

求解不相关并行机混合流水线调度问题的人工蜂群算法

王凌, 周刚, 许焯, 王圣尧

(清华大学自动化系, 北京 100084)

摘要: 针对不相关并行机混合流水线调度问题的特点, 设计了一种基于排列的编码和解码方法, 提出了一种有效的人工蜂群算法. 在引领蜂和跟随蜂搜索阶段采用3种有效的邻域搜索方法, 以丰富搜索行为; 在侦察蜂搜索阶段通过随机搜索对种群进行更新, 以增强种群多样性. 同时, 通过试验设计方法对算法的参数设置进行了分析, 给出指导性参数组合. 通过基于典型实例的数值仿真以及与已有代表性算法的比较, 验证了所提算法的有效性和鲁棒性.

关键词: 混合流水线调度; 不相关并行机; 人工蜂群算法; 实验设计

中图分类号: TP18 **文献标识码:** A

An artificial bee colony algorithm for solving hybrid flow-shop scheduling problem with unrelated parallel machines

WANG Ling, ZHOU Gang, XU Ye, WANG Sheng-yao

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

Abstract: According to the characteristics of the hybrid flow-shop scheduling problem with unrelated parallel machines (HFSP-UPM), we design a permutation-based method for encoding and decoding, and propose an effective artificial bee colony (ABC) algorithm. At the employed bee phase and the onlooker bee phase, three effective neighbor-search approaches are used to enrich the searching behavior; at the scout bee phase, the population is updated by using random search to enhance the diversity of population. Based on Taguchi method for experiment design (DOE), the effect from the parameter-setting is investigated and suitable parameter values are suggested. Numerical simulation based on benchmark examples and comparisons with the existing typical algorithms demonstrate the effectiveness and robustness of the proposed algorithm.

Key words: hybrid flow-shop scheduling; unrelated parallel machine; artificial bee colony; design of experiment

1 引言(Introduction)

混合流水线调度问题(hybrid flow-shop scheduling problem, HFSP)^[1-2]具有很强的应用背景, 广泛存在于化工、冶金、纺织、机械、半导体、物流等领域, 大量制造与服务问题均可归结为HFSP. HFSP整体上具有流水作业特征, 同时某些工序上存在并行机加工的特点, 求解难度大, 属于非确定性多项式(non-deterministic polynomial hard, NP-hard)问题. 因此, HFSP的研究具有重要的学术意义和应用价值. HFSP按照并行机类型分为3类: 相同并行机混合流水线^[3], 即每阶段上工件在任一台并行机上的加工时间相同; 均匀并行机混合流水线^[4], 即每阶段上工件在任一台并行机上的加工时间与该台机器的加工速度成反比; 不相关并行机混合流水线^[5], 即每阶段上工件在任两台并行机上的加工时间互不相关, 取决于工件与机器的匹配程度.

HFSP的求解方法早期主要包括精确算法^[6]和

启发式方法^[7]. 精确算法在理论上能得到最优解, 但计算时间通常难以接受, 一般只适于解决小规模问题. 启发式方法可在较短时间内快速构造解, 但难以保证解的质量. 近年来, 许多智能方法被应用于HFSP, 譬如遗传算法(genetic algorithm, GA)^[8]、模拟退火^[9]、禁忌搜索^[10]、蚁群算法^[11]、微粒群优化^[12]、人工免疫系统(artificial immune system, AIS)^[13]等. 人工蜂群算法(artificial bee colony, ABC)^[14]是一种模仿蜜蜂觅食的群智能算法, 已被广泛用于连续和组合优化问题, 如约束优化^[15]、聚类问题^[16]等.

在调度方面, Pan等^[17]针对批量流水线调度提出了一种基于自适应搜索策略的离散人工蜂群算法, 使用排列编码并采用插入和交换等操作实现离散空间的搜索; Li等^[18]针对多目标柔性作业车间调度, 提出了一种有效的ABC算法. 本文针对不相关并行机混合流水线调度问题(HFSP with unrelated parallel

machines, HFSP-UPM)的特点,设计一种基于排列的编码和解码方法,提出了3种有效的邻域搜索方法进行引领蜂和跟随蜂阶段的操作,并运用侦察蜂的随机搜索增强种群多样性.通过试验设计方法分析参数对性能的影响并给出有效的参数组合.最后通过比较验证所提算法的有效性和鲁棒性.

2 不相关并行机混合流水线调度(HFSP-UPM)

2.1 问题描述(Problem statement)

HFSP-UPM可描述为^[5]: n 个工件在流水线上进行 S 个阶段的加工,各阶段至少有一台机器且至少有一个阶段存在并行机,同一阶段上各机器加工同一工件的加工时间有所不同,在每一阶段各工件均要完成一道工序,但各工件的每道工序可在相应阶段上的任意机器上加工,已知工件各道工序在各机器上的处理时间,要求确定所有工件的排序以及每一阶段上机器的分配情况,使得某种调度指标最小,如最大完工时间.图1给出了问题的一个图例.

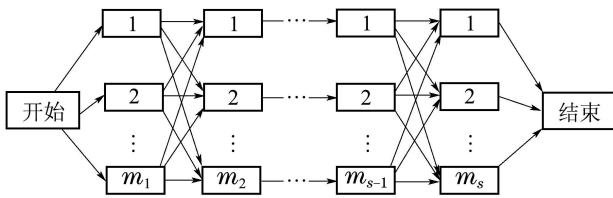


图1 HFSP-UPM图例
Fig. 1 An example of HFSP-UPM

2.2 数学模型(Mathematical formulation)

假设工件一旦开始加工便不可中断;一台机器同一时刻只能加工一个工件;一个工件同一时刻只能在一台机器上加工;工件可在每阶段的任意一台机器上加工^[19].记 J_i 为工件序号, $i = 1, 2, \dots, n$; m_j 为每一阶段的机器数, $j = 1, 2, \dots, S$; $t_{i,j,k}$ 为工件 J_i 第 j 道工序在第 k 台机器上的加工时间; $s_{i,j,k}$ 和 $e_{i,j,k}$ 分别为工件 J_i 第 j 道工序在第 k 台机器上开始加工时间和加工完成时间; C_i 为工件 J_i 的加工完毕时间; $C_{\max} = \max\{C_1, C_2, \dots, C_n\}$ 为最大完成时间; L 为足够大数. HFSP-UPM的数学模型如下:

$$\min C_{\max} \tag{1}$$

$$\text{s.t. } \sum_{l=1}^n x_{i,l} = 1, l = 1, 2, \dots, n, \tag{2}$$

$$\sum_{l=1}^n x_{i,l} = 1, i = 1, 2, \dots, n, \tag{3}$$

$$\begin{cases} \sum_{k=1}^{m_j} y_{i,j,k} = 1, \\ i = 1, 2, \dots, n, j = 1, 2, \dots, S, \end{cases} \tag{4}$$

$$\begin{cases} e_{i,j,k} = s_{i,j,k} + t_{i,j,k}, i = 1, 2, \dots, n, \\ j = 1, 2, \dots, S, k = 1, 2, \dots, m_j, \end{cases} \tag{5}$$

$$\begin{cases} e_{i,j,k} \leq s_{i,j+1,k'}, \\ i = 1, 2, \dots, n, j = 1, 2, \dots, S-1, \\ k = 1, 2, \dots, m_j, k' = 1, 2, \dots, m_{j+1}, \end{cases} \tag{6}$$

$$\begin{cases} \sum_{i=1}^n x_{i,l} s_{i,1,k} \leq \sum_{i=1}^n x_{i,l+1} s_{i,1,k}, \\ i = 1, 2, \dots, n, l = 1, 2, \dots, n-1, \\ k = 1, 2, \dots, m_1, \end{cases} \tag{7}$$

$$\begin{cases} \sum_{i=1}^n x_{i,l_1} y_{i,j,k} e_{i,j,k} \leq \\ \sum_{i=1}^n x_{i,l_2} y_{i,j,k} s_{i,j,k'} + (1 - \sum_{i=1}^n x_{i,l_2} y_{i,j,k}) L, \\ l_1, l_2 = 1, 2, \dots, n, l_1 \leq l_2, \\ j = 1, 2, \dots, S, k, k' = 1, 2, \dots, m_j, \end{cases} \tag{8}$$

$$x_{i,l} = \begin{cases} 1, & \text{若工件 } i \text{ 被安排在第 } l \text{ 个位置,} \\ 0, & \text{否则,} \end{cases} \tag{9}$$

$$y_{i,j,k} = \begin{cases} 1, & \text{若工件 } i \text{ 的工序 } j \text{ 在机器 } k \text{ 上,} \\ 0, & \text{否则,} \end{cases} \tag{10}$$

其中:式(1)为调度性能指标;式(2)确保每个优先级位置只能对应一个工件;式(3)确保每个工件只有一个优先级位置;式(4)表示任一阶段每个工件只能在一台机器上加工;式(5)表示同一阶段上工序完成时间和开始时间的关系;式(6)则表示同一工件不同工序间的先后制约关系;式(7)表示第1阶段上调度排列中排位越前的工件开始处理时间越早;式(8)表示同一阶段分配在同一机器上的工件排位靠后的工件必须等靠前的工件加工完毕后方可进行加工,当处于不同位置的工件不在同一阶段的同一机器上加工时,足够大的 L 保证了不等式恒成立.

3 人工蜂群算法(The artificial bee colony algorithm)

人工蜂群算法^[14]是基于自然界中蜂群的自组织模型和群体智能的一种仿生算法.蜂群由引领蜂、跟随蜂和侦查蜂组成,食物源的位置代表问题的解,食物源的个数等于引领蜂和跟随蜂的个数.引领蜂首先寻找食物源,跟随蜂按比例概率 $P_i = f_i / \sum_{j=1}^{P_{\text{size}}} f_j$ 选择引领蜂带回的食物源,其中: f_i 是第 i 个食物源的适应度(本文设置为 $1/C_{\max}$), P_{size} 为种群规模,适应度越大的食物源被选择的概率越大.随后,跟随蜂对选中食物源进行一次邻域搜索,并保留好的解.若某食物源连续 Limit 代未得到改善,则对应于该食物源的引领蜂就变为侦查蜂,随机寻找新

的食物源.

ABC算法流程如图2所示^[18].

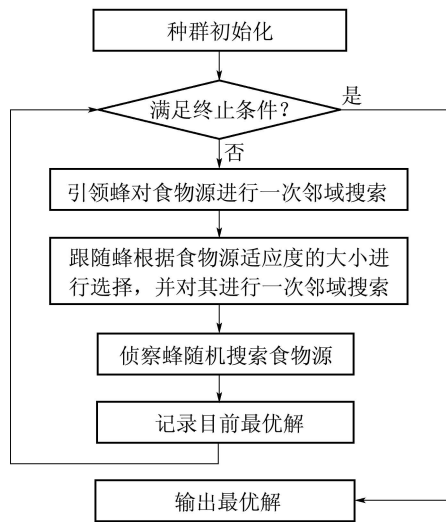


图 2 ABC算法流程

Fig. 2 Framework of the ABC algorithm

4 HFSP-UPM的ABC算法(ABC for HFSP-UPM)

针对HFSP-UPM的特点, 本节设计ABC算法中解的编码和解码、初始化以及引领蜂阶段、跟随蜂阶段和侦察蜂阶段的搜索操作, 最后给出算法流程.

4.1 编码和解码(Encoding and decoding scheme)

种群中每一个体对应问题的一个解, 个体采用排列编码, 即所有工件序号的排列组合, 工件号在个体中的位置表示工件在第1阶段的加工顺序. 譬如, 对于6个工件的HFSP-UPM, 个体{2, 4, 5, 1, 6, 3}表示在加工的第1阶段, 工件2最先加工, 依次是工件4和工件5, 工件3最后加工.

对于一个排列, 按机器分配规则依次将所有工件分配到各阶段机器上, 形成一个可行调度并计算目标值的过程称为解码. HFSP-UPM的解码可分为两部分: 工件的排序和机器的分配. 在工件排序部分, 针对多阶段加工的特点, 采用如下策略: 第1阶段($j = 1$)按照编码确定工件安排的先后顺序, 后续阶段基于先入先出方式确定操作序列, 即对于 $j(j > 1)$ 阶段所有工件按照前一阶段的完成时间排定顺序, 前阶段先完成的工件先加工, 若多个工件在前一阶段的完成时间相同, 则随机确定这些工件的加工顺序. 在机器分配方面, 在原有最先空闲机器规则^[20]的基础上, 提出如下机器分配规则:

步骤 1 $j = 1$;

步骤 2 根据各工件在阶段 j 上的加工顺序依次判断每个工件 J_i 在第 k 台机器上的最早允许加工时间, 即机器 k 的释放时间 r_k 和 J_i 在阶段 $j - 1$ 上的完成

时间 $C_{i,j-1}$ 之间的最大值. 对于 $j = 1$, 有 $C_{i,j-1} = 0$;

步骤 3 对于每个 J_i 根据 $\max(r_k, C_{i,j-1}) + t_{i,j,k}$ 选择值最小的机器作为 J_i 的加工机器, 更新工件 J_i 在第 j 阶段上的加工完成时间及机器 k 的释放时间;

步骤 4 $j = j + 1$;

步骤 5 重复步骤2-4直到所有阶段加工完成.

示例 考虑如下3阶段HFSP-UPM. 假设每阶段都有两台机器, 共6个工件, 各工件在各阶段机器上的加工时间如表1所示. 个体{2, 4, 5, 1, 6, 3}经上述解码后所得调度的甘特图如图3所示, 最大完成时间为14, 图中“2-3”代表第2个工件的第3道工序.

表 1 加工时间

Table 1 Processing time of problem 1

工件	阶段1		阶段2		阶段3	
	机器1	机器2	机器3	机器4	机器5	机器6
1	2	2	4	3	1	1
2	2	3	2	3	2	1
3	4	3	2	3	2	2
4	2	3	1	1	2	1
5	4	2	2	1	4	5
6	1	2	3	2	3	6

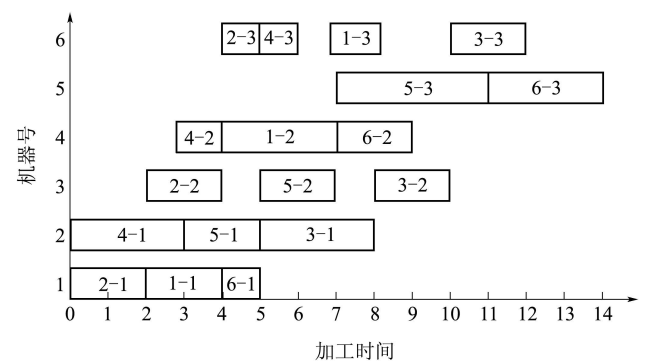


图 3 示例对应调度的甘特图

Fig. 3 The Gantt chart of the solution

4.2 种群初始化(Population initialization)

ABC算法通常采用随机方法产生初始种群中的解 $s_i, i = 1, 2, \dots, P_{size}$. 本文同样采用随机初始化产生种群, 以保证初始种群的分散性.

4.3 邻域解的产生(Exploitation search operators)

对于引领蜂和跟随蜂阶段的操作, 本文提出3种邻域搜索方法以丰富搜索行为, Scheme#1, Scheme#2和Scheme#3. 3种方法在算法中的使用概率分别设为 p_1, p_2 和 $1 - p_1 - p_2$.

Scheme#1 随机选择个体中的两个工件, 通过交换位置产生新解.

Scheme#2 随机选择阶段 $j(j > 1)$, 选择两个在该阶段不同机器上的工件, 交换其位置. j 阶段之前工件排序和机器分配保持不变, j 阶段之后各阶段上工件排序和机器分配情况则按改动后阶段 j 上各工件完成时间的先后顺序作调整. 譬如, 交换示例中第2阶段上的工件1和工件5的位置, 则所得调度如图4所示, 最大完成时间为13.

Scheme#3 随机选择阶段 $j(j > 1)$, 选择两个在该阶段同一机器上相邻的两个工件, 交换其位置. j 阶段之前的工件排序和机器分配保持不变, j 阶段之后的各工件排序和机器分配情况则按改动后阶段 j 上工件完成时间先后顺序作调整.

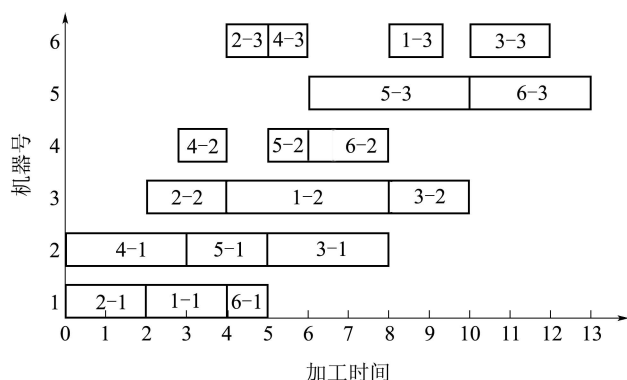


图4 Scheme#2产生的解

Fig. 4 Solution after using Scheme#2

4.4 引领蜂操作(Employed bee phase)

在基本ABC算法中, 每个引领蜂首先对相应的食物源 s_i 进行一次邻域搜索, 如果搜索到的新食物源 s_{new} 优于 s_i , 则用新食物源代替旧食物源; 否则保持旧食物源不变. 本文采用4.3节中的3种方法进行新食物源的搜索.

4.5 跟随蜂操作(Onlooker bee phase)

在所有引领蜂完成邻域搜索后, 跟随蜂根据食物源信息按比例概率选择食物源, 并采用4.3节中的3种方法进行食物源的邻域搜索, 同时并采用贪婪准则更新食物源.

4.6 侦查蜂操作(Scout bee phase)

在ABC算法中, 如果某个解连续Limit代搜索均不能得到改善, 则认为该解已陷入局部最优, 从而由侦查蜂随机产生新解来替换该解, 以增强种群多样性, 并避免搜索陷入局部极小.

4.7 HFSP-UPM的ABC算法流程(Procedure of ABC for HFSP-UPM)

步骤1 初始化种群, 并计算解的适应度值;

步骤2 引领蜂执行邻域搜索产生新解并计算适应度值, 基于贪婪策略进行解的替换;

步骤3 跟随蜂根据概率选择食物源并执行邻

域搜索产生新解, 基于贪婪策略替换解;

步骤4 判断是否要放弃解, 如果需要则用侦查蜂随机产生新解替换旧解;

步骤5 判断终止条件是否满足? 若是则输出优化结果; 否则转步骤2.

可见, 算法在引领蜂和跟随蜂搜索阶段通过多邻域搜索实现改良搜索, 并通过侦查蜂搜索实现探索搜索. 两种搜索相结合, 将使算法具有较好性能.

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

选取两个实例对算法进行性能测试和比较:

实例1 对应某汽车发动机厂对12个工件的加工过程^[21], 每个工件都有车、刨、磨3个工序, 现有3台车床, 2台刨床, 4台磨床, 每台机床的加工能力不同, 具体加工时间如表2所示, 表中表示为M1-M9.

实例2 对应钢铁生产中某炼钢-精炼-连铸-轧制过程^[22], 有3台炼钢炉, 3台精炼炉, 2台铸机, 2台轧机, 各机器的加工能力不同, 具体加工时间如表3所示, 表中表示为M1-M10. 算法用C++编程, 在3.21-GB RAM/2.83-GHz机器上运行. 与文献[21]相同, 对于每个实例均独立运行10次.

表2 实例1的加工时间表

Table 2 Processing time of instance 1

工件	阶段1			阶段2			阶段3		
	M1	M2	M3	M4	M5	M6	M7	M8	M9
1	2	2	3	4	5	2	3	2	3
2	4	5	4	3	4	3	4	5	4
3	6	5	4	4	2	3	4	2	5
4	4	3	4	6	5	3	6	5	8
5	4	5	3	3	1	3	4	6	5
6	6	5	4	2	3	4	3	9	5
7	5	2	4	4	6	3	4	3	5
8	3	5	4	7	5	3	3	6	4
9	2	5	4	1	2	7	8	6	5
10	3	6	4	3	4	4	8	6	7
11	5	2	4	3	5	6	7	6	5
12	6	5	4	5	4	3	4	7	5

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

ABC算法有如下关键参数: 种群规模 P_{size} , Scheme#1 执行概率 p_1 , Scheme#2 执行概率 p_2 , 参数Limit. 首先利用实例1采用试验设计方法^[23]探讨4个参数对算法性能的影响. 各参数均取4个水平, 如表4所示. 算法在每种参数组合下均独立运行50次, 设置总评价次数为3000作为终止条件, 50次运行所得平均值AVG作为评价指标. 选择规模为 $L_{16}(4^4)$ 的

正交试验, 正交表和所得AVG如表5所示.

表 3 实例2的加工时间表
Table 3 Processing time of instance 2

工件	阶段1			阶段2			阶段3		阶段4	
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
1	45	48	50	35	35	30	30	35	25	26
2	45	50	45	35	36	35	35	34	25	30
3	50	45	46	35	36	36	31	34	30	31
4	50	48	48	34	38	35	32	33	27	31
5	45	46	48	30	35	50	34	32	28	31
6	45	45	45	30	35	50	33	32	30	26
7	47	50	47	31	30	35	35	31	29	25
8	50	45	48	32	30	34	34	30	24	27
9	48	46	46	33	34	30	34	30	25	25
10	45	47	47	33	33	30	35	34	32	26
11	46	50	45	34	30	50	30	35	31	25
12	48	50	47	35	31	35	32	30	25	30

表 4 参数水平
Table 4 Parameter values

参数	水平			
	1	2	3	4
P_{size}	10	20	30	40
p_1	0	0.2	0.3	0.5
p_2	0	0.2	0.3	0.5
Limit	5	10	15	20

表 5 正交表和AVG统计
Table 5 Orthogonal array and AVG values

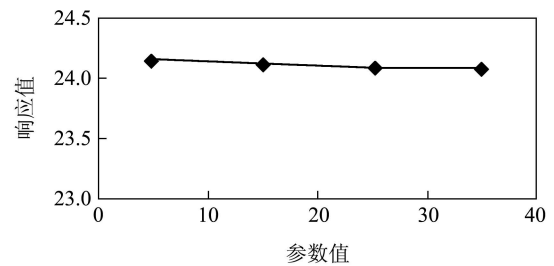
参数组合	水平				AVG
	P_{size}	p_1	p_2	Limit	
1	1	1	1	1	25.08
2	1	2	2	2	23.94
3	1	3	3	3	23.82
4	1	4	4	4	23.84
5	2	1	2	3	24.84
6	2	2	1	4	23.92
7	2	3	4	1	23.92
8	2	4	3	2	23.82
9	3	1	3	4	24.58
10	3	2	4	3	23.92
11	3	3	1	2	23.88
12	3	4	2	1	23.98
13	4	1	4	2	24.74
14	4	2	3	1	23.94
15	4	3	2	4	23.98
16	4	4	1	3	23.70

根据正交表, 各参数的极差和重要程度如表6所示. 各参数对算法性能的影响趋势如图5所示. 由表6可见, 参数 p_1 的极差最大, 这表明 p_1 对性能影响最大, 其原因在于 p_1 对应的Scheme#1操作是针对第1阶段进行的, 而第1阶段的编码对调度过程尤为重要. 参数Limit的影响程度其次, 当Limit = 15时算法性能最好, 其原因在于Limit过小不利于对局部最优解的深度搜索, 而过大会不利于种群跳出局部极小. 其余两个参数极差较小, 这说明它们对算法影响不大.

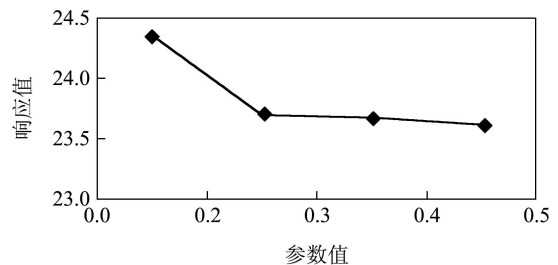
基于上述分析, 兼顾优化质量和效率, 参数设置如下: $P_{size} = 40, p_1 = 0.5, p_2 = 0.3, Limit = 15$.

表 6 各参数响应值
Table 6 Response value

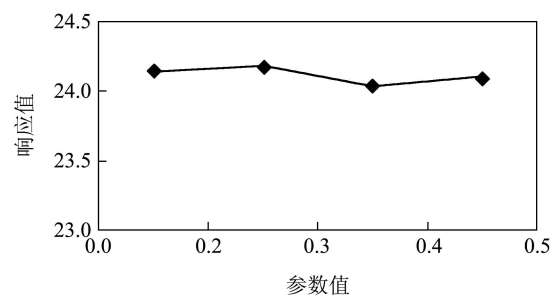
水平	P_{size}	p_1	p_2	Limit
1	24.17	24.81	21.145	24.23
2	24.125	23.93	24.185	24.095
3	24.09	23.90	24.04	24.07
4	24.09	23.835	24.105	24.08
极差	0.08	0.975	0.145	0.16
排名	4	1	3	2



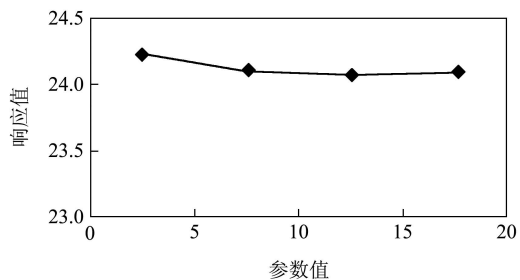
(a) P_{size}



(b) p_1



(c) p_2



(d) Limit

图 5 参数水平趋势

Fig. 5 Response value versus factor level

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

对于实例1, 将ABC与AIS^[24]、GA^[21]、差分进化(DE)^[25]进行比较, 采用相同的评价次数10000. 文献[24]仅给出AIS的一次仿真结果, 其他算法则给出10次独立运行的结果, 结果如表7所示.

表 7 求解实例1的统计结果和比较
Table 7 Comparison with AIS, GA and DE for instance 1

次数	1	2	3	4	5	6	7	8	9	10
AIS	27	—	—	—	—	—	—	—	—	—
GA	30	27	26	27	29	27	26	27	26	28
DE	24	24	24	24	24	25	24	24	24	25
ABC	23	23	23	24	23	23	23	24	23	23

由表7可见, ABC能够找到其他算法不能找到的优良解, 调度指标为23, 相应的甘特图如图6所示. 就整体性能而言, ABC能够在10次独立运行中8次找到最优解, 平均性能也优于其他算法. 因此, ABC的优化质量和鲁棒性均好于其他算法. 另外, 图7给出了群体中最优个体的目标值和所有个体的平均目标值的变化曲线. 由最优个体目标值的变化情况可见, ABC算法在50多代搜索后即可找到最优调度. 由种群平均目标值的变化情况可见, 算法在搜索过程中能够保持种群一定的多样性.

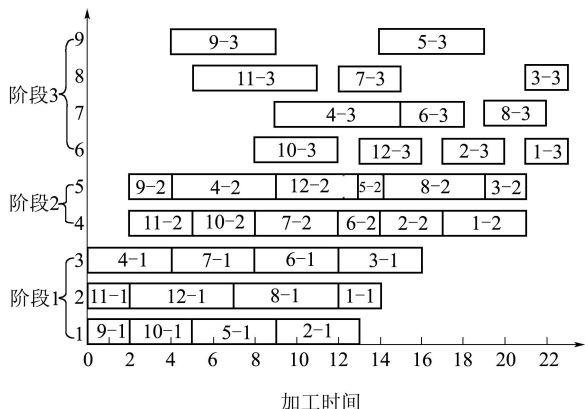


图 6 实例1最优调度的甘特图

Fig. 6 Gantt chart of the optimal solution

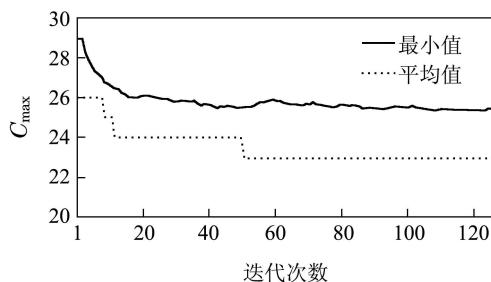


图 7 种群最小值和平均值的变化曲线

Fig. 7 Convergence curves of the best and average values

对于阶段数更大的实例2, 将ABC与GA^[22]、DE^[25]进行比较, 采用相同评价次数18000. 文[22]仅给出GA的一次仿真结果, 文[25]给出了DE的10次仿真结果. ABC与这些算法的仿真结果如表8所示.

表 8 求解实例2的统计结果和比较
Table 8 Comparison with AIS, GA and DE for instance 2

次数	1	2	3	4	5	6	7	8	9	10
GA	347	—	—	—	—	—	—	—	—	—
DE	313	319	313	302	302	315	315	319	299	299
ABC	297	297	297	297	297	297	297	297	297	297

由表8可见, ABC算法所得结果大大好于GA和DE的结果, 且ABC算法在10次实验中能100%找到最优解, 这再次验证了ABC算法的有效性和鲁棒性.

6 结论(Conclusions)

针对不相关并行机混合流水线调度问题, 本文提出了一种有效的人工蜂群算法, 通过设计基于排列的编码和解码实现调度解与搜索个体的转换, 通过采用基于多操作的邻域搜索丰富了搜索行为, 通过引入随机搜索增强了种群多样性, 通过试验设计方法考察了参数设置对算法性能的影响并给出了指导性的参数, 通过基于典型实例的算法比较验证了算法的有效性和鲁棒性. 进一步的工作将针对带多处理任务的HFSP设计有效的ABC算法.

参考文献(References):

- 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.)
- SALVADOR M S. A solution of a special class of flow shop scheduling problems [C] // *Proceedings of the Symposium on the Theory of Scheduling and Its Applications*. Berlin: Springer-Verlag, 1973: 83 – 91.
- ORTMANN M C, VIGNIER A, DARDILHAC D, et al. Branch and bound crossed with GA to solve hybrid flowshops [J]. *European Journal of Operational Research*, 1998, 107(2): 389 – 400.
- SOEWANDI H, ELMAGHRABY S E. Sequencing on two-stage hybrid flowshops with uniform machines to minimize makespan [J]. *IIE Transactions*, 2003, 35(5): 467 – 477.

- [5] FIGIELSKA E. A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flow shop with additional resources [J]. *Computers & Industrial Engineering*, 2009, 56(1): 142 – 151.
- [6] 轩华, 唐立新. 实时无等待HFS调度的一种拉格朗日松弛算法 [J]. *控制与决策*, 2006, 21(4): 376 – 380.
(XUAN Hua, TANG Lixin. Lagrangian relaxation algorithm for real-time hybrid flow-shop scheduling with no-wait in process [J]. *Control and Decision*, 2006, 21(4): 376 – 380.)
- [7] RIANE F, ARTIBA A, ELMAGHRABY S E. Sequencing a hybrid two-stage flow shop with dedicated machines [J]. *International Journal of Production Research*, 2002, 40(17): 4353 – 4380.
- [8] XIAO W, HAO P, ZHANG S, et al. Hybrid flow shop scheduling using genetic algorithms [C] // *Proceedings of the 3rd World Congress on Intelligent Control and Automation*. Shanghai: IEEE, 2000: 537 – 541.
- [9] LOW C. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines [J]. *Computers & Operations Research*, 2005, 32(8): 2013 – 2025.
- [10] WANG X, TANG L. A tabu search heuristic for the hybrid flowshop scheduling with finite intermediate buffers [J]. *Computers & Operations Research*, 2008, 36(3): 907 – 918.
- [11] ALAYKYRAN K, ENGIN O, DOYEN A. Using ant colony optimization to solve hybrid flow shop scheduling problems [J]. *International Journal of Advanced Manufacturing Technology*, 2007, 35(5/6): 541 – 550.
- [12] TSENG C T, LIAO C J. A particle swarm optimization algorithm for hybrid flow-shop scheduling with multiprocessor tasks [J]. *International Journal of Production Research*, 2008, 46(17): 4655 – 4670.
- [13] ENGIN O, DOYEN A. A new approach to solve hybrid flow shop scheduling problems by artificial immune system [J]. *Future Generation Computer Systems*, 2004, 20(6): 1083 – 1095.
- [14] KARABOGA D. *An idea based on honey bee swarm for numerical optimization* [EB/OL]. Turkey: Computer Engineering Department, Erciyes University, 2005.
- [15] KARABOGA D, AKAY B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems [J]. *Foundations of Fuzzy Logic and Soft Computing*, 2007, 4529(1): 789 – 798.
- [16] ZHANG C S, OUYANG D T, NING J X. An artificial bee colony approach for clustering [J]. *Expert Systems with Applications*, 2010, 37(7): 4761 – 4767.
- [17] PAN Q K, TASGETIREN M F, SUGANTHAN P N, et al. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem [J]. *Information Sciences*, 2010, 181(12): 2455 – 2468.
- [18] LI J Q, PAN Q K, GAO K Z. Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems [J]. *International Journal of Advanced Manufacturing Technology*, 2011, 55(1/2/3/4): 1159 – 1169.
- [19] 王凌, 周刚, 许焯, 等. 混合流水线调度研究进展 [J]. *化工自动化及仪表*, 2011, 38(1): 1 – 8.
(WANG Ling, ZHOU Gang, XU Ye, et al. Advances in the study on hybrid flow-shop scheduling [J]. *Control and Instrument in Chemical Industry*, 2011, 38(1): 1 – 8.)
- [20] BRAH S A, LUAN L L. Heuristics for scheduling in a flowshop with multiple processors [J]. *European Journal of Operational Research*, 1999, 113(1): 113 – 122.
- [21] 周辉仁, 唐万生, 魏颖辉. 柔性Flow-Shop调度的遗传算法优化 [J]. *计算机工程与应用*, 2009, 45(30): 224 – 226.
(ZHOU Huiran, TANG Wansheng, WEI Yinghui. Optimize flexible flow-shop scheduling using genetic algorithm [J]. *Computer Engineering and Applications*, 2009, 45(30): 224 – 226.)
- [22] 崔建双, 李铁克, 张文新. 混合流水车间调度模型及其遗传算法 [J]. *北京科技大学学报*, 2005, 27(5): 623 – 626.
(CUI Jianshuang, LI Tiek, ZHANG Wenxin. Hybrid flowshop scheduling model and its genetic algorithm [J]. *Journal of University of Science and Technology Beijing*, 2005, 27(5): 623 – 626.)
- [23] MONTGOMERY D C. *Design and Analysis of Experiments* [M]. Arizona: John Wiley & Sons, 2005.
- [24] LIU F, ZHANG X P, ZOU F X, et al. Immune clonal selection algorithm for hybrid flow-shop scheduling problem [C] // *Proceedings of Chinese Control and Decision Conference*. Guilin: IEEE, 2009: 2605 – 2609.
- [25] XU Y, WANG L. Differential evolution algorithm for hybrid flow-shop scheduling problem [J]. *Journal of Systems Engineering and Electronics*, 2011, 22(5): 794 – 798.

作者简介:

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

周刚 (1987–), 男, 硕士研究生, 主要研究方向为智能调度方法, E-mail: g-zhou09@mails.tsinghua.edu.cn;

许焯 (1987–), 男, 博士研究生, 主要研究方向为智能优化理论与算法, E-mail: xuye05@mails.tsinghua.edu.cn;

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