一种求解随机有限缓冲区流水线调度的混合差分进化算法

胡 蓉 [1] 钱 斌 [1]

摘 要 针对随机有限缓冲区流水线调度问题 (Flow shop scheduling problem, FSSP), 提出混合差分进化 (Differential evolution, DE) 算法 OHTDE, 用来最小化提前/拖后指标和最小化总体完成时间指标. OHTDE 将 DE 和最优计算量分配 (Optimal computing budget allocation, OCBA) 技术以及假设检验 (Hypothesis test, HT) 有效结合. DE 用于执行全局搜索和局部搜索; OCBA 用于对有限计算量进行合理分配, 从而保证优质解得到较多仿真计算量, 提高了在噪声环境下获得优质解的置信度; HT 用于在统计意义上比较解的性能, 从而一定程度上避免在解空间相近区域进行重复搜索. 进而, 对由 OCBA 和 HT 确定的优质解执行一种特殊的交叉操作, 加强 DE 的局部开发能力. 同时也采用有限马氏链理论对 OHTDE 的随机收敛性进行了分析. 仿真实验和算法比较验证了算法的有效性和鲁棒性.

关键词 随机车间调度, 差分进化, 最优计算量分配, 假设检验
中图分类号 TP18

A Hybrid Differential Evolution Algorithm for Stochastic Flow Shop Scheduling with Limited Buffers

HU Rong[1] QIAN Bin[1]

Abstract A hybrid differential evolution (DE) algorithm, namely OHTDE, is proposed to solve the stochastic flow shop scheduling problem (FSSP) with limited buffers between consecutive machines. Two different criteria are considered. The first is to minimize a general earliness/tardiness cost function and the second is to minimize the total completion time. The proposed algorithm is a hybrid of DE and two techniques: the optimal computing budget allocation (OCBA) technique and hypothesis test (HT). DE is used to execute both global and local search. OCBA is utilized to reasonably allocate a limited sampling budget for every solution, assigning more for better individuals to improve the confidence level of getting good solutions under noise environment. HT is adopted to perform a statistical comparison on solutions′ performances so that the repeated search on the similar region of the solution space can be avoided to some extent. Moreover, a special crossover, which is performed on some good solutions identified by OCBA and HT, is designed to enhance the capability of local search. Furthermore, the stochastic convergence property of OHTDE is analyzed by using the theory of finite Markov chain. Experimental results and comparison show the effectiveness and robustness of the proposed OHTDE.

Key words Stochastic flow shop scheduling, differential evolution, optimal computing budget allocation, hypothesis test

收稿日期 2008-07-10 收修改稿日期 2009-07-05
Received July 10, 2008; in revised form July 5, 2009
国家自然科学基金 (60904081), 云南省应用基础研究基金 (2009ZC015X), 昆明理工大学引进人才基金 (KKZ3200903021) 资助
Supported by National Natural Science Foundation of China (60904081), Applied Basic Research Foundation of Yunnan Province (2009ZC015X), and Talent Introduction Foundation of Kunming University of Science and Technology (KKZ3200903021)
1. 昆明理工大学信息工程与自动化学院自动化系 昆明 650051
1. Department of Automation, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650051
DOI: 10.3724/SP.J.1004.2009.01580

在化工、钢铁和制药等实际生产系统中, 缓冲区 (即中间存储设备) 的大小通常是有限的. 近年来, 随着 Just-in-time (JIT) 制造和 Kanban 控制的兴起与发展, 对有限缓冲区流水线调度问题 (Flow shop scheduling problem, FSSP) 的研究具有重要实际意义, 因此越来越受到重视[1−3]. 本文研究的是相邻两台机器间存在有限缓冲区的 FSSP. 在这种情况下, 工件在当前机器上加工完成后, 如果下游缓冲区已满, 则该工件将停滞在这台机器上, 直到下游缓冲区非满[3]. 特别地, 当所有缓冲区大小均为 0 时, 这类问题称为 Blocking FSSP[4]. Papadimitriou 等[5] 证明了两台机器以上的有限缓冲区 FSSP 是一个 NP-hard 问题.

在实际生产系统中, 往往存在着大量不确定性, 因此, 相比确定性调度问题, 开发求解随机调度问题的有效算法在工程和理论领域均有更重要的价值. 目前, 已有学者对加工时间不确定的调度问题进行研究[6−10], 但尚无学者对加工时间随机分布的有限缓冲区 FSSP 进行研究. 因此, 本文将研究这一类问题. 对于求解随机优化问题, 如何对解进行选择和评价是关键. 最优计算量分配 (Optimal computing budget allocation, OCBA) 是 Chen 等[11] 提出的在随机环境下选择和评价解的有效算法, 也是最主要的方法之一[12]. 根据不同的评价标准, OCBA 已发展出了几种变形算法[13−15]. 文献 [15] 的研究表明, Chen 等[11] 提出的以问题满意解的正确选择概率为评价标准的算法是目前最有效的算法之一. 因此, 本文采用文献 [11] 中的 OCBA 对随机问题的解进行性能评估.

近年来, 差分进化算法 (Differential evolution, DE)[16−17] 一直是研究的热点. DE 是基于群体智能理论的优化算法, 通过种群中个体间的合作与竞争产生的群体智能指导优化搜索. DE 具有概念简单、易于实现和收敛性好等优点, 如今已在很多领域得到了广泛应用[18]. 然而, DE 采用实数编码, 故对 DE 的研究主要集中于连续优化领域. 目前, DE 在生产调度领域的研究屈指可数. DE 作为一种有效和简单的并行搜索算法, 对其进行应用和理论研究具有重要的工程和理论价值.

截至 2008 年 5 月, 尚无其他学者基于 DE 进行随机调度问题的研究. OCBA 和假设检验 (Hypothesis test, HT) 是处理随机优化问题的有效方法. Wang 等分别提出了基于 OCBA 的遗传算法[9] 和基于 HT 的遗传算法[10], 用于求解加工时间随机分布的 FSSP. Pan 等[19] 首次结合 OCBA 技术和 HT 策略, 提出了求解随机连续函数优化问题的几种混合粒子群优化算法 (Particle swarm optimization, PSO), 其中同时使用 OCBA 和 HT 的混合算法取得最优性能. 但是针对复杂的随机组合优化问题, 如何结合 OCBA 和 HT 的优势来设计有效求解算法仍然处于空白状态. 本文提出 OHTDE, 将 DE 和 OCBA 以及 HT 结合来有效处理加工时间随机分布的有限缓冲区 FSSP 问题. 文献 [20] 的研究表明, FSSP 的解空间是一种峡谷型 (Big valley) 地貌 (Landscape), 局部最优和全局最优解大都集中在峡谷底部. 文献 [21] 表明 DE 的全局搜索能在整个 FSSP 问题解空间中进行有效搜索, 从而发现性能良好的区域. 但对于随机 FSSP, 峡谷底部总是在变化, 这意味着只使用 DE 很难找到足够好的解. 因此, 在 OHTDE 中, DE 用于执行全局搜索和局部搜索; OCBA 用于对有限计算量进行分配, 从而保证优质解得到较多仿真计算量, 提高了在噪声环境下获得优质解的置信度; 同时, 采用 HT 在统计意义上比较解的性能, 从而一定程度上避免在解空间相近区域进行重复搜索. 进而, 对由 OCBA 和 HT 确定

的优质解执行一种特殊的交叉操作, 加强 DE 的局部开发能力. 测试结果验证了算法的有效性.

1 随机有限缓冲区 FSSP

有限缓冲区 FSSP 描述如下[3]: 考虑 n 个工件 $J=\{1,2,\cdots,n\}$ 在 m 台机器 $M=\{1,2,\cdots,m\}$ 上的加工过程, 该 n 个工件将依相同次序在机器 $1\sim m$ 上进行加工, 且每台机器上工件的加工顺序相同. 在每两台相邻的机器 i 和 $(i-1)$ 之间存在缓冲区, 并且所有工件在缓冲区中均服从先入先出规则. 求解目标为发现工件的一个排列, 使得某一给定指标达到最优. 对于实际生产问题, 通常要考虑库存费用和生产效率等因素, 故本文对提前/延迟 (Earliness/Tardiness) 指标和总体完成时间 (Total completion time) 指标进行优化.

实际生产过程中往往存在不确定性. 设工件 j 在机器 i 上进行加工的时间 $p_{i,j}$ 服从均匀分布 $U((1-\eta)P_{i,j},(1+\eta)P_{i,j})$, 其中 $P_{i,j}$ 为算例的标准数据, η 代表噪声幅度. 令 $E/T(\boldsymbol{\pi},\boldsymbol{\xi})$ 和 $C_T(\boldsymbol{\pi},\boldsymbol{\xi})$ 分别为工件排序 $\boldsymbol{\pi}$ 在噪声 $\boldsymbol{\xi}$ 下采样的提前/延迟时间和总体完成时间, Π 为所有解的集合. 对于 E/T 指标, 随机有限缓冲区 FSSP 的求解目标为找到一个工件排列 $\boldsymbol{\pi}^*$, 使得

$$\boldsymbol{\pi}^*=\arg\min_{\boldsymbol{\pi}\in\Pi}\{\mathrm{E}[E/T(\boldsymbol{\pi},\boldsymbol{\xi})]\} \tag{1}$$

其中, $\mathrm{E}[E/T(\boldsymbol{\pi},\boldsymbol{\xi})]$ 为 $E/T(\boldsymbol{\pi},\boldsymbol{\xi})$ 的数学期望.

类似地, 对于 C_T 指标, 随机有限缓冲区 FSSP 的求解目标为找到 $\boldsymbol{\pi}^*$, 使得

$$\boldsymbol{\pi}^*=\arg\min_{\boldsymbol{\pi}\in\Pi}\{\mathrm{E}[C_T(\boldsymbol{\pi},\boldsymbol{\xi})]\} \tag{2}$$

其中, $\mathrm{E}[C_T(\boldsymbol{\pi},\boldsymbol{\xi})]$ 为 $C_T(\boldsymbol{\pi},\boldsymbol{\xi})$ 的数学期望.

2 DE、OCBA 和 HT

2.1 DE 简介

DE[16−17] 在一定程度上是一种自适应、自组织、自学习的迭代寻优过程, 采用实数编码、基于差分的变异操作和一对一的竞争生存策略, 避免了复杂的遗传操作, 同时它可以根据当前种群的个体间差异动态调整其搜索策略, 具有较好的全局寻优能力和内在的隐并行性. 在算法的每一代中, 对当前个体进行交叉和变异来得到新个体, 然后通过选择操作来得到新种群. 根据各类问题上的应用经验, DE 已发展出了 10 余种变形算法, 具体详见 http://www.icsi.berkeley.edu/storn/code.html.

2.2 OCBA 简介[11, 22]

序优化 (Ordinal optimation, OO) 是解决复杂离散事件动态系统优化问题的一种实用优化技术, 其主要思想包括序比值容易确定和目标软化[23]. Dai[24] 证明了在序比较下问题满意解的正确选择概率 ($P(CS)$) 指数收敛. Chen 等[11] 在序优化框架下提出了一种估计 $P(CS)$ 的方法, 并进一步给出了可提高 $P(CS)$ 的最优计算量分配技术 OCBA, 用于给真正好的解分配更多的计算量.

2.3 HT 简介[10, 22]

假设检验是统计推断中的一种重要方法, 它首先对所研究的总体作某种假设, 然后根据样本对所提假设作出接受或拒绝的判断. 对于随机优化问题, 每个解 $\boldsymbol{X}_i$ 的性能估计 $\hat{J}(\boldsymbol{X}_i)$ 可视为一个独立分布的随机变量, 当随机采样次数 $N_i\to\infty$ 时, 根据大数定理和中心极限定理, $\hat{J}(\boldsymbol{X}_i)$ 服从正态分布. 另外, 在理论上一般可假设特定问题所有解的性能方差相同[25]. 因此, 对于任意两个解 $\boldsymbol{X}_1$ 和 $\boldsymbol{X}_2$ 性能的数学期望 μ_1 和 μ_2, 可用 t 检验法检验是否相等, 其中零假设 H_0 为 $\mu_1=\mu_2$, 备择假设 H_1 为 $\mu_1\neq\mu_2$. 若零假设成立, 则认为 $\boldsymbol{X}_1$ 和 $\boldsymbol{X}_2$ 的性能在统计上不存在明显差异.

3 随机有限缓冲区 FSSP 的 OHTDE 算法

3.1 解的表达

由于标准 DE 种群中个体是以连续变量表示的, 因而标准 DE 不能直接用于 FSSP. 由此, 采用 LOV (Largest order value) 规则[21] 实现从个体 $\boldsymbol{X}_i=[x_{i,1},x_{i,2},\cdots,x_{i,n}]$ 到离散加工排序 $\boldsymbol{\pi}_i=[\pi_{i,1},\pi_{i,2},\cdots,\pi_{i,n}]$ 的转换, 从而将 DE 在实数空间的搜索映射为调度问题解空间的搜索. 通过转化, 保证了调度的可行性, 同时无需修改 DE 的进化操作方式. 在 LOV 规则中, $\boldsymbol{X}_i=[x_{i,1},x_{i,2},\cdots,x_{i,n}]$ 首先按降序排列得到中间序列 $\boldsymbol{\varphi}_i=[\varphi_{i,1},\varphi_{i,2},\cdots,\varphi_{i,n}]$. 然后, 工件的排序由下式计算获得:

$$\pi_{i,\varphi_{i,k}}=k \tag{3}$$

3.2 随机 FSSP 处理技术

为了能处理随机 FSSP, 我们采用了如下技术.

1) 轮盘赌规则确保 OCBA 发挥作用

在 Chen 等[11] 提出的 OCBA 中, T 为总体计算量或仿真次数, n_0 为初始对每个解平均分配的计算量, N_i 为分配给第 i 个解的仿真计算量, Δ 为每次增加的仿真量, $\delta_{b,i}=\bar{J}_b-\bar{J}_i$, 其中 $\bar{J}_b\leq\min_{i=1,\cdots,K}\bar{J}_i$, K 为可行解总数. 对随机 FSSP, $\delta_{b,i}$ 和 $\delta_{b,j}$ $(i\neq j)$ 之间的差别通常要远比 σ_i 和 σ_j 之间的大. 这会导致 $\sum_{i=1}^{K}\max\{0,(N_i^{l+1}-N_i^l)\}\gg\Delta$, 从而使得 OCBA 算法[11] 的步骤 3 中每次实际增加的仿真评价次数明显大于 Δ. 在这一情况下, OCBA 算法中步骤 2 和步骤 3 的重复执行次数将明显小于 $(T-n_0)/\Delta$, 这样 OCBA 的机制就不能很好发挥作用. 因此, 在 OHTDE 中, 采用轮盘赌规则来进行每次计算量 Δ 的分配, 也就是每次只分配 Δ. 具体来说, 需要把 OCBA 算法的步骤 3 做如下修改.

步骤 3.1. 计算 DE 种群中每个个体 $\boldsymbol{X}_i$ $(i=1,2,\cdots,K)$ 的分配概率 $P(\boldsymbol{X}_i)$, 即 $P(\boldsymbol{X}_i)=N_i^{l+1}/\sum_{i=1}^{K}N_i^{l+1}$;

步骤 3.2. 使用轮盘赌规则将 Δ 分配给各个个体 $\boldsymbol{X}_i$, 然后令 $l=l+1$, 返回步骤 1.

2) 特殊的交叉操作提高 OHTDE 的局部搜索能力

对于随机 FSSP, 解空间的形状和峡谷底部区域在每次评价采样时均会发生变化. 这意味着只使用 DE 的交叉和变异很难对随机意义下的解空间进行有效搜索. 由 Schiavinotto 和 Stützle 的研究[26] 可知, Interchange 邻域的直径在常用邻域中是最小之一, 故 Interchange 邻域构成的解空间中解之间的距离也是最小之一. 这表明在相同计算量下, Interchange 邻域能进行更高效和彻底的搜索. 因此, 在 OHTDE 中, 采用了如下特殊的交叉操作, 即对一些质量差的个体并不进行 DE 的交叉和变异, 而是用由 OCBA 和 HT 以高置信度确定出的优质个体对其进行替换, 然后对这些好的个体执行基于 Interchange 邻域的操作. 调度问题一般都符合近邻假设, 即优质解的附近存在着多个优质解. 因此, 这种特殊的交叉操作能引导整个搜索在优质解附近的区域做细致搜索, 从而提高 DE 的局部搜索的效率.

3.3 OHTDE

在 OHTDE 中, 基于 DE 的全局搜索采用 DE/rand-to-best/1/exp 机制, rand-to-best 表示选择当前种群的最好个体作为基准个体, 因此种群中的个体在搜索过程中能分享最好个体的信息, 这在一定程度上考虑了局部搜索.

为有效求解随机调度问题, OHTDE 采用 DE 在解空间进行全局搜索和一定程度的局部搜索. OHTDE 在种群的个体评价时使用了 OCBA 技术, 将有限评价次数进行合理分配, 从而在随机环境下以高置信度确定种群中的优质解, 这样就能引导搜索集中在有效区域进行; 同时, OHTDE 在 DE 选择阶段采用了 HT 技术, 使得实验种群中的个体与当前种群中的对应个体在性能上存在明显差异时才可被接受, 从而避免在解空间相近区域进行搜索, 保证了算法搜索区域的广度; 另外, OHTDE 也利用一种特殊的交叉操作来加强算法的局部搜索能力, 使得算法在全局和局部搜索上达到平衡.

OHTDE 的算法具体步骤如下所示.

步骤 0. 令 $Pop(t)$ 为有 PS 个个体的种群的第 t 代, $W_Pop(t)$ 保存需要进行特殊交叉操作的个体, $\boldsymbol{X}_i(t)$ 为 $Pop(t)$ 中维数为 N $(N = n)$ 的第 i 个个体, $x_{i,k}(t)$ 为 $\boldsymbol{X}_i(t)$ 中的第 k 维分量, $lower(x_{i,k})$ 为 $x_{i,k}(t)$ 取值的下界, $upper(x_{i,k})$ 为 $x_{i,k}(t)$ 取值的上界, $Tmp_Pop(t)$ 为有 PS 个个体的临时种群的第 t 代, $\boldsymbol{Tmp_X}_i(t)$ 为 $Tmp_Pop(t)$ 中的第 i 个个体, tmp_k 为实验个体 $\boldsymbol{tmp}$ 中的第 k 维分量, CR 为交叉概率, F 为比例因子, random$(0,1)$ 为区间 [0, 1] 上的随机值, τ 为零假设 H_0 的拒绝域值.

步骤 1. 设定 N, $PS \geq 3$, $CR \in [0,1]$, $lower(x_{i,k})$, $upper(x_{i,k})$, $k = 1, \cdots, N$.

步骤 2. 种群初始化.

步骤 2.1. $x_{i,k}(0) = lower(x_{i,k}) + \text{random}(0,1) \times (upper(x_{i,k}) - lower(x_{i,k}))$, $k = 1, \cdots, N$, $i = 1, \cdots, PS$.

步骤 2.2. 应用 OCBA 对 $Pop(0)$ 中个体进行评价, 设定 $W_Pop(0)$ 为空集.

步骤 3. 设定 $t = 1$, 从 $Pop(0)$ 中选择最优个体 $\boldsymbol{X}_{\text{best}}(0)$, $\boldsymbol{bestit} = \boldsymbol{X}_{\text{best}(0)}$.

步骤 4. 进化阶段 (步骤 4 ~ 9), 设定 $i = 1$.

步骤 5. 变异和交叉阶段.

步骤 5.1. 若 $\boldsymbol{X}_i(t-1)$ 属于 $W_Pop(t-1)$, 则执行特殊的交叉操作.

步骤 5.1.1. 从 $Pop(t-1)$ 中最优的 5 个个体中随机选择一个, 作为 $\boldsymbol{tmp}$.

步骤 5.1.2. 随机生成 u 和 v, $u \neq v$, 对 $\boldsymbol{tmp}$ 的第 u 维和第 v 维变量执行 Interchange 操作 (注: 由 LOV 规则映射到问题解空间也是执行 1 次 Interchange 搜索).

步骤 5.1.3. 令 $\boldsymbol{Tmp_X}_i(t) = \boldsymbol{tmp}$, 转到步骤 6.

步骤 5.2. 执行 DE 的交叉和变异, 令 $\boldsymbol{tmp} = \boldsymbol{X}_i(t-1)$, $L = 0$, 随机选择 $r1, r2 \in (1, \cdots, PS)$ 且 $r1 \neq r2 \neq i$, 随机选择 $k \in (1, \cdots, N)$.

步骤 5.2.1. $tmp_k = tmp_k + F \times (bestit_k - tmp_k) + F \times (x_{r1,k}(t-1) - x_{r2,k}(t-1))$.

步骤 5.2.2. 令 $k = (k+1) \mod N$ 且 $L = L + 1$.

步骤 5.2.3. 若 $(\text{random}(0,1) < CR)$ 且 $(L < N)$, 则转到步骤 5.2.1.

步骤 5.2.4. 令 $\boldsymbol{Tmp_X}_i(t) = \boldsymbol{tmp}$.

步骤 6. 令 $i = i + 1$, 若 $i \leq PS$ 则转到步骤 5.

步骤 7. 应用 OCBA 对 $Tmp_Pop(t)$ 中的个体进行评价.

步骤 8. 选择阶段, 应用假设检验对 $\boldsymbol{Tmp_X}_i(t)$ 和 $\boldsymbol{X}_i(t-1)$ 进行选择, $i = 1, \cdots, PS$.

步骤 8.1. 若零假设不成立, 且 $\bar{J}(\boldsymbol{X}_i(t-1)) - \bar{J}(\boldsymbol{Tmp_X}_i(t)) \geq \tau$, 则 $b_replace = \text{true}$; 否则 $b_replace = \text{false}$.

步骤 8.2. 若 $b_replace = \text{true}$, 则 $\boldsymbol{X}_i(t) = \boldsymbol{Tmp}_X_i(t)$, 否则 $\boldsymbol{X}_i(t) = \boldsymbol{X}_i(t-1)$.

步骤 8.3. 若 $b_replace = \text{true}$, 且 $\bar{J}(\boldsymbol{Tmp_X}_i(t)) < \bar{J}(\boldsymbol{bestit})$, 则 $\boldsymbol{bestit} = \boldsymbol{Tmp_X}_i(t)$.

步骤 8.4. 若 $b_replace = \text{false}$, 且 $\boldsymbol{X}_i(t-1)$ 属于 $Pop(t-1)$ 中最差 10% 的个体, 则将 $\boldsymbol{X}_i(t-1)$ 加入 $W_Pop(t)$.

步骤 9. 令 $t = t + 1$, 若 $t \leq t_\max$ (即最大进化代数), 则转到步骤 4.

步骤 10. 输出 $\boldsymbol{bestit}$ 及其对应的目标值.

OHTDE 的算法流程如图 1 所示.

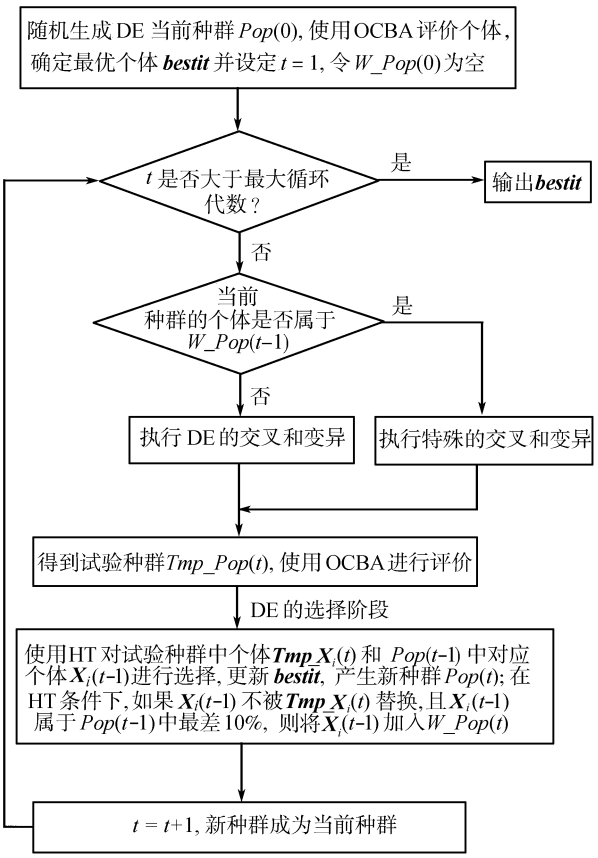

图 1 OHTDE 流程图

Fig. 1 The framework of the OHTDE

从上述算法可见, OHTDE 不仅利用 DE 本身具有的分散搜索能力在解空间进行全局搜索和一定程度的局部搜索, 同时还采用了基于随机问题解空间特性的处理技术来提高算法的局部搜索能力. 由于全局和局部搜索能力的平衡, 有助于 OHTDE 获得随机 FSSP 较好的解.

3.4 算法收敛性分析

定理 1. 如果 $P(CS) = 1$, OHTDE 以概率 1 收敛到全

局最优.

证明. 当 $P(CS)=1$ 时, 必有 $T\to\infty$, 且 $\tau=0$, 这时采样均值等于期望值, 因此 OCBA 和基于 HT 的 DE 选择操作均能作出完全正确的选择. 这样随机优化转化为确定性优化.

对于确定性优化问题, OHTDE 可以描述为一个齐次有限状态的马氏链. 令 Ω 为 OHTDE 中个体的状态空间 (LOV 规则使得 DE 个体的状态由实数向量转换为离散排序), $|\Omega|$ 为 Ω 的大小, G 为整个种群的状态空间, $|G|$ 为 G 的大小. 设 OHTDE 中个体的维数为 n, 种群大小为 PS, 则 $|\Omega|=n!$ 和 $|G|=|\Omega|^{PS}$.

令 OTHDE 中整个种群的一步转移概率矩阵为 $P=(p_{ij})_{|G|\times|G|}$, 其中 p_{ij} 为从种群状态 i 到种群状态 j 的转移概率. 在 OHTDE 的变异和交叉阶段, 实验个体 $\boldsymbol{tmp}$ 的状态能以一定概率转移到解空间中的任意个体的状态, 即可在整个解空间进行搜索; 在 OHTDE 的选择阶段, 临时种群中个体 $\boldsymbol{Tmp_X}_i(t)$ 的状态只有优于 $Pop(t-1)$ 中的个体 $\boldsymbol{X}_i(t-1)$ 的状态才能被保留. 这意味着任意种群状态 i 只能转移到更好的状态 j 或保持不变. 因此, OHTDE 搜索过程的数学模型是一个带吸收态的齐次有限状态马氏链, 其中, 整个状态空间中最好的状态为吸收态, 其他状态都是瞬态.

将所有状态按评价指标从好到差排列 (越好的状态在 P 中排列越靠前), 这样 p_{11} 就表示了从最好状态 (即种群中所有个体均为全局最优) 到最好状态的转移概率, 其他依次类推. 因此, P 是一个下三角阵:

$$P=\begin{bmatrix} p_{11} & & & \\ p_{21} & p_{22} & & \\ \vdots & & \ddots & \\ p_{|G|1} & p_{|G|2} & \cdots & p_{|G||G|} \end{bmatrix}=\begin{bmatrix} C & 0 \\ R & T \end{bmatrix} \tag{4}$$

其中, $p_{ij}=\begin{cases} \geq 0,\ i\geq j \\ >0,\ j=1 \\ =0,\ i<j \end{cases}$, $C=p_{11}=1$, $R=\begin{bmatrix} p_{21} \\ \vdots \\ p_{|G|1} \end{bmatrix}>0$, $T\neq 0$. 由于 P 是一个可约随机矩阵, 根据可约随机矩阵的性质有: $P^{\infty}=\begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}$. 因此, 当进化代数 $t\to\infty$ 时, OHTDE 的种群将以概率 1 收敛到最优种群 (即每个个体对应的指标值最小). □

注 1. 以上证明假设不同的个体状态对应不同指标值. 当存在不同个体状态对应相同指标值时, 令 Ω 和 G 分别为 OHTDE 中个体指标值和整个种群指标值的状态空间, 采用上述证明可得同样结论.

虽然实际运用时 $P(CS)=1$ 难以满足, 但是 OCBA 通过合理分配有限仿真量提高了 $P(CS)$ 的收敛速度; 同时特殊的交叉操作通过引导搜索到存在优质解的区域进行搜索, 进一步加快了算法的收敛速度. 因此, OHTDE 在保证高置信度地获得较好解的同时, 也保证了较快的收敛速度.

4 仿真实验与算法比较

4.1 实验设置

本节将通过数值仿真验证 OHTDE 的优化性能. 数值仿真通常采用标准 FSSP 测试问题, 即 Benchmark 问题. 本节选用 Reeves[27] 设计的 11 个经典测试问题 Rec05, Rec11, Rec15, Rec25, Rec29, $\cdots$, Rec41. 这些问题在大量相关文献中被用于算法性能测试. 每个问题中工件的实际加工时间服从均匀分布 $U((1-\eta)P_{i,j},(1+\eta)P_{i,j})$, 其中 $P_{i,j}$ 为算例中的标准数据, η 为噪声幅度.

其中每个工件的期望完成时间 (计算 E/T 指标时需要) 采用如下步骤生成:

步骤 1. 对于每个问题 i, 随机生成一组工件排列;

步骤 2. 计算步骤 1 得到排列中每个工件的加工完成时间;

步骤 3. 设定每个工件的期望完成时间为

$$d_{i,j}=c_{i,j}+\text{random}\left[-\frac{C_i^*}{10},\frac{C_i^*}{10}\right] \tag{5}$$

其中, $d_{i,j}$ 为工件 j 在问题 i 上的期望完成时间, $c_{i,j}$ 为工件 j 在问题 i 上的加工完成时间, C_i^* 为问题 i 的最优最大完成时间 Makespan, random$[-C_i^*/10,C_i^*/10]$ 为在区间 $[-C_i^*/10,C_i^*/10]$ 上的随机数.

为了分析 OCBA、HT 和特殊的交叉操作在 OHTDE 中的作用, 采用如下基于 OHTDE 的变种进行比较和分析.

1) PDE: OHTDE 中不使用 OCBA、HT 和特殊的交叉.

2) ODE: OHTDE 中不使用 HT 和特殊的交叉.

3) HTDE: OHTDE 中不使用 OCBA, 每个解均分配 20 次仿真进行评价.

具体细节如下:

a) OHTDE 中不使用 OCBA: 在 OHTDE 的步骤 2.2 和步骤 7 中, 每个解均分配 20 次仿真进行评价;

b) OHTDE 中不使用 HT: 在 OHTDE 的步骤 8 中, 用普通的 DE 选择操作替换基于 HT 的选择操作;

c) OHTDE 中不使用特殊的交叉操作: 不执行 OHTDE 中的步骤 5.1.

PDE、ODE、HTDE 和 OHTDE 均采用 DE 的交叉和变异来对解空间进行搜索. HTDE 和 OHTDE 在此基础上采用特殊的交叉来进一步提高算法局部搜索能力.

另外, 也采用遗传算法 (Genetic algorithm, GA) 进行性能比较. GA 实现采用工件置换编码、随机初始种群、两点交叉、插入式 (Insert) 变异和轮盘赌方式选择, 保优过程用新旧种群中最好个体替换新种群中最差个体. GA 参数设置为: 种群规模为 100, 交叉概率为 1, 变异概率为 0.05, 每个个体均评价 20 次.

所有相关程序均使用 Delphi 7.0 编写, 运行环境为 Pentium IV 3 GHz, RAM 1 G, 操作系统为 Windows XP. OHTDE 的参数设置为: 种群规模 $PS=100$, 缩放比例因子 $F=0.7$, 交叉概率 $CR=0.1$, $T=2\,000$, $n_0=10$, $\Delta=20$, $\eta=0.1$, 假设检验中使用的分位数 $\alpha=0.05$, 提前交货和拖后交货的惩罚系数均为 0.1, 所有比较算法均运行 300 代, 并且对每个问题均独立运行 20 次.

4.2 仿真实验与算法比较

首先, 采用 E/T 指标对缓冲区大小为 0 和 2 的问题进行测试. 测试结果如表 1 所示, AEV 表示算法所得解 $\boldsymbol{\pi}$ 在测试问题上的平均期望 E/T 值 (零噪声时的平均 E/T 值). AEV 可反映各算法的统计性能. 图 2 显示了 OHTDE 在求解 Rec21 (缓冲区大小为 2, OHTDE 的步骤 7) 时的第 50 代进化中 OCBA 对 $Temp_Pop(t)$ 中各个体的计算

量分配情况. 图 2 的纵轴表示评价次数, 横轴从左到右为 $Temp_Pop(t)$ 的个体根据评价值从好到坏的排列.

表 1 GA, PDE, ODE, HTDE 和 OHTDE 在 E/T 指标下的比较结果

Table 1 Comparison of GA, PDE, ODE, HTDE, and OHTDE when criterion is E/T

Problem	n, m	GA	PDE	ODE	HTDE	OHTDE
$/buffer$		AEV	AEV	AEV	AEV	AEV
Rec05/0	20, 5	488.1	136.1	136.1	**135.9**	136.1
Rec11/0	20, 10	507.4	168.8	168.7	168.6	**168.5**
Rec15/0	20, 15	627.8	201.7	**199.7**	202.3	201.3
Rec25/0	30, 10	1 462.4	272.1	274.5	260.8	**260.1**
Rec29/0	30, 15	1 441.5	387.1	378.0	364.2	**356.8**
Rec31/0	50, 10	4 160.8	1 097.9	1 066.5	929.0	**860.5**
Rec33/0	50, 10	4 325.0	1 250.7	1 263.1	1 078.9	**1 024.2**
Rec35/0	50, 10	4 170.5	1 036.5	1 046.5	903.6	**852.4**
Rec37/0	75, 20	1 0512.0	3 398.7	3 343.9	2 640.6	**2 480.6**
Rec39/0	75, 20	10 750.3	3 775.2	3 838.7	3 156.8	**2 917.4**
Rec41/0	75, 20	10 906.2	4 127.8	4 048.7	3 401.4	**3 319.1**
Rec05/2	20, 5	460.8	82.0	**81.8**	82.0	81.9
Rec11/2	20, 10	455.2	73.3	**72.7**	72.8	**72.7**
Rec15/2	20, 15	507.2	106.8	**106.5**	107.6	106.7
Rec25/2	30, 10	1 243.9	153.3	153.4	148.6	**148.1**
Rec29/2	30, 15	1 332.8	169.7	162.8	158.6	**155.6**
Rec31/2	50, 10	3 689.0	607.4	599.3	480.4	**464.1**
Rec33/2	50, 10	3 801.2	550.0	554.8	445.1	**437.5**
Rec35/2	50, 10	3 778.8	601.3	612.1	482.3	**461.1**
Rec37/2	75, 20	9 064.0	2 326.4	2 312.1	1 750.1	**1 704.9**
Rec39/2	75, 20	9 064.9	2 267.2	2 277.6	1 685.2	**1 524.6**
Rec41/2	75, 20	9 305.5	2 298.9	2 332.1	1 687.1	**1 626.7**
Average		4 184.3	1 140.4	1 137.7	924.6	**880.0**

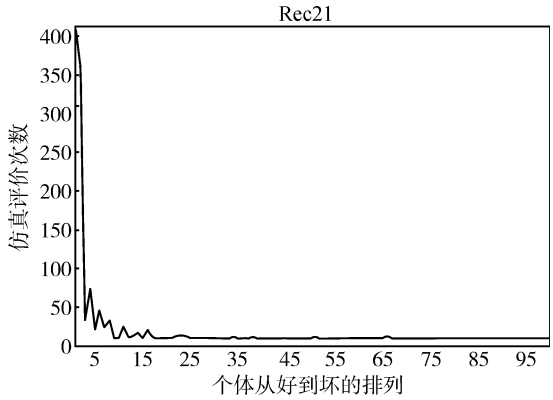

图 2 OCBA 在 OHTDE 中的计算量分配

Fig. 2 Computing budget allocation of OCBA in OHTDE

由表 1 可知, PDE 的 AEV 指标均比 GA 的小, 这表明对于随机 FSSP, PDE 具有更强的搜索能力. ODE 的 AEV 指标在平均性能上比 PDE 的小, 而 HTDE 和 OHTDE 的 AEV 指标几乎均比 PDE 的小, 这说明在 DE 算法中加入 OCBA、HT 和特殊的交叉均能提高算法性能. OHTDE 的 AEV 指标在绝大部分问题上均优于 ODE 和 HTDE, 这表明在 DE 中同时加入 OCBA、HT 和特殊的交叉能使算法更加有效. HTDE 和 OHTDE 的性能优于 PDE 和 ODE, 这验证了特殊的交叉的作用. 另外, OHTDE 的性能优于 HTDE, 这意味着 OCBA 有助于引导搜索到真正的优良区域进行, 进而使得 OHTDE 的性能获得改进. 从图 2 可见, OCBA 可根据解的性能将有限计算量智能地分配给种群中的个体, 使得性能较好的个体获得较多评价次数. 随着解性能的逐渐变差, 被分配的评价次数也逐渐减少. 对于 E/T 指标和 C_T 指标下的各问题, 均有和图 2 相似的结果.

其次, 采用 C_T 指标对缓冲区大小为 0 和 2 的问题进行测试, 测试结果见表 2. AEV 表示算法所得解 $\boldsymbol{\pi}$ 在测试问题上的平均期望 C_T 值 (零噪声时的平均 C_T 值). 图 3 显示了用 GA、PDE 和 OHTDE 对 Rec35 ($buffer = 2$) 独立运行 20 次时最优个体的平均期望 C_T 值从第 0 代至第 300 代的变化过程. 图 4 显示了用 GA、PDE 和 OHTDE 对 Rec35 ($buffer = 2$) 独立运行 20 次时种群平均期望 C_T 值从第 0 代至第 300 代的变化过程.

表 2 GA, PDE, ODE, HTDE 和 OHTDE 在 C_T 指标下的比较结果

Table 2 Comparison of GA, PDE, ODE, HTDE, and OHTDE when criterion is C_T

Problem	GA	PDE	ODE	HTDE	OHTDE
$/buffer$	AEV	AEV	AEV	AEV	AEV
Rec05/0	17 679.4	16 848.5	16 809.7	**16 706.3**	16 717.3
Rec11/0	21 726.5	20 430.9	20 420.1	**20 364.2**	20 367.7
Rec15/0	30 165.8	29 211.6	29 229.4	29 096.9	**29 080.0**
Rec25/0	59 593.7	56 616.8	56 506.4	56 008.2	**55 889.3**
Rec29/0	54 651.8	50 734.7	50 543.4	50 193.4	**49 841.0**
Rec31/0	116 493.8	109 163.4	108 511.3	105 853.8	**104 902.9**
Rec33/0	119 750.7	114 486.5	113 919.7	112 199.6	**110 320.3**
Rec35/0	119 359.6	112 180.4	111 447.0	108 656.9	**107 740.6**
Rec37/0	300 297.4	291 190.9	289 404.1	283 863.6	**282 314.3**
Rec39/0	304 095.5	294 367.4	291 716.0	285 138.1	**283 071.5**
Rec41/0	303 971.4	295 795.1	294 499.5	288 852.7	**285 997.8**
Rec05/2	15 985.4	15 512.8	15 421.4	15 389.8	**15 379.7**
Rec11/2	20 894.2	19 575.8	19 532.6	19 484.4	**19 461.0**
Rec15/2	29 139.5	27 914.1	27 896.6	27 951.9	**27 867.0**
Rec25/2	55 446.7	52 697.3	52 674.3	52 575.8	**52 083.9**
Rec29/2	51 538.9	47 527.9	47 277.1	47 197.8	**47 029.1**
Rec31/2	101 720.6	93 867.6	93 322.1	92 243.8	**91 844.5**
Rec33/2	102 328.8	96 770.4	96 413.1	95 085.6	**94 673.2**
Rec35/2	103 635.1	96 598.0	96 051.8	94 996.0	**94 864.3**
Rec37/2	266 230.2	255 893.8	255 670.9	250 891.5	**250 477.3**
Rec39/2	268 634.9	257 367.1	256 488.3	251 946.9	**249 578.0**
Rec41/2	270 620.5	257 423.1	256 371.9	**251 462.7**	251 923.5
Average	124 270.9	118 735.2	118 187.6	116 189.1	**115 519.3**

由表 2 可知, OHTDE 的性能在总体上仍然明显优于其他三种算法, 这验证了 OHTDE 的有效性和鲁棒性. 另外, 表 2 中 ODE 的 AEV 指标几乎均比 PDE 的小, 这说明了 OCBA 的作用. 由图 3 可知, PDE 的收敛速度明显快于 GA, 这说明 DE 本身的搜索能力优于 GA; 而 OHTDE 的收敛速

度明显快于 PDE, 这表明 OCBA、HT 和特殊交叉能有效提高算法的收敛速度. 由图 4 可知, 随着运行代数的增加, 各算法对应的整个种群平均 C_T 值逐渐减小, 这说明整个种群逐渐向着解空间期望峡谷底部 (零噪声时的峡谷底部) 移动, 从而引导整个搜索趋向存在优质解的区域. 同时, 图 4 也显示了 OHTDE 的整个种群收敛速度明显快于 GA 和 PDE 的种群收敛速度, 这说明 OHTDE 能引导整体搜索更快地到达期望峡谷底部, 从而具有更好的搜索效率. 对于 E/T 指标和 C_T 指标下的各问题, 均有类似图 3 和图 4 的结果.

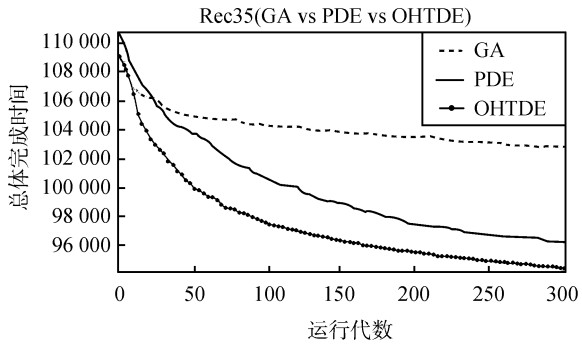

图 3　GA、PDE 和 OHTDE 中最优个体的平均期望 C_T 值随进化代数的变化

Fig. 3　The mean expected C_T of the best individual in different generations of GA, PDE, and OHTDE

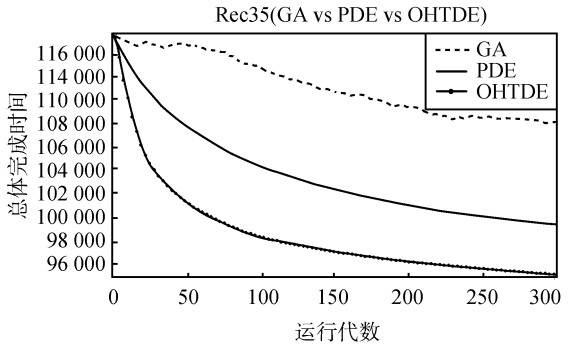

图 4　GA、PDE 和 OHTDE 中种群平均期望 C_T 值随进化代数的变化

Fig. 4　The mean expected C_T of the population in different generations of GA, PDE, and OHTDE

结合表 1、表 2、图 3 和图 4 可知, OHTDE 能快速地取得较好解, 是一种求解随机有限缓冲区 FSSP 的有效算法.

5　结论和展望

本文提出一种融合 DE、OCBA 和 HT 的算法 OHTDE, 用来求解随机有限缓冲区 FSSP. OHTDE 采用 OCBA 智能地将有限仿真评价次数分配给种群中的个体, 使得优良个体获得较多评价次数, 进而以高置信度获得优良个体; 采用 HT 来避免在解空间相近区域的重复搜索, 并保留优良个体; 同时采用特殊的交叉操作引导搜索在优良个体的邻域进行. 这些技术使得 OHTDE 在求解随机调度问题时能做到全局搜索和局部搜索的均衡, 从而有效处理了这一类问题. 仿真实验验证了 OHTDE 是一种有效鲁棒的算法. 下一步工作将主要研究更有效的搜索操作和策略, 从而进一步提高 OHTDE 的搜索效率, 并将本文算法推广应用于其他随机调度问题.

References

1 Tang L X, Xuan H. Lagrangian relaxation algorithms for real-time hybrid flowshop scheduling with finite intermediate buffers. *Journal of the Operational Research Society*, 2006, **57**(3): 316−324

2 Wang L, Zhang L, Zheng D Z. An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. *Computers and Operations Research*, 2006, **33**(10): 2960−2971

3 Qian B, Wang L, Huang D X, Wang W L, Wang X. An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers. *Computers and Operations Research*, 2009, **36**(1): 209−233

4 Grabowski J, Pempera J. The permutation flow shop problem with blocking: a tabu search approach. *Omega*, 2007, **35**(3): 302−311

5 Papadimitriou C H, Kanellakis P C. Flow shop scheduling with limited temporary storage. *Journal of the Association Computing Machine*, 1980, **27**(3): 533−549

6 Beraldi P, Ghiani G, Guerriero E, Grieco A. Scenario-based planning for lot-sizing and scheduling with uncertain processing times. *International Journal of Production Economics*, 2006, **101**(1): 140−149

7 Janak S L, Lin X X, Floudas C A. A new robust optimization approach for scheduling under uncertainty II: uncertainty with known probability distribution. *Computers and Chemical Engineering*, 2007, **31**(3): 171−195

8 Xia Y, Chen B T, Yue J F. Job sequencing and due date assignment in a single machine shop with uncertain processing times. *European Journal of Operational Research*, 2008, **184**(1): 63−75

9 Wang L, Zhang L, Zheng D Z. Genetic ordinal optimisation for stochastic flow shop scheduling. *International Journal of Advanced Manufacturing Technology*, 2005, **27**(1-2): 166−173

10 Wang L, Zhang L, Zheng D Z. A class of hypothesistest-based genetic algorithm for flow shop scheduling with stochastic processing time. *International Journal of Advanced Manufacturing Technology*, 2005, **25**(11-12): 1157−1163

11 Chen C H, Lin J W, Yucesan E, Chick S E. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems*, 2000, **10**(3): 251−270

12 Branke J, Chick S E, Schmidt C. Selecting a selection procedure. *Management Science*, 2007, **53**(12): 1916−1932

13 Chen C H, He D H. Intelligent simulation for alternatives comparison and application to air traffic management. *Journal of Systems Science and Systems Engineering*, 2005, **14**(1): 37−51

14 Chen E J, Kelton W D. Sequential selection procedures: using sample means to improve efficiency. *European Journal of Operational Research*, 2005, **166**(1): 133−153

15 He D H, Chick S E, Chen C H. Opportunity cost and OCBA selection procedures in ordinal optimization for a fixed number of alternative systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2007, **37**(5): 951−961

16 Storn R, Price K. Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, **11**(4): 341−359

17 Feoktistov V. *Differential Evolution: In Search of Solutions*. Berlin: Springer, 2006

18 Zhou Yan-Ping, Gu Xing-Sheng. Development of differential evolution algorithm. *Control and Instruments in Chemical Industry*, 2007, **34**(3): 1−5
(周艳平, 顾幸生. 差分进化算法研究进展. 化工自动化及仪表, 2007, **34**(3): 1−5)

19 Pan H, Wang L, Liu B. Particle swarm optimization for function optimization in noisy environment. *Applied Mathematics and Computation*, 2006, **181**(2): 908−919

20 Nowicki E, Smutnicki C. Some aspects of scatter search in the flow-shop problem. *European Journal of Operational Research*, 2006, **169**(2): 654−666

21 Qian B, Wang L, Hu R, Wang W L, Huang D X, Wang X. A hybrid differential evolution method for permutation flowshop scheduling. *International Journal of Advanced Manufacturing Technology*, 2008, **38**(7-8): 757−777

22 Wang Ling, Liu Bo. *Particle Swarm Optimization and Scheduling Algorithms*. Beijing: Tsinghua University Press, 2008
(王凌, 刘波. 微粒群优化与调度算法. 北京: 清华大学出版社, 2008)

23 Deng M, Ho Y C. Iterative ordinal optimization and its applications. In: Proceedings of the 36th IEEE Conference on Decision and Control. San Diego, USA: IEEE, 1997. 3562−3567

24 Dai L. Convergence properties of ordinal comparison in the simulation of discrete event dynamic systems. *Journal of Optimization Theory and Applications*, 1996, **91**(2): 363−388

25 Liu X L. *Introduction to Statistics Theory*. Beijing: Tsinghua University Press, 1998

26 Schiavinotto T, Stützle T. A review of metrics on permutations for search landscape analysis. *Computers and Operations Research*, 2007, **34**(10): 3143−3153

27 Reeves C R. A genetic algorithm for flowshop sequencing. *Computers and Operations Research*, 1995, **22**(1): 5−13

胡 蓉 昆明理工大学副教授. 主要研究方向为机器学习、生产计划与调度. 本文通信作者. E-mail: ronghu@vip.163.com
(**HU Rong** Associate professor at Kunming University of Science and Technology. Her research interest covers machine learning, production planning and scheduling. Corresponding author of this paper.)

钱 斌 博士, 昆明理工大学讲师. 主要研究方向为复杂生产过程调度理论与方法. E-mail: bin.qian@vip.163.com
(**QIAN Bin** Ph. D., lecturer at Kunming University of Science and Technology. His research interest covers scheduling theory and algorithms for complex production process.)