

求解置换流水线调度问题的混合离散果蝇算法

郑晓龙, 王凌[†], 王圣尧

(清华大学自动化系, 北京 100084)

摘要: 针对置换流水线调度问题, 提出了一种新颖的混合离散果蝇算法. 算法每一代进化包括 4 个搜索阶段: 嗅觉搜索、视觉搜索、协作进化和退火过程. 在嗅觉搜索阶段, 采用插入方式生成邻域解; 在视觉搜索阶段, 选择最优邻域解更新个体; 在协作进化阶段, 基于果蝇个体间的差分信息产生引导个体; 在退火操作阶段, 以一定概率接受最优引导个体从而更新种群. 同时, 通过试验设计方法对算法参数设置进行了分析, 并确定了合适的参数组合. 最后, 通过基于标准测试集的仿真结果和算法比较验证了所提算法的有效性和鲁棒性.

关键词: 置换流水车间调度; 离散果蝇算法; 协作进化; 混合算法

中图分类号: TP18 文献标识码: A

A hybrid discrete fruit fly optimization algorithm for solving permutation flow-shop scheduling problem

ZHENG Xiao-long, WANG Ling[†], WANG Sheng-yao

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

Abstract: To solve the permutation flow-shop scheduling problem (PFSP), we propose a novel hybrid discrete fruit fly optimization algorithm (HDFOA). Each generation of evolution in the algorithm contains four search stages: smell-based search, vision-based search, co-evolutionary search and annealing procedure. In the smell-based search stage, an insertion operator is adopted to produce neighbors. In the vision-based search stage, the individuals are replaced by their best neighbors. In the co-evolutionary search stage, the guiding individuals are produced based on the differential information among fruit flies. In the annealing procedure, the best guiding fruit flies are accepted according to certain acceptance probabilities for updating the population. Moreover, the effect from parameter setting is analyzed by using the experiment design method, and a combination of suitable parameter values is determined. Finally, simulation results and comparisons based on the benchmark testing sets demonstrate the effectiveness and robustness of the proposed algorithm.

Key words: permutation flow-shop scheduling; discrete fruit fly optimization algorithm; co-evolution; hybrid algorithm

1 引言(Introduction)

置换流水线调度问题(permutation flow-shop scheduling problem, PFSP)^[1]是许多实际生产调度问题的简化模型, 已被证明为NP-hard问题^[2], 其研究具有重要的理论意义和应用价值. PFSP的求解方法主要包括精确算法、启发式方法和智能算法. 精确算法^[3]计算复杂度大, 仅适合于求解小规模问题. 启发式方法根据问题特性, 能在较短时间内构造出解, 但难以保证解的质量. 随着计算智能的发展, 求解PFSP的智能算法得到广泛研究, 如遗传算法^[4]、模拟退火^[5]、禁忌搜索^[6-8]、微粒群优化^[9]、差分进化^[10]和蛙跳算法^[11]等.

果蝇算法(fruit fly optimization algorithm, FOA)是最近提出的一种群体进化算法^[12]. 不同于原有进化算法, 果蝇算法操作简单、原理易懂、易于实现, 具有较

强的局部搜索能力. 目前, FOA已在金融^[12]、电力负荷预测^[13]、物流服务^[14]、参数整定^[15]等连续优化问题上得到了研究与应用. 然而, FOA在离散优化问题上至今仍少有研究. 本文针对PFSP提出一种混合离散果蝇算法(hybrid discrete FOA, HDFOA). 针对问题的最优解被多个局部极小解包围的特点, 设计了利用全局信息的果蝇协作机制, 并结合模拟退火的概率突跳机制增强全局搜索. 同时, 通过试验设计方法对参数设置进行分析, 并通过仿真测试与算法比较验证所提算法的有效性和鲁棒性.

2 问题描述(Problem statement)

PFSP通常可描述为^[1]: n 个工件在 m 台机器上加工, 每个工件在各机器上加工顺序相同, 且每台机器上各工件的加工顺序也相同. 每个工件在每台机器上

只加工一次, 每台机器一次在某一时刻只能加工一个工件, 各工件在各机器上所需的加工时间已知, 期望获得一个可行调度使得所有工件的最大完成时间(makespan)最小.

令 $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ 为所有工件的一个排序, $p_{\pi_i, j}$ 和 $C_{\pi_i, j}$ 分别为 π_i 在机器 j 上的加工时间和完成时间, 则最小化最大完成时间的PFSP的数学描述如下^[1]:

$$\begin{aligned} \min : & C_{\max}(\pi), \\ \text{s.t.} & \begin{cases} C_{\max}(\pi) \geq C_{\pi_i, j}, i = 2, 3, \dots, n; \\ j = 2, 3, \dots, m, \\ C_{\pi_1, 1} = p_{\pi_1, 1}, \\ C_{\pi_i, 1} = C_{\pi_{i-1}, 1} + p_{\pi_i, 1}, i = 2, 3, \dots, n, \\ C_{\pi_1, j} = C_{\pi_1, j-1} + p_{\pi_1, j}, j = 2, 3, \dots, m, \\ C_{\pi_i, j} = \max\{C_{\pi_{i-1}, j}, C_{\pi_i, j-1}\} + p_{\pi_i, j}, \\ i = 2, 3, \dots, n; j = 2, 3, \dots, m, \end{cases} \end{aligned} \quad (1)$$

其中 $C_{\max}(\pi)$ 为最大完成时间.

3 果蝇算法(FOA)

果蝇算法是一种基于果蝇觅食行为而提出的群智能优化算法. 算法将寻找优化解的过程模拟为果蝇种群的觅食过程, 基于果蝇觅食行为中对嗅觉和视觉的运用来设计搜索操作, 通过迭代实现对果蝇种群中心位置的优化, 进而获得问题优化解. 原始果蝇算法的流程如下^[12]:

步骤 1 初始化种群中心位置.

步骤 2 嗅觉搜索. 根据种群中心位置随机产生 NP 个邻域解.

步骤 3 评价个体. 计算每个个体的评价价值.

步骤 4 视觉搜索. 选择最优邻域解, 替换更新种群中心位置.

步骤 5 判断终止准则是否满足. 是, 则输出最优解; 否则, 转至步骤2.

4 求解PFSP的HDFOA(HDFOA for PFSP)

HDFOA不同于原始FOA: 首先, 采用置换编码来表征种群中的每个解; 其次, 根据PFSP的先验知识, 应用NAWAZ-ENSCORE-HAM(NEH)启发式方法^[16]和随机方法初始化种群; 第三, 在嗅觉搜索阶段采用优化后的插入操作产生邻域解; 最后, 设计了一种利用全局信息的协作进化机制, 并采用概率突跳机制增强全局搜索.

4.1 编码(Encoding)

种群中的每一个个体对应问题的一个解, HDFOA中个体采用长度为 n 的置换编码, 即所有工件的无重复排列. 工件号在序列中的位置表示其在各机器上的加工顺序. 以一个4个工件的问题为例, 个体(2, 1, 4, 3)表示工件2最先加工, 随后依次是工件1和工件4, 工件3最后加工.

4.2 种群初始化(Population initialization)

为了兼顾初始种群的质量和多样性, 初始种群中10%的个体基于NEH启发式方法^[16]产生, 剩余90%的个体通过随机方式生成.

4.3 嗅觉和视觉搜索(Smell and vision-based search)

影响FOA优化性能的核心是如何生成嗅觉搜索阶段的邻域个体. 对于PFSP, 插入操作是最有效的邻域解生成方式^[4]. 因此, HDFOA的嗅觉搜索过程采用插入操作 $\text{insert}(r, k)$, 即随机选择 r 位置的工件插入到 k 位置的工件后, 如图1所示.

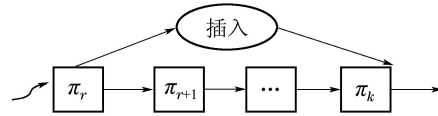


图1 插入操作 $\text{insert}(r, k)$

Fig. 1 Insert operator $\text{insert}(r, k)$

为了提高邻域个体的质量, 对于随机选择的位置 r 的工件, 将其插入到使该个体makespan最小的位置 k^* . 按此方式, 在嗅觉搜索阶段, 每个个体产生 SN 个邻域个体. 在视觉搜索阶段, 种群中每个个体将自身位置更新为其最优邻域个体. 基于上述设计, HDFOA的嗅觉和视觉搜索过程的伪代码实现如下:

嗅觉视觉搜索过程:

Procedure

```
FOR  $i = 1$  to  $NP$  //果蝇  $Ff_i$ 
  FOR  $j = 1$  to  $SN$  //果蝇  $nFf_j$ 
    随机选择位置  $r$ ;
     $\text{insert}(r, k^*)$  生成邻域个体  $nFf_j$ ;
  END FOR
  选择最优邻域个体  $BnFf$ ;
  果蝇  $Ff_i$  更新为  $BnFf$ ;
END FOR
```

4.4 协作机制(Co-evolution scheme)

果蝇种群在觅食过程中, 果蝇个体通过学习种群中其他个体的位置信息来调整自身的位置, 从而实现种群内的协作, 如此充分利用食物源位置信息, 使得种群最终逼近食物源. 为了提高算法的全局搜索能力, 模拟果蝇种群中的协作行为, 在此设计了果蝇种群内一种利用个体间差分信息的协作机制. 对于果蝇 Ff_i , 随机选择种群中不同的两个个体 Ff_{r_1} 和 Ff_{r_2} , 果蝇 Ff_i 以协作参与度 F 利用 Ff_{r_1} , Ff_{r_2} 的差分信息, 得到引导个体. 具体的实现方式如下:

$$Ff'_i = Ff_i \oplus F \otimes (Ff_{r_1} - Ff_{r_2}), \quad (3)$$

其中式(3)主要包括如下两步操作:

第1步 以概率 F 获取果蝇 Ff_{r_1} , Ff_{r_2} 之间的差分信息:

$$\Delta_i = F \otimes (Ff_{r_1} - Ff_{r_2}) \Leftrightarrow$$

$$\Delta_{i,j} = \begin{cases} Ff_{r_1,j} - Ff_{r_2,j}, & \text{rand}(\cdot) < F, \\ 0, & \text{其他,} \end{cases} \quad (4)$$

其中rand(·)是(0, 1)上均匀分布的随机数.

第2步 利用获取的差分信息更新自身位置

$$Ff'_i = Ff_i \oplus \Delta_i \Leftrightarrow Ff'_{i,IP_f(j)} = Ff_{i,j}, \quad (5)$$

其中 $IP_f(j)$ 表示果蝇 Ff_i 利用所获差分信息 Δ_i 移动工件位置后, 位于位置 j 的工件移动后的位置.

移动工件位置的过程如下: 对于果蝇 $Ff_i = (\pi_1, \pi_2, \dots, \pi_n)$, 其对应的位置下标向量为 $IP = (1, 2, \dots, n)$, 利用所获差分信息 Δ_i 更新位置下标向量, 即 $IP_t = IP + \Delta_i$. 对 IP_t 中位置下标按升序排序, 排序后得到最终的位置下标向量 $IP_f = (I_{p1}, I_{p2}, \dots, I_{pn})$, 如果 $I_{pk} = j$, 则 $\text{rank}^{-1}(j) = k$. 若存在多个相同位置下标, 则位置靠前的下标排序靠后. 如此, 既保证了个体的可行性, 又可增加生成个体的多样性.

下面以一个5个工件的问题为例对协作过程给予说明. 假设 $Ff_i = (3, 1, 5, 4, 2)$, $Ff_{r_1} = (2, 4, 3, 1, 5)$, $Ff_{r_2} = (3, 1, 2, 5, 4)$, 则 Ff_{r_1}, Ff_{r_2} 之间的差分信息为 $(-1, 3, 1, -4, 1)$. 若随机产生的服从(0, 1)区间均匀分布的5个随机数为 $(0.52, 0.15, 0.22, 0.18, 0.76)$ 且 $F = 0.5$, 则以0.5的概率获取到的差分信息为 $\Delta_i = (0, 3, 1, -4, 0)$. Ff_i 的原始位置下标向量为 $IP = (1, 2, 3, 4, 5)$. 根据差分信息 Δ_i 移动工件位置, 移动后的位置下标向量为 $IP_t = IP + \Delta_i = (1, 5, 4, 0, 5)$. 对移动后的位置下标按升序排序, 排序后的位置下标向量为 $IP_f = (2, 5, 3, 1, 4)$, 则向量 $\text{rank}^{-1} = (4, 1, 3, 5, 2)$, 于是得到新果蝇 $Ff'_i = Ff_{i,\text{rank}^{-1}(j)} = (4, 3, 5, 2, 1)$. 上述过程如图2所示.

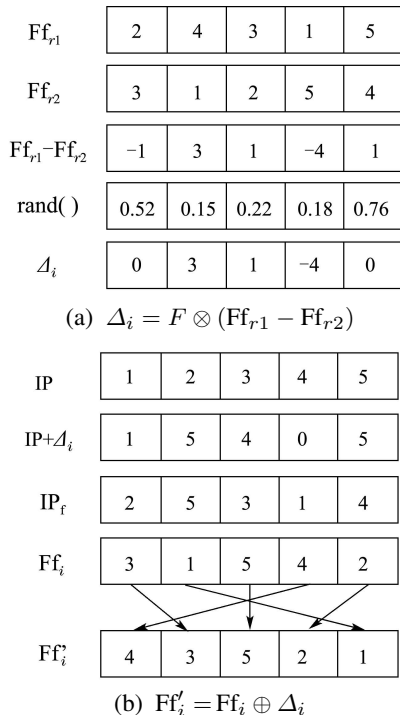


图 2 引导个体生成机制

Fig. 2 Generating scheme of guiding individual

4.5 退火机制(Annealing scheme)

针对PFSP存在多个分布不规则的局部极小解的特点, 将模拟退火的概率突跳机制嵌入HDFOA, 以突跳概率接受果蝇协作进化过程产生的最优引导个体, 从而一定程度上避免搜索过程陷入局部极小. 退火机制设计如下:

初温. 当初种群产生后, 确定其中的最优和最差个体, 相应目标值分别为 f_b 和 f_w , 确定初始的接受概率为 P_0 , 则初温 $T_0 = -(f_w - f_b) / \ln P_0$.

突跳概率. 若 $\min\{1, \exp(-D/T_g)\} > \text{rand}[0, 1]$, 则算法接受新个体. 其中 D 为新旧个体的目标值之差, T_g 为第 g 代进化时的温度.

退温. 采用指数退温的方式, 即 $T_{g+1} = \lambda \times T_g$, 本文设置 $\lambda = 0.95$.

4.6 HDFOA流程(Flowchart of HDFOA)

求解PFSP的HDFOA的流程如图3所示. 由图3可见, HDFOA包括4个主要环节: 插入操作产生邻域解的嗅觉搜索过程、贪婪策略更新个体的视觉搜索过程、基于差分信息产生引导个体的协作进化机制、概率接受最优引导个体的退火机制.

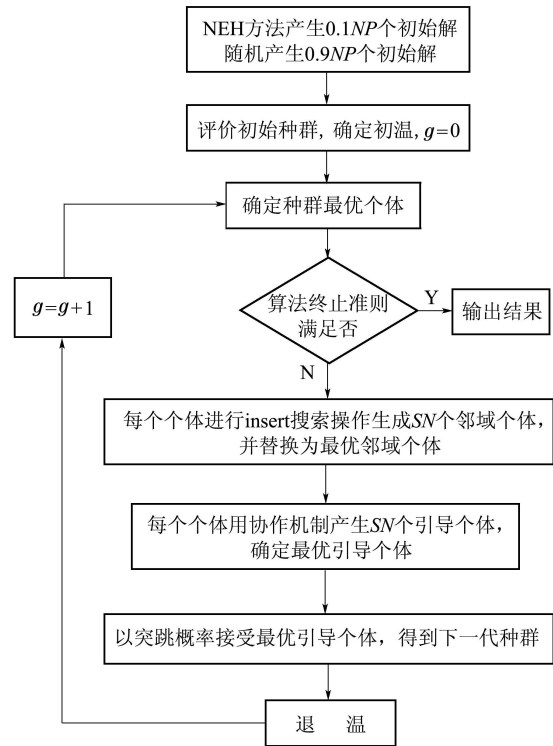


图 3 HDFOA的流程

Fig. 3 Flowchart of HDFOA

4.7 计算复杂度分析(Analysis of computational complexity)

HDFOA算法种群中每个个体采用插入操作产生一个邻域解的复杂度为 $O(n^2)$. 由于嗅觉搜索阶段种群中的 NP 个个体产生 SN 个邻域解, 因此总计算复杂度为 $O(NP \times SN \times n^2)$.

视觉搜索过程每个个体从 SN 个邻域解中选择最优解更新自身位置, 总计算复杂度为 $O(NP \times SN)$.

对于种群中 NP 个个体, 采用快速排序算法对位置下标排序, 并利用差分信息产生 SN 个引导个体, 因此总计算复杂度为 $O(SN \times NP \times n \log n)$.

综合算法各主要搜索操作的计算复杂度可见, HDFOA的整体计算复杂度并不大.

5 仿真测试和比较(Simulation test and comparisons)

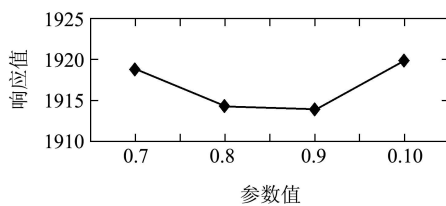
选取国际标准测试集Car类和Rec类共29实例进行算法性能测试和比较, 仿真计算机为Intel Core2(TM) Q9550(2.83 GHz), 编程语言为C++. 由于算法性能可能依赖于参数设置, 因此先对参数设置进行试验探讨.

5.1 参数设置探讨 (Investigation of parameters setting)

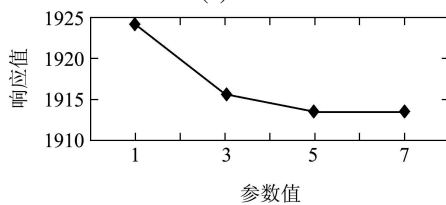
与文献[10]相同, HDFOA设置种群规模 $NP = 2n$, 终止准则设置为300代总进化代数. 此外, HDFOA还有3个关键参数: 嗅觉搜索个体数和引导个体生成数 SN 、协作参与度 F 和退火初始接受概率 P_0 . 在此利用Rec17算法采用试验设计方法(design of experiment, DOE)^[17]探讨参数对算法性能的影响. 各参数均选取4个水平值, 如表1所示, 从而建立规模为 $L_{16}(4^3)$ 的正交试验表. 算法在每种参数组合下均独立运行20次, 所得平均性能(average, AVG)作为评价指标. 正交表和所得评价指标如表2所示. 根据正交表, 计算各参数的极差和重要程度, 如表3所示, 进而绘制各参数对算法性能的影响趋势, 如图4所示.

由表3可见, 参数 SN 的极差最大, 这表明嗅觉搜索个体数和引导个体生成数 SN 对算法影响性能最大, 原因在于参数 SN 决定了算法对局部探索的深度和全局搜索的广度. 影响程度其次的是退火初始接受概率 P_0 , 由于问题的多极小特性, P_0 太小不足以使算法跳出局部极小. 协作参与度 F 的影响最弱, 但太小或太大都不利于全局搜索.

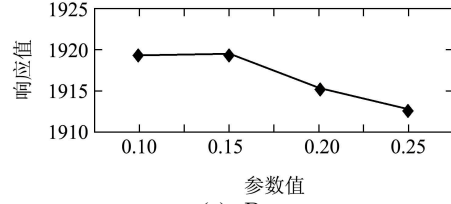
基于以上分析, 兼顾算法搜索质量和复杂度, 建议参数设置为: $F = 0.9$, $SN = 5$, $P_0 = 0.25$.



(a) F



(b) SN



(c) P_0

图4 各参数对算法性能影响的水平趋势

Fig. 4 Factor level trend of the parameters

表1 参数水平

Table 1 Parameter values

参数	水平			
	1	2	3	4
F	0.7	0.8	0.9	0.95
SN	1	3	5	7
P_0	0.1	0.15	0.2	0.25

表2 正交表和AVG统计

Table 2 Orthogonal array and AVG values

参数组合编号	水平			AVG
	FN	SN	P_0	
1	1	1	1	1930.55
2	1	2	2	1920.20
3	1	3	3	1914.25
4	1	4	4	1910.60
5	2	1	2	1924.85
6	2	2	1	1914.25
7	2	3	4	1909.80
8	2	4	3	1908.35
9	3	1	3	1923.35
10	3	2	4	1912.75
11	3	3	1	1908.35
12	3	4	2	1911.15
13	4	1	4	1918.00
14	4	2	3	1915.55
15	4	3	2	1921.85
16	4	4	1	1923.95

表3 各参数响应值

Table 3 Response value

水平	参数		
	F	SN	P_0
1	1918.90	1924.19	1919.27
2	1914.31	1915.69	1919.51
3	1913.90	1913.56	1915.38
4	1919.84	1913.51	1912.79
极差	5.94	10.68	6.73
等级	3	1	2

5.2 实验统计结果与比较 (Results and comparisons)

为了验证算法的性能, 将HDFOA与求解PFSP的一些现有的混合遗传算法(hybrid genetic algorithm, HGA)^[4]、模拟退火算法 OSA (Osman's simulated annealing)^[5]、混合差分进化(hybrid differential evolution, HDE)^[10]进行统计比较. 同时为了验证嵌入退火机制的有效性, 将HDFOA与没有嵌入退火的DFOA进行比较. 各实例算法独立运行20次, 实验结

果如表4所示, 其中BRE, ARE分别为各算法所得最优结果、平均结果相对已知最优值 C^* 的百分相对偏差, SD为20次结果的标准差.

由表4可见, HDFOA在Car类问题上能100%找到最优解, 在绝大部分Rec类问题上都能得到优于其他算法的解.

同时, HDFOA对所有问题的平均性能远远好于其他算法, 而且整体方差很小, 仅为HDE的25%. 这表明HDFOA的优化质量和鲁棒性好于文献[4-5, 10]算法.

表 4 结果比较

Table 4 Comparison of results

问题	$n \times m$	c^*	HDFOA				DFOA			HDE ^[10]			OSA ^[5]			HGA ^[4]		
			BRE	ARE	SD	CPU/s	BRE	ARE	SD	BRE	ARE	SD	BRE	ARE	SD	BRE	ARE	SD
Car1	11×5	7038	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0
Car2	13×4	7166	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0
Car3	12×5	7312	0	0	0	0.1	0	0.098	18.35	0	0.536	44.35	0	0.625	47.19	0	1.504	78.14
Car4	14×4	8003	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0
Car5	10×6	7720	0	0	0	0.1	0	0.042	5.779	0	0.593	49.54	0	0.801	50.73	0	0.938	66.34
Car6	8×9	8505	0	0	0	0.1	0	0.051	16.21	0	0.153	27.41	0	2.093	274.7	0	2.132	84.13
Car7	7×7	6590	0	0	0	0.1	0	0	0	0	0.448	60.22	0	1.483	114.2	0	1.003	79.72
Car8	8×8	8366	0	0	0	0.1	0	0	0	0	0	0	0	2.297	254.6	0	1.281	63.86
Rec01	20×5	1247	0	0.096	0.980	0.5	0.160	0.160	0	0	0.152	0.447	0.160	0.160	0	0.160	0.192	1.265
Rec03	20×5	1109	0	0	0	0.1	0	0.078	0.957	0	0.153	1.252	0	0.189	1.853	0	0.271	1.160
Rec05	20×5	1242	0	0.221	0.766	0.5	0.242	0.242	0	0.242	0.383	3.327	0.242	0.588	4.620	0.242	0.628	4.131
Rec07	20×10	1566	0	0	0	0.1	0	0.749	7.759	0	0.920	7.589	0	0.434	11.59	0.115	1.149	5.692
Rec09	20×10	1537	0	0	0	0.1	0	0.399	7.247	0	0.273	11.59	0	0.690	12.39	0.605	1.627	9.903
Rec11	20×10	1431	0	0	0	0.1	0	0.154	3.081	0	0	0	0	2.215	37.60	0	1.532	15.14
Rec13	20×15	1930	0	0.194	1.699	3.4	0.311	0.960	6.365	0.259	0.705	8.566	0.311	1.793	14.69	0.415	1.974	12.25
Rec15	20×15	1950	0	0.118	1.900	3.4	0.667	0.937	4.203	0.051	0.995	14.96	0.718	1.569	16.07	0.615	2.385	14.01
Rec17	20×15	1902	0	0.037	2.100	3.4	0.736	1.370	7.585	0.368	1.309	11.87	1.840	3.796	36.72	1.840	2.482	9.693
Rec19	30×10	2093	0.287	0.506	4.063	20.3	0.812	1.354	6.063	0.287	0.908	9.104	0.287	0.803	9.484	1.113	2.676	14.59
Rec21	30×10	2017	0.446	1.385	5.157	20.3	1.438	1.567	1.625	0.198	1.284	9.171	1.438	1.477	1.687	1.522	1.636	6.001
Rec23	30×10	2011	0.348	0.604	2.816	20.3	0.497	0.620	2.825	0.497	0.696	10.63	0.497	0.845	10.82	0.497	2.188	12.72
Rec25	30×15	2513	0.398	1.146	7.935	31.6	0.796	1.281	5.822	0.676	1.429	16.49	1.194	1.938	15.06	1.922	2.705	10.96
Rec27	30×15	2373	0.337	0.995	5.352	31.6	0.759	1.149	4.669	0.843	1.197	4.600	0.843	1.854	21.06	1.551	2.318	9.976
Rec29	30×15	2287	0.481	0.947	4.798	31.7	0.918	1.729	8.024	0.525	1.299	13.34	0.612	2.882	38.83	2.610	3.629	22.43
Rec31	50×10	3045	0.854	1.349	7.971	101.9	1.839	2.496	6.995	0.427	1.192	13.11	0.296	1.333	30.39	1.156	2.759	14.03
Rec33	50×10	3114	0.032	0.396	4.756	102.3	0.289	0.692	4.884	0.353	0.787	4.743	0.128	0.732	7.315	0.450	1.188	5.376
Rec35	50×10	3277	0	0	0	5.2	0	0	0	0	0	0	0	0	0	0	0.131	3.335
Rec37	75×20	4951	2.182	2.847	12.200	178.6	2.949	3.513	13.22	1.697	2.632	33.41	2.000	2.751	25.43	4.312	5.096	23.90
Rec39	75×20	5087	1.199	1.853	12.780	180.3	2.516	3.144	12.43	1.278	1.543	7.863	0.767	1.240	12.31	2.597	3.025	10.72
Rec41	75×20	4960	2.460	2.935	13.300	179.5	3.851	4.216	11.49	1.714	2.615	36.39	1.734	2.726	39.38	4.133	5.599	22.55
平均			0.311	0.539	3.054	31.6	0.648	0.931	5.384	0.325	0.766	13.79	0.451	1.287	37.54	0.892	1.801	20.73

另外, 相比于DFOA, HDFOA在所有问题上都取得了更好的结果, 而且稳定性更好. 这表明引入概率突跳机制对混合算法的设计是有效的, 有利于算法取得更好性能的解.

就几种算法的终止条件而言, HDFOA和HDE^[10]的总进化代数都设置为300, HGA^[4]的总进

化代数设置为机器数与工件数的乘积 $m \times n$, OSA^[5]的总迭代步数设置为 $300 \times n \times (n - 1)$. 因此, 考虑种群规模和局部搜索等因素, HDFOA对大规模问题的总计算量并不比文献[4-5]算法大. 由表4可见, HDFOA对所有测试问题的平均运行时间约为31.6s, 即使对于75个工件20台机器的问题也

仅需约180 s CPU时间. 因此, HDFOA对求解置换流水线调度问题还是很高效的.

图5给出HDFOA求解Rec17问题的收敛曲线. 由图可见, HDFOA具有较快的收敛性, 而且算法在搜索过程中具有跳出局部极小的能力, 最终能获得性能优良的解.

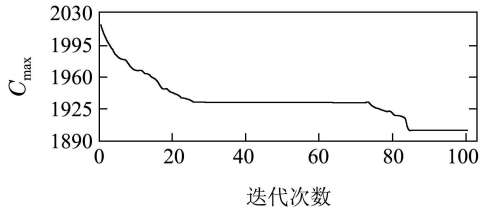


图5 HDFOA的收敛曲线

Fig. 5 The convergence curve of HDFOA

6 结论(Conclusions)

针对PFSP问题首次提出了一种混合离散果蝇算法. 通过设计嗅觉搜索过程、视觉搜索过程、协作进化过程和退火过程, 很好地均衡了全局和局部搜索能力. 分析了算法的计算复杂度, 探讨了参数对性能的影响, 算法比较验证了混合算法的有效性和鲁棒性. 进一步的工作是研究多目标生产调度问题的高效果蝇算法以及针对其他类型的调度指标或调度问题^[18]的有效果蝇算法.

参考文献(References):

- [1] 王凌. 车间调度及其遗传算法 [M]. 北京: 清华大学出版社, 2003. (WANG Ling. *Shop Scheduling with Genetic Algorithms* [M]. Beijing: Tsinghua University Press, 2003.)
- [2] GAREY M R, JOHNSON D S, SETHY R. The complexity of flow-shop and job-shop scheduling [J]. *Mathematics of Operations Research*, 1976, 1(2): 117 – 129.
- [3] PINEDO M L. *Scheduling: Theory, Algorithms, and Systems* [M]. Berlin: Springer, 2012.
- [4] WANG L, ZHENG D Z. An effective hybrid heuristic for flow-shop scheduling [J]. *International Journal Advanced Manufacturing Technology*, 2003, 21(1): 38 – 44.
- [5] OSMAN I H, POTTS C N. Simulated annealing for permutation flow-shop scheduling [J]. *Omega*, 1989, 17(6): 551 – 557.
- [6] NOWICKI E, SMUTNICKI C. A fast tabu search algorithm for the permutation flow-shop scheduling [J]. *European Journal of Operational Research*, 1996, 91(1): 160 – 175.
- [7] GRABOWSKI J, WODECKI M. A very fast tabu search algorithm for the permutation flow-shop problem with makespan criterion [J]. *Computers & Operations Research*, 2004, 31(11): 1891 – 1909.
- [8] 金锋, 宋士吉, 吴澄. 一类基于FSP问题Block性质的快速TS算法 [J]. *控制与决策*, 2007, 22(3): 247 – 251. (JIN Feng, SONG Shiji, WU Chen. Fast TS algorithm based on block properties of FSP [J]. *Control and Decision*, 2007, 22(3): 247 – 251.)
- [9] 张长胜, 孙吉贵, 杨轻云, 等. 一种求解车间调度的混合算法 [J]. *自动化学报*, 2009, 35(3): 332 – 336. (ZHANG Changsheng, SUN Jigui, YANG Qingyun, et al. A hybrid algorithm for flow-shop scheduling problem [J]. *Acta Automatica Sinica*, 2009, 35(3): 332 – 336.)
- [10] QIAN B, WANG L, HU R, et al. A hybrid differential evolution for permutation flow-shop scheduling [J]. *International Journal Advanced Manufacturing Technology*, 2008, 38(7/8): 757 – 777.
- [11] 潘玉霞, 潘全科, 李俊青. 蛙跳优化算法求解多目标无等待流水线调度 [J]. *控制理论与应用*, 2011, 28(10): 1363 – 1368. (PAN Yuxia, PAN Quanke, LI Junqing. Shuffled frog-leaping algorithm for multi-objective no-wait flowshop scheduling [J]. *Control Theory & Applications*, 2011, 28(10): 1363 – 1368.)
- [12] PAN W T. A new fruit fly optimization algorithm: taking the financial distress model as an example [J]. *Knowledge-Based Systems*, 2012, 26(2): 69 – 74.
- [13] LI H, GUO S, LI C, et al. A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm [J]. *Knowledge-Based Systems*, 2013, 37(1): 378 – 387.
- [14] LIN S M. Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network [J]. *Neural Computing & Applications*, 2013, 22(3/4): 783 – 791.
- [15] HAN J, WANG P, YANG X. Tuning of PID controller based on fruit fly optimization algorithm [C] // *Proceedings of IEEE International Conference on Mechatronics and Automation*. Washington: IEEE, 2012: 409 – 413.
- [16] NAWAZ M, ENSCORE J, HAM I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem [J]. *Omega*, 1983, 11(1): 91 – 95.
- [17] MONTGOMERY D C. *Design and Analysis of Experiments* [M]. Hoboken: John Wiley and Sons, 2005.
- [18] 王凌, 周刚, 许焯, 等. 求解不相关并行机混合流水线调度问题的人工蜂群算法 [J]. *控制理论与应用*, 2012, 29(12): 1151 – 1157. (WANG Ling, ZHOU Gang, XU Ye, et al. An artificial bee colony algorithm for solving hybrid flow-shop scheduling problem with unrelated parallel machines [J]. *Control Theory & Applications*, 2012, 29(12): 1151 – 1157.)

作者简介:

郑晓龙 (1988–), 男, 博士研究生, 主要研究方向为智能优化理论与算法, E-mail: zhengxl11@mails.tsinghua.edu.cn;

王凌 (1972–), 男, 教授, 博士生导师, 主要研究方向为优化与调度理论与方法, Email: wangling@tsinghua.edu.cn;

王圣尧 (1988–), 男, 博士研究生, 主要研究方向为智能调度理论与方法, E-mail: wangshengyao@tsinghua.org.cn.