

# 人工蜂群算法求解混合约束流水车间调度问题

孙厚权 张其亮

(江苏科技大学 电气与信息工程学院 江苏 张家港 215600)

**摘要:** 流水车间调度问题是一类经典的组合优化问题,但传统的流水车间调度问题因忽视了不同工序间的缓冲约束,难以被应用于一些复杂的实际问题。据此,提出了一种不同工序具有不同缓冲约束的流水车间调度问题,并设计了离散人工蜂群算法 DABC(discrete artificial bee colony)进行求解。算法基于排列形式进行编码,以 PF-NEH(profile fitting & NEH)算法为基础构造初始解,提高初始种群初始解的质量;在雇佣蜂阶段,在迭代贪婪算法基础上提出了分段破坏迭代贪婪算法产生邻域个体;在观察蜂阶段,同时挑选较优解和较差解,并基于 Path-relinking 算法进一步挖掘搜索;在侦查蜂阶段,除了选择解的质量较差的个体被淘汰外,还设计了扰动策略使算法能跳出局部收敛。通过标准实例测试,验证了所提算法的有效性。

**关键词:** 离散人工蜂群算法;流水车间调度;最小化最大完工时间;混合约束

中图分类号: TP18

文献标识码: A

文章编号: 1673-629X(2019)03-0144-05

doi: 10.3969/j.issn.1673-629X.2019.03.030

## Artificial Bee Colony Algorithm for Flow Shop Scheduling Problem with Mixed Buffering Requirements

SUN Hou-quan ZHANG Qi-liang

(School of Electrical and Information Engineering, Jiangsu University of Science and Technology, Zhangjiagang 215600, China)

**Abstract:** The flow shop scheduling is a classical combinatorial optimization problem, but the traditional flow shop scheduling is difficult to be applied to some complex practical problems because it ignores the buffer constraints between different processes. Therefore, we propose a new flow shop scheduling problem with different stage buffering requirements, and put forward the discrete artificial bee colony (DABC) to resolve it. In DABC, the permutation based encoding schemes is designed, PF-NEH algorithm is used to construct the initial populations to improve the quality of populations. In employed bee phase, on the basis of iterative greedy algorithm, the iterated greedy algorithm with destruction operation of sections is proposed to generate the neighborhood individual. In onlooker bee phase, the better and worse solutions are selected together, and Path-relinking algorithm is proposed to make further search. In scout bee phase, in addition to eliminating worse individuals, perturbation strategy is designed to jump out of the local best. Effectiveness of the proposed algorithm is validated through a group of benchmark instances.

**Key words:** discrete artificial bee algorithm; flow shop scheduling; makespan; mixed buffering requirements

### 0 引言

流水车间调度问题是一类经典的组合优化问题,当机器数  $m > 2$  时,该类问题被证明为 NP-难问题<sup>[1]</sup>。传统的 FSS 问题是在假设相邻工序之间存在无限大的缓冲区这一前提下进行的,然而在实际生产过程中,工序之间的缓冲区往往是有限的甚至是不存在的。例如,在钢铁轧制过程中,为了防止金属成分遭到破坏,在钢板冷却之前,板坯的一系列加热过程必须连续进

行,工件在工序之间的加工不能有停顿,此类流水车间调度问题被称为无等待流水车间调度(no wait flow shop scheduling, NWFSS)问题;另外,在一些食品加工生产线,食品在一个工序加工完成后,若紧邻下一道工序的加工机器非空闲,则加工食品将被阻塞在当前工序的加工机器上,直到紧邻下一道工序的加工机器可用,该类问题被称为阻塞流水车间调度(blocking flow shop scheduling, BFSS)问题。

收稿日期: 2018-04-12

修回日期: 2018-08-16

网络出版时间: 2018-12-20

基金项目: 国家自然科学基金(11401262)

作者简介: 孙厚权(1997-)男,研究方向为数字图像处理与智能算法优化;张其亮,博士,副教授,通信作者,研究方向为智能算法、优化调度理论。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20181219.1542.066.html>

相对于传统的流水车间调度问题, NWFSS 问题和 BFSS 问题更具有实际意义。查阅文献知, 国内外已有不少专家学者对 NWFSS 问题、BFSS 问题进行了研究, 并取得了较好的成果。但是目前的研究大都假定所有工序间的约束是单一约束, 对工序间可能具有不同缓冲约束的调度问题研究相对较少, 目前类似的研究文献主要有: 文献[2]将传统 BFSS 问题定义为 RSB (release when starting blocking), 在此基础上提出了一种新的阻塞调度问题, 称之为 RCB (release when completing blocking), 并对工序具有两种混合阻塞约束的调度问题进行了分析和求解; 文献[3]针对工序间 RSB 约束和 RCB 约束及 RCB\* (release when completing blocking\*) 约束混合存在的流水车间调度问题设计了遗传算法进行求解, 算法虽然能得到较优解, 但是在样本空间较大时, 算法收敛速度较慢, 效率不高; 文献[4]利用人工蜂群算法求解与文献[3]相同的问题, 取得了较好的解; 文献[5]进一步进行拓展, 提出了 Wb (without blocking) 约束、RSB 约束和 RCB 约束及 RCB\* 约束混合存在的车间调度问题, 并利用遗传算法进行了求解; 文献[6]提出了阻塞约束和无等待约束混合的车间调度问题, 并设计了分支限界法进行求解; 文献[7]将列车调度问题归结为阻塞约束和与无等待约束混合的柔性车间调度问题, 并将问题建立了析取图模型, 设计了两阶段混合启发式算法进行求解, 结果满足列车运行要求。

虽然已有文献对工序具有不同约束需求的调度问题进行了研究, 但对于在流水车间调度中, 工序具有不同缓冲需求问题的研究还很少。因此, 在现有研究基础上, 文中设计了一个工序间具有无等待约束和阻塞约束两种不同需求的流水车间调度 (NW-BFSS) 问题, 并提出了求解的离散人工蜂群算法, 并通过实验证明算法的可行性和有效性。

## 1 NW-BFSS 问题模型

NW-BFSS 问题可描述为: 工件具有相同的加工工艺,  $n$  个工件按顺序在  $m$  道工序 (一道工序具有一台加工机器) 上进行加工; 某一时刻每台机器只能加工一个工件; 部分工序间具有阻塞约束, 部分工序间具有无等待约束。第  $i$  ( $0 < i \leq n$ ) 个工件在第  $j$  ( $0 < j \leq m$ ) 台机器上的加工时间  $P_{i,j}$  已知, 问题目标是找到一个工件的加工序列  $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ , 该加工序列具有的最大完工时间  $C_{\max}(\pi)$  最小。

令  $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$  为该问题的一个加工序列,  $R_{i,j}$  表示工件  $\pi_i$  释放机器  $j$  的时间,  $S_{i,j}$  表示工件  $\pi_i$  在机器  $j$  上的开始加工时间,  $C_{i,j}$  为工件  $\pi_i$  在机器  $j$  上的完工的时间。给出 NW-BFSS 问题的数学模型

如下:

$$\min C_{\max}(\pi) = \min C_{n,m} \quad (1)$$

s. t.

$$R_{i,j} \geq R_{i-1,j+1} \quad i \in \{2, 3, \dots, n\} \quad k \in \{1, 2, \dots, m\} \quad (2)$$

$$R_{i,j} = \max(R_{i,j-1} + P_{i,k}, R_{i-1,j+1}) \quad i = 2, 3, \dots, n \quad j = 2, 3, \dots, m-1 \quad (3)$$

$$S_{i,j} = R_{i,j-1} \quad j = \{2, 3, \dots, m\} \quad (4)$$

$$S_{i,j} = C_{i,j-1} \quad j = \{2, 3, \dots, m\} \quad (5)$$

$$C_{i,j} = R_{i,j} \quad (i = 1, 2, \dots, n \quad j = 1, 2, \dots, m) \quad (6)$$

其中, 式 1 为问题目标, 即最小化最大完工时间;

式 2 描述了加工工件在机器上加工时的阻塞约束; 式 3 表示阻塞约束下工件释放机器时间的计算方法; 式 4 表示阻塞约束下工件在机器上开始加工时间的计算方法; 式 5、式 6 表示工件在机器上加工时的无等待约束关系。

## 2 基本人工蜂群算法

人工蜂群算法 (ABC) 是根据蜜蜂寻找食物的过程演化而来, 是一种新的群体智能算法<sup>[8]</sup>。在 ABC 算法中, 问题的优化解对应食物源, 食物源的蜂蜜价值即为优化问题中的目标函数。算法根据分工将种群分为雇佣蜂、跟随蜂和侦查蜂三类, 雇佣蜂负责在食物源周围进行搜索, 以寻找到更好的食物源; 雇佣蜂经搜索后把食物源的信息带回, 守在蜂巢中的跟随蜂则根据概率选择较好的食物源进一步进行搜索; 经过若干周期后, 如果食物源没有更新, 则该雇佣蜂转变为侦查蜂, 在整个解空间中随机搜索新的食物源。基本 ABC 算法的主要步骤为:

(1) 随机初始化 SN 个食物源;

(2) 雇佣蜂阶段: 在食物源附近进行邻域搜索, 产生邻域解;

(3) 跟随蜂阶段: 选择较好的食物源进一步进行搜索;

(4) 侦查蜂阶段: 在解空间内随机选择新的食物源替代有限周期内没有更新的食物源;

(5) 判断算法是否终止。如果终止, 则输出最优解, 否则转步骤 2 执行。

## 3 求解 NW-BFSS 问题的离散人工蜂群算法

文献[9-10]等都利用人工蜂群解决了作业调度问题, 但纵观现有文献, 大都利用 ABC 算法求解连续优化问题, 为此文中对 ABC 算法进行了离散化以求解离散 NW-BFSS 问题。

### 3.1 问题编码

文中采用排列编码形式,每一个食物源表示问题的一个解,每个解都由  $n$  个不重复的工件构成,工件在解中的顺序代表了工件的加工次序。比如:有 6 个工件的 NW-BFSS 问题,食物源  $\{3, 2, 6, 4, 1, 5\}$  表示工件的加工顺序为:工件 3 首先加工,依次是工件 2, 6, 4, 1, 工件 5 最后加工。

### 3.2 初始解集构建

为了保证初始解的质量及多样性,借助 PF\_NEH( $\lambda$ ) 算法<sup>[11]</sup>构造部分初始优化解,剩余的解随机产生。初始解的产生策略如下:

- (1) for  $\lambda = 2$  to  $n$  利用 PF\_NEH( $\lambda$ ) 算法产生问题解,并将解按照目标值由小到大的顺序排列放入到队列  $\mu$ ;
- (2) 从队列  $\mu$  的队首依次取出  $k$  个解,并将此  $k$  个解作为初始解;
- (3)  $Cnt = k$ ;
- (4) 若  $Cnt = n$ , 则终止初始化过程,否则随机产生一个初始解。若该随机解不同于  $Cnt$  个解中的任何一个,则将解作为一个初始解,  $Cnt = Cnt + 1$ ; 否则,将该解忽略;
- (5) 跳转到步骤 4。

### 3.3 邻域结构

文中设计了三种邻域搜索的方法,邻域结构分别定义如下:

交换邻域,记为  $N_1$ 。产生邻域的策略为:对工件序列  $\pi$ , 随机选择一个位置  $i$ , 将位置  $i$  所对应的工件  $\pi_i$  与  $\pi$  中所有其他位置工件进行交换,选取目标值最小的序列替换  $\pi$ 。

插入邻域,记为  $N_2$ 。产生邻域的策略为:随机在个体  $\pi$  中选择两个位置  $a, b, a \neq b$ , 若  $a < b$ , 则将  $\pi_a$  插入到序列  $\pi$  中  $b$  的位置之前,可得到序列  $\pi' = \{\pi_1, \dots, \pi_{a-1}, \pi_{a+1}, \dots, \pi_a, \pi_b, \pi_{b+1}, \dots, \pi_n\}$ ; 若  $a > b$ , 则将  $\pi_a$  插入到序列  $\pi$  中  $b$  位置之前,可得到序列  $\pi' = \{\pi_1, \dots, \pi_a, \pi_b, \pi_{b+1}, \dots, \pi_{a-1}, \pi_{a+1}, \dots, \pi_n\}$ 。

组插入邻域,记为  $N_3$ 。产生邻域的策略为:随机在个体  $\pi$  中选择位置  $a$ , 将位置  $[a, a + \lambda]$  ( $\lambda > 0$ ) 所对应的工件作为一组工件子序列  $\pi'$ , 并从  $\pi$  中删除  $\pi'$  得到序列  $\pi''$ , 将  $\pi'$  插入到  $\pi''$  中所有可能位置,取目标值最小的序列替换  $\pi$ 。

### 3.4 信息交换策略

为了使蜂群在搜索过程中进行信息共享,设计了两种信息交换策略  $R_1, R_2$ , 定义如下:

$R_1$  即为 Path\_relinking 策略,其信息交换的步骤为:

- (1) 对于当前解  $\pi$ , 选择要进行信息交换的更优解  $\pi'$ , 定义  $k = 0$ ;
- (2) 若  $k = n$ , 转步骤 4; 否则选择  $\pi'$  中第  $k$  个位置工件  $\pi'_k$ , 在  $\pi$  中查找  $\pi'_k$ , 并记录  $\pi'_k$  在  $\pi$  中的位置  $k'$ , 若  $k' \neq k$ , 则将工件  $\pi'_k$  移动到  $\pi$  中位置  $k$ , 得到解  $\pi''$ , 并将  $\pi''$  放入队列  $L, \pi = \pi''$ ;
- (3)  $k = k + 1$ , 返回步骤 2;
- (4) 从  $L$  中选择目标值最小的解  $\pi_{\min}, \pi = \pi_{\min}$ 。

$R_2$  采用部分交叉策略进行信息交换,即在个体  $\pi_1, \pi_2$  中随机选取交叉区域,假定交叉区域中的工件序列为  $\pi'_1, \pi'_2$ , 定义集合  $S_1, S_2, S_1 = \{x | x \in \pi'_1 \cap x \notin \pi'_2\}, S_2 = \{x | x \in \pi'_2 \cap x \notin \pi'_1\}$ 。将  $\pi'_1, \pi'_2$  中的工件进行互换,并把个体  $\pi_1, \pi_2$  中重复的工件分别用  $S_1, S_2$  中的工件进行替换,从而得到交叉后的两个个体。

### 3.5 局部搜索策略

基于传统的迭代贪婪算法,设计了一种分段破坏的局部搜索策略,用于在给定解的周围进一步搜索优化解。该策略将工件序列分为若干段,基于每段进行破坏,破坏后执行构建操作,选择最优解,并基于该最优解进行下一分段的破坏,算法的具体描述如图 1 所示。

### 3.6 雇佣蜂阶段

雇佣蜂的主要任务是在给定的食物源上进一步进行搜索,找到更好的食物源。文中设计的离散 ABC 算

```

IG_DS( )算法:
输入: 当前解  $\pi$ , 分段数  $\mu$ 
输出: 更新后的解  $\pi_b$ 
Begin:
    令  $k=0$ ;
    步骤 1 若  $k < \mu$ , 执行步骤 2, 否则结束搜索
    步骤 2 在  $k$  分段内随机毁坏  $x$  个作业, 得到毁坏后的工件序列  $\pi'$ 
    步骤 3 执行 IG 算法的构建操作, 将  $x$  个毁坏作业依次插入到  $\pi'$  中所有可能位置, 得到最优序列  $\pi_{best}$ 
    步骤 4 若  $C\pi_{best} < C\pi, \pi_b = \pi_{best}, \pi = \pi_{best}$ 
    步骤 5 利用  $N_2$  及  $N_3$  邻域结构对  $\pi$  进行邻域搜索, 取最优的邻域解  $\pi_b$ , 若  $C\pi_b < C\pi$ , 则  $\pi_b = \pi_b, \pi = \pi_b, k = k + 1$ , 返回步骤 1
End
输出: 个体  $\pi_b$ 

```

图 1 局部搜索策略执行步骤

法中, 雇佣蜂的搜索步骤为:

(1) 对当前解集中每个食物源分配一个雇佣蜂;

(2) 对解  $\pi_i (0 \leq i < n)$ , 其对应的雇佣蜂按照 3.5 节中设计的分段破坏局部搜索策略, 对当前食物源进一步进行局部搜索, 得到最优解  $\pi'_i$ ;

(3) 用  $\pi'_i$  替换  $\pi_i$ 。

### 3.7 跟随蜂阶段

跟随蜂主要负责选择优质食物源, 并在食物源周围继续深入搜索。基本 ABC 算法采用轮盘赌方式<sup>[12]</sup>选择继续开采的食物源, 为了进一步搜索优化解并避免算法陷入局部最优, 采用按比例选择较优食物源和较差食物源的方式进行。跟随蜂的具体操作如下:

(1) 对所有解按照目标值由小到大顺序进行排列, 并放入队列  $L$ ;

(2) 从队列  $L$  的尾部选择  $\lambda'$  个解, 对当前解  $\pi_i$ , 从  $L$  中前  $l$  个解中随机选择一个解  $\pi'_i$ , 利用  $R_1$  策略将解  $\pi_i$  与  $\pi'_i$  进行信息交换, 得到解  $\pi''_i$ , 用  $\pi''_i$  替换  $\pi_i$ ;

(3) 从队列  $L$  中选择前  $\lambda$  个解, 将当前解  $\pi_i$  利用  $R_1$  或  $R_2$  策略与种群目前为止的最优解进行信息交换, 得到解  $\pi'_i$ , 用  $\pi'_i$  替换  $\pi_i$ 。

### 3.8 侦查蜂阶段

基本 ABC 算法中, 侦查蜂只有一个, 哪个食物源在指定的次数内未更新, 就转变为侦查蜂, 在解空间随机产生一个解替换原来的解。为了增加算法的全局搜索能力, 设计的侦查蜂过程为:

(1) 找到指定次数内未更新的所有解, 利用 PF\_NEH( $\lambda$ ) 算法随机产生一个解, 并替换原来的解;

(2) 对剩余的解, 利用  $N_1$  邻域结构进行扰动, 利用扰动后得到的解替换原解。

## 4 实验结果及分析

### 4.1 测试实例

算法基于 VC++6.0 实现, 在处理器为 Intel 2-

Core-i3-3240、3.4 GHz, 内存为 4 G 的 PC 机上运行。为了验证算法的有效性, 采用表 1 的实例数据进行测试。该实例共有 20 个工件, 5 道工序, 表中数字代表该工件分别在 5 道工序机器上的加工时间。

表 1 实例数据

工件编号	加工时间	工件编号	加工时间
1	4 6 2 1 3	11	2 1 3 4 3
2	7 3 2 6 4	12	3 1 2 4 6
3	2 1 5 3 6	13	2 1 3 3 2
4	5 2 1 2 8	14	3 2 1 2 2
5	7 3 1 5 8	15	5 3 1 2 6
6	3 5 7 2 4	16	1 2 5 4 2
7	6 8 2 7 1	17	2 3 5 4 1
8	2 5 7 9 4	18	1 5 6 3 4
9	1 4 3 6 2	19	2 3 1 6 4
10	9 2 5 7 1	20	4 2 3 5 1

工序间的约束如表 2 所示, 无等待约束和阻塞约束混合存在。

表 2 工序间约束关系

工序	加工约束
1	无等待约束
2	阻塞约束
3	无等待约束
4	阻塞约束

设定迭代次数为 100, 解集大小为 10, 雇佣蜂数量  $E = 10$ , 跟随蜂数量  $O = 10$ 。根据所设计的离散人工蜂群算法求解该 NW-BFSS 问题, 计算得到最优解的序列为: {10, 12, 3, 18, 19, 2, 8, 14, 4, 15, 7, 17, 6, 9, 5, 11, 1, 16, 0, 13}, 调度甘特图如图 2 所示。由图 2 可以看出, 在第 1 道工序和第 3 道工序满足无等待约束, 在第 2 道工序和第 4 道工序工件加工时满足阻塞约束, 图中黑色矩形为阻塞时间。可见, 算法求解 NW-BFSS 问题是可行的、有效的。

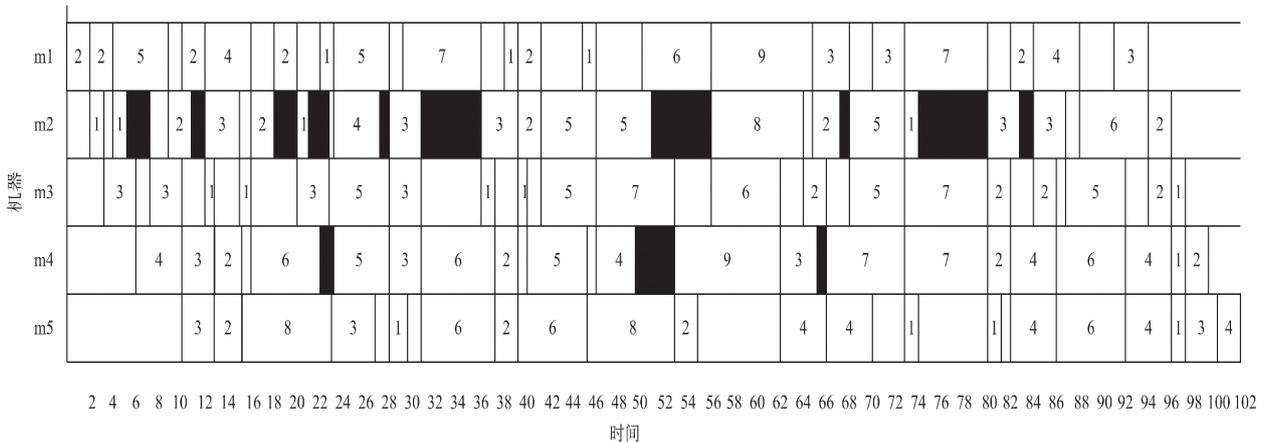


图 2 调度甘特图

为了进一步验证算法的求解质量,以 Taillard<sup>[13]</sup> 经典实例作为测试数据,测试问题规模从 20×5 到 200×20。由于没有相同问题的研究结果,故在比较算法有效性时对问题做了特殊化,将所有工序约束归为无等待约束,从而使文中算法与现有文献中的算法具有可比性。

算法中设置的关键参数为:解集大小为 10,雇佣蜂数量  $E = 10$ ,跟随蜂数量  $O = 10$ ;算法的终止条件为迭代的次数,迭代的次数在  $n = 20$  时设置为  $G = 2\ 000$ ,在  $n = 50$  时设置为  $G = 5\ 000$ ,在  $n = 100$  时设置为  $G = 15\ 000$ 。

#### 4.2 结果比较及分析

以 ARPD(average relative percentage deviation) 作为评价指标,将算法与其他解决该问题的算法进行比较,计算公式为:

$$ARPD = \frac{1}{M} \sum_{i=1}^M \frac{C_i - C^*}{C^*} \times 100 \quad (7)$$

其中,  $C_i$  为算法在第  $i$  次实验中得到的解;  $C^*$  为文献 [14] 的求解结果;  $M$  为计算次数。

算法计算得到的 ARPD 越小,表明所求得解越优。表 3 为文中算法与 HGA<sup>[14]</sup> 算法、TMIIG<sup>[15]</sup> 算法、HDTPL<sup>[16]</sup> 算法在求解 NW-BFSS 时的比较结果。

表 3 算法比较结果

问题规模 $n \times m$	HGA ( ARPD)	TMIIG ( ARPD)	HDTPL ( ARPD)	文中算法 ( ARPD)
20 × 5	0.03	0.00	0.00	0.00
20 × 10	0.06	0.00	0.00	0.00
20 × 20	0.03	0.00	0.00	0.00
50 × 5	0.55	0.18	0.24	0.06
50 × 10	0.47	0.06	0.19	0.04
50 × 20	0.47	0.05	0.18	0.03
100 × 5	0.84	0.42	0.41	0.21
100 × 10	0.85	0.12	0.33	0.05
100 × 20	0.91	0.10	0.41	0.06
200 × 10	1.68	0.39	0.62	0.30
200 × 20	1.52	0.21	0.63	0.16

从表 3 可以看出,在较小规模  $n = 20$  时,文中算法与其他算法一致,在  $n = 50$  和  $n = 100$  规模时,文中算法要优于其他算法。

#### 4.3 收敛性分析

为了体现算法的收敛性,给出了算法在求解  $n = 100$  时实例 Ta061 的收敛曲线,如图 3 所示。可以看出,算法在迭代初期收敛速度较快,但随着迭代次数的增加收敛变缓,甚至出现局部收敛。而算法在局部收敛后,能跳出局部搜索,最终得到较优化的解。

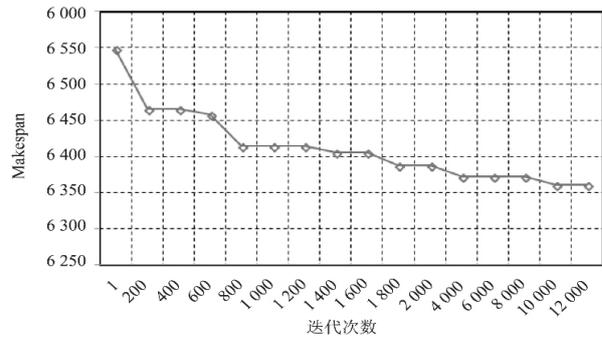


图 3 Ta061 实例收敛曲线

## 5 结束语

在传统流水车间调度问题、阻塞流水车间调度问题及无等待流水车间调度问题的基础上,提出了一个工序间对阻塞约束和无等待约束有不同需求的流水车间调度问题,并提出了求解的离散人工蜂群算法。通过 PF\_NEH( $\lambda$ ) 算法得到初始优质食物源,利用雇佣蜂阶段的分段破坏迭代贪婪算法,跟随蜂阶段的 Path\_relinking 算法,对解空间进行深入搜索。利用标准数据进行测试发现,算法具有较高的求解质量。单目标的流水车间调度相对简单,多目标的流水车间调度问题及混合流水车间调度具有更高的复杂度,因此下一步将继续改进算法,使其能解决更复杂的调度问题。

#### 参考文献:

- [1] GAREY M R, JOHNSON D S, SETHI R. The complexity of flow shop and job shop scheduling [J]. Mathematics of Operations Research, 1976, 1(2): 117-129.
- [2] MARTINEZ S, DAUZE`RE-PERE`S S, GUERET C, et al. Complexity of flowshop scheduling problems with a new blocking constraint [J]. European Journal of Operational Research, 2006, 169: 855-864.
- [3] WAJDI T, CHRISTOPHE S, SAUER N. Heuristics and meta-heuristics for mixed blocking constraints flowshop scheduling problems [J]. Computers & Operations Research, 2012, 39(11): 2520-2527.
- [4] MOSTAFA K, VAHID R. A bee colony optimization approach for mixed blocking constraints flow shop scheduling problems [J]. Mathematical Problems in Engineering, 2015, 2015(3): 1-10.
- [5] SAUVEY C, SAUER N, TRABELSI W. Job-shop scheduling with mixed blocking constraints between operations [C]// Modelling, computation and optimization in information systems and management sciences. [s. l.]: [s. n.], 2015: 331-342.
- [6] MASCIS A, PACCIARELLI D. Job-shop scheduling with blocking and no-wait constraints [J]. European Journal of Operational Research, 2002, 143(3): 498-517.

(下转第 153 页)

小的企业,应该改进经营管理理念,否则就会被强者吃掉。

(3) 官方在某一时刻分别在 $\overline{RCN}$ ,  $\underline{RCN}$ 中以最大加权重为源节点,以加权重由大到小为查找条件,沿着网络的最短路径,搜集节点的属性 $\overline{a}$ ,  $\underline{a}$ ,当节点数达到样本容量收集数据结束,构建模型得到风险评估结果。

#### 4 结束语

文中构建的粗糙网络风险评估模型具有以下优点和创新性:考虑了粗糙复杂网络的不确定信息和网络结构的不可分辨性,更接近于实际的复杂网络;从粗糙复杂网络效益风险存在的根源—管理者和商家的努力水平来分析问题,建立二级风险评估指标体系,与现有的第三方支付平台效益评价指标体系相比,指标的数目大幅度减少,考虑了产生效益的主观原因—努力水平;在数据收集时,运用了粗糙网络度分布的特性,使用了最大加权重节点搜索法,减少了操作的盲目性,为平台效益风险的评估节约了成本。

#### 参考文献:

- [1] 曹黎侠,黄光球,况湘玲.基于粗糙集理论的第三方支付平台的效益问题[J].统计与信息论坛,2016,31(1):62-68.
- [2] VINODHINI G, CHANDRASEKARAN R M. Measuring the quality of hybrid opinion mining model for e-commerce application[J]. Measurement, 2014, 55: 101-109.
- [3] DU Gengshen, RUHE G. Two machine-learning techniques for mining solutions of the release planner decision support system[J]. Information Sciences, 2014, 259: 474-489.
- [4] LIU Jun, XIONG Qingyu, SHI Weiren, et al. Evaluating the importance of nodes in complex networks[J]. Physica A: Statistical Mechanics and Its Applications, 2016, 452: 209-219.
- [5] 陈维滨.电子商务第三方支付模式的选择及风险防范[D].泉州:华侨大学,2010.
- [6] ZHAO Shu, ZHANG Ling, XU Xiansheng, et al. Hierarchical description of uncertain information[J]. Information Sciences, 2014, 268: 133-146.
- [7] 洪之旭,陈浩,程亮.基于大数据的社会治理数据集成及决策分析方法[J].清华大学学报:自然科学版,2017,57(3):264-269.
- [8] SHALAGINOV A, FRANKE K. Big data analytics by automated generation of fuzzy rules for network[J]. Applied Soft Computing, 2017, 52: 359-375.
- [9] 汪小帆,李翔,陈关荣.复杂网络理论及其应用[M].北京:清华大学出版社,2005.
- [10] 曹黎侠,黄光球,李艳.基于网络基的粗糙复杂网络决策方法及应用[J].计算机科学与探索,2016,10(11):1601-1613.
- [11] CAO Lixia, HUANG Guangqiu. Concept design and construction algorithm of rough complex networks[J]. Journal of Intelligent & Fuzzy Systems, 2017, 33(3): 1441-1451.
- [12] CAO Lixia, HUANG Guangqiu. Study on mixed strategy Nash equilibrium based on rough set theory and particle swarm optimization[J]. The Open Cybernetics & Systemic Journal, 2015, 9: 88-92.
- [13] FERRETTI S. Gossiping for resource discovering: an analysis based on complex network theory[J]. Future Generation Computer Systems, 2013, 29: 1631-1644.
- [14] 曹黎侠,黄光球.第三方支付平台交易的粗糙博弈模型及算法研究[J].运筹与管理,2016,25(5):46-52.
- [15] 余鹰,苗夺谦,赵才荣,等.基于粗糙集的多标记决策系统知识获取方法[J].计算机科学与探索,2015,9(1):94-104.
- [7] LIU Shiqiang, KOZAN E. Scheduling trains with priorities: a no-wait blocking parallel-machine job-shop scheduling model[J]. Transportation Science, 2011, 45(2): 175-198.
- [8] KARABOGA D. An idea based on honey bee swarm for numerical optimization[R]. [s. l.]: [s. n. ] 2005.
- [9] 宋栓军,杨佩莉,石雯丽.基于人工蜂群算法的柔性工艺与车间调度集成优化[J].计算机应用,2017,37(2):523-529.
- [10] 李修琳,鲁建厦,柴国钟,等.混流装配排序问题的改进人工蜂群优化[J].计算机集成制造系统,2011,17(12):2599-2609.
- [11] PAN Q K, WANG L, SANG H Y, et al. A high performing memetic algorithm for the flowshop scheduling problem with blocking[J]. IEEE Transactions on Automation Science and Engineering, 2013, 10(3): 741-756.
- [12] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. Journal of Global Optimization, 2007, 39(3): 459-471.
- [13] TAILLARD E. Benchmarks for basic scheduling problems[J]. European Journal of Operational Research, 1993, 64(2): 278-285.
- [14] JARBOUI B, EDDALY M, SIARRY P. A hybrid genetic algorithm for solving no-wait flow shop scheduling problems[J]. International Journal of Advanced Manufacturing Technology, 2014, 54(9-12): 1129-1143.
- [15] DING J Y, SONG S, GUPTA J N D, et al. An improved iterated greedy algorithm with a Tabu-based reconstruction strategy for the no-wait flow shop scheduling problem[J]. Applied Soft Computing, 2015, 30(5): 604-613.
- [16] SHAO W S, PI D C, SHAO Z S. A hybrid discrete optimization algorithm based on teaching-probabilistic learning mechanism for no-wait flow shop scheduling[J]. Knowl-

(上接第148页)