全局搜索, 将发现的优质非劣解保存于非劣解集中. 在OEEDA后一阶段, 提出了基于序关系的四维矩阵(four-dimensional matrix based on ordered relationship, OFDM)用于明确保存解内部工件块结构信息及工件块在解中的位置信息(即序关系信息), 然后设计了在解内部部分位置固定工件块的新个体采样机制, 可对问题解空间执行较细范围的全局搜索. 在上述两个阶段中, 均引入了基于解、工厂间、工厂内的3种不同Insert融合的局部搜索策略, 对每个阶段全局搜索得到的解空间优质区域进行较为细致的搜索, 使得算法在全局和局部搜索之间达到良好平衡. OEEDA的关键参数采用实验设计方法进行了确定. 其有效性通过在不同规模测试问题上的仿真实验和算法比较得到了验证. 下一步工作将针对不确定分布式制造问题, 研究基于混合EDA的有效求解方法.

参考文献:

[1] WANG L, SHEN W. *Process Planning and Scheduling for Distributed Manufacturing*. London: Springer, 2007.

[2] FANG K, UHAN N, ZHAO F, et al. A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *Journal of Manufacturing Systems*, 2011, 30(4): 234 – 240.

[3] GILLES M, MEHMET B, JANET T. Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research*, 2007, 45(18/19): 4247 – 4271.

[4] YAN J, LI L, ZHAO F, et al. A multi–level optimization approach for energy-efficient flexible flow shop scheduling. *Journal of Cleaner Production*, 2016, 137: 1543 – 1552.

[5] TANG D, DAI M, SALIDO M A, et al. Energy–efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. *Computers in Industry*, 2016, 81(C): 82 – 95.

[6] ZHANG R, CHIONG R. Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *Journal of Cleaner Production*, 2015, 112: 3361 – 3375.

[7] DING J Y, SONG S, WU C. Carbon-efficient scheduling of flow shops by multi–objective optimization. *European Journal of Operational Research*, 2015, 248(3): 758 – 771.

[8] AI Ziyi, LEI Deming. A novel shuffled frog leaping algorithm for low carbon flexible job shop scheduling. *Control Theory & Applications*,2017, 34(10): 1361 – 1368.
(艾子义, 雷德明. 基于新型蛙跳算法的低碳柔性作业车间调度. 控制理论与应用, 2017, 34(10): 1361 – 1368.)

[9] LU C, GAO L, LI X, et al. Energy–efficient permutation flow shop scheduling problem using a hybrid multi–objective backtracking search algorithm. *Journal of Cleaner Production*, 2017, 144: 228 – 238.

[10] ZHENG H Y, WANG L. Reduction of carbon emissions and project makespan by a Pareto–based estimation of distribution algorithm. *International Journal of Production Economics*, 2015, 164: 421 – 432.

[11] DAI M, TANG D, GIRET A, et al. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, 2013, 29(5): 418 – 429.

[12] NADERI B, RUIZ R. A scatter search algorithm for the distributed permutation flowshop scheduling problem. *European Journal of Operational Research*, 2014, 239(2): 323 – 334.

[13] GAO J, CHEN R. An NEH-based heuristic algorithm for distributed permutation flowshop scheduling problems. *Scientific Research and Essays*, 2011, 6(14): 3094 – 3100.

[14] NADERI B, RUIZ R. The distributed permutation flowshop scheduling problem. *Computers & Operations Research*, 2010, 37(4): 754 – 768.

[15] XU Y, WANG L, WANG S, et al. An effective hybrid immune algorithm for solving the distributed permutation flow-shop scheduling problem. *Engineering Optimization*, 2014, 46(9): 1269 – 1283.

[16] RIFAI A P, NGUYEN H T, DAWAL S Z M. Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling. *Applied Soft Computing*, 2016, 40(C): 42 – 57.

[17] DENG J, WANG L, WU C, et al. A competitive memetic algorithm for carbon–efficient scheduling of distributed flow–shop. *International Conference on Intelligent Computing*. Lanzhou: ICIC, 2016, 9771: 476 – 488.

[18] BALUJA S. *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning*. Pittsburgh: Carnegie Mellon University, 1994.

[19] MUHLENBEIN H, PAASS G. From recombination of genes to the estimation of distributions I: Binary parameters. *Lecture Notes in Computer Science*, 1996, 1141(1): 178 – 187.

[20] LARRAANAGA P, LOZANO J A. Estimation of distribution algorithms: a new tool for evolutionary computation. *Kluwer Academic Publishers*, 2001, 64(5): 454 – 468.

[21] JARBOUI B, EDDALY M, SIARRY P. An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. *Computers & Operations Research*, 2009, 36(9): 2638 – 2646.

[22] SALHI A, ZHANG Q. An estimation of distribution algorithm with guided mutation for a complex flow shop scheduling problem. *Genetic & Evolutionary Computation Conference*, 2007, 14(14): 570 – 576.

[23] WANG F, TANG Q H, RAO Y Q, et al. Efficient estimation of distribution for flexible hybrid flow shop scheduling. *Zidonghua Xuebao/Acta Automatica Sinica*, 2017, 43(2): 280 – 293.

[24] WANG S Y, WANG L, LIU M, et al. Estimation of distribution algorithm for solving hybrid flow-shop scheduling problem with identical parallel machine. *International Journal of Advanced Manufacturing Technology*, 2013, 68(9/12): 2043 – 2056.

[25] QIAN B, LI Z C, HU R. A copula-based hybrid estimation of distribution algorithm for M-machine reentrant permutation flow-shop scheduling problem. *Applied Soft Computing*, 2017, 46(1): 139 – 149.

[26] RAWAT V P S, CUSAN M, DESHPANDE A, et al. Two new robust genetic algorithms for the flowshop scheduling problem. *Omega*, 2006, 34(5): 461 – 476.

[27] CHEN S F, QIAN B, LIU B, et al. Bayesian statistical inference-based estimation of distribution algorithm for the re-entrant job-shop scheduling problem with sequence-dependent setup times. *International Conference on Intelligent Computing*. Taiyuan: ICIC, 2014, 8589: 686 – 696.

[28] 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.

[29] MONTGOMERY D C. *Design and Analysis of Experiments*. Hoboken: John Wiley and Sons, 2005.

[30] MAY G, STAHL B, TAISCH M, et al. Multi-objective genetic algorithm for energy-efficient job shop scheduling. *International Journal of Production Research*, 2015, 53(23): 7071 – 7089.

[31] WANG S Y, WANG L, LIU M, et al. An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. *International Journal of Production Economics*, 2013, 145(1): 387 – 396.

作者简介:

杨晓林 硕士研究生, 目前研究方向为智能算法与优化调度, E-mail: xiaolin_yang436@163.com;

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;

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;

吴丽萍 讲师, 硕士生导师, 目前研究方向为优化调度理论与方法、智能优化方法, E-mail: 554350754@qq.com.