

# A Hybrid EDA with GA for the Permutation Flow Shop Scheduling Problem

Zhuzhi Liu, Kai Wang

School of Economics and Management, Wuhan University, Wuhan Hubei  
Email: [lzz\\_cau\\_wuhu@163.com](mailto:lzz_cau_wuhu@163.com)

Received: Feb. 6<sup>th</sup>, 2015; accepted: Feb. 18<sup>th</sup>, 2015; published: Feb. 26<sup>th</sup>, 2015

Copyright © 2015 by authors and Hans Publishers Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

The permutation flow shop scheduling problem is a classical combinatorial optimization in industrial engineering. Population-based evolutionary algorithms (EA) are the common methods to solve this problem. As a novel EA, estimation of distribution algorithm (EDA) directs the algorithm search towards good solutions by statistical learning. However, this algorithm may trap into the local optimal and lead to the premature convergence. To overcome the drawback of EDA, this paper incorporates EDA with GA to address the PFSP. The participation rates of EDA and GA are adaptively regulated by fuzzy logic controller. The experiment results on the benchmarks validate the efficiency of the proposed algorithm.

## Keywords

Permutation Flow Shop Scheduling Problem, Estimation of Distribution Algorithm, Genetic Algorithm, Fuzzy Logic Controller

---

## 基于EDA-GA的置换流水车间调度算法

刘祝智, 王 恺

武汉大学经济与管理学院, 湖北 武汉  
Email: [lzz\\_cau\\_wuhu@163.com](mailto:lzz_cau_wuhu@163.com)

收稿日期: 2015年2月6日; 录用日期: 2015年2月18日; 发布日期: 2015年2月26日

## 摘要

置换流水车间调度问题是工业工程中经典的组合优化问题，一般采用智能优化算法来求解该问题。作为一种新颖的优化算法，分布估计算法主要使用统计学习的方法指导搜索过程。然而，这种算法容易陷入到局部最优而出现过早收敛的现象。本文将分布估计算法与遗传算法结合，通过模糊逻辑控制来调节两种算法生成个体的比例。该算法有利于保持种群的多样性，避免了过早收敛。以Car类和Rec类算例进行测试，实验结果证实了本文所提出的混合算法的有效性。

## 关键词

置换流水车间调度，分布估计算法，遗传算法，模糊逻辑控制

## 1. 前言

置换流水车间调度问题(Permutation Flow Shop Scheduling Problem, PFSP)是工业工程中经典的规划问题。它在制造系统、生产线组装和信息设备服务上都有着很重要的运用[1]。在该问题中，有  $n$  个待加工工件需要在  $m$  台机器上进行加工，问题的目标是寻找这  $n$  个工件在机器上的加工顺序，从而使得工件的某个调度指标达到最优，最常用的指标为工件的总完工时间(makespan)最短。

PFSP 最早由 Johnson 于 1954 年进行研究[2]，具有 NP 难性质[3]。常用的求解方法主要有数学规划，启发式方法和基于人工智能的元启发式算法[4]。数学规划适用于小规模问题，无法解决计算复杂性高的 PFSP 问题，而启发式方法求解时间虽短，但是不能保证解的质量。随着计算智能的发展，基于人工智能的元启发式优化算法成为研究的重点，目前应用得较多的算法有模拟退火算法、遗传算法、蚁群算法、粒子群算法等等。

分布估计算法(Estimation of Distribution Algorithm, EDA)是近些年来最为热门的智能优化算法之一，它是在遗传算法(Genetic Algorithm, GA)的基础上改进而来的，最初由 H. Muhlenbein 在 1996 年提出[5]。分布估计算法不进行交叉变异操作，而是用统计学习的方法，记录选择后的个体在解空间上的分布并建立一个概率模型，通过概率模型生成下一代个体。作为一种全新的搜索策略，分布估计算法侧重于在宏观层面上指导种群的进化，得到了学术界的广泛关注。

EDA 在解空间上具有较强的搜寻能力，然而对解的开发利用不足，从而导致对问题求解时很容易出现过早收敛的现象，尤其是在处理高维优化问题时[6] [7]。对此，学术界提出了解决方法，包括提高 EDA 在运行时种群中个体的多样性、改进概率模型生成个体的机制以及将 EDA 与其他优化算法结合等等[7]。本文将设计一种 EDA 与 GA 结合的混合算法来求解 PFSP 问题，混合算法通过 EDA 的概率模型和 GA 的交叉变异操作两种方式来生成个体，并引入模糊控制理论[8]来自适应调节两种算法生成个体的比例。

## 2. 置换流水车间调度问题

PFSP 问题有如下规定：

- 1)  $n$  台工件在  $m$  台机器上加工
- 2) 每个工件以相同的顺序在  $m$  台机器上加工
- 3) 每个工件在每台机器上的加工时间是预先确定的
- 4) 每台机器只能同时加工一个工件

令  $\Omega = \{J_1, \dots, J_n\}$  代表所有工件， $\pi = (\pi_1, \pi_2, \dots, \pi_n)$  为一个待加工工件的排列，其中  $\pi_i \in \Omega$  ( $i=1, 2, \dots, n$ )，则以总完工时间最短为优化目标的 PFSP 数学模型描述如下：

$$\min z = \min \{C(\pi_n, m)\}$$

$$s.t. \begin{cases} C(\pi_1, 1) = t_{\pi_1, 1} \\ C(\pi_1, k) = C(\pi_1, k-1) + t_{\pi_1, k}, k = 2, \dots, m \\ C(\pi_j, 1) = C(\pi_{j-1}, 1) + t_{\pi_j, 1}, j = 2, \dots, n \\ C(\pi_j, k) = \max(C(\pi_{j-1}, k), C(\pi_j, k-1)) + t_{\pi_j, k}, j = 2, \dots, n; k = 2, \dots, m \end{cases}$$

其中， $m$  表示机器数， $n$  表示工件数， $t_{i,j}$  为工件  $i$  在机器  $j$  的加工完成时间。

### 3. 算法设计

#### 3.1. 种群初始化

种群包括 PS 个个体，初始时利用经典的 NEH [9] 算法产生 1 个个体，采用随机初始化的方法生成 PS-1 个个体，以充分利用解空间的分布信息。

#### 3.2. 选择

根据事前确定的加工时间表计算出所有个体的总完工时间  $C_{\max}$ ，显然  $C_{\max}$  越小个体的质量就越好，据此可将评价个体好坏的适应度函数(fitness function)设为  $1/C_{\max}$ 。使用轮盘赌法(Roulette Wheel Selection)对个体进行选择，每个个体被选择的概率大小与其适应度值正相关，某个体的总完工时间越短，相应的  $C_{\max}$  越小，其适应度值就越高，该个体被选中的概率也就越大。

#### 3.3. 概率模型

概率模型是 EDA 的核心，它指导着 EDA 在解空间上的搜索，概率模型一般为  $n \times n$  矩阵  $P$  [10]， $p_{ij}$  代表工件  $j$  在解的第  $i$  位上或之前出现的概率，初始时  $p_{ij} = 1/n$ ，之后每一次迭代后  $p_{ij}$  计算公式如下：

$$p_{ij}(l+1) = (1-\alpha)p_{ij}(l) + \frac{\alpha}{i \times SP\_Size} \sum_{k=1}^{SP\_Size} I_{ij}^k, \forall i, j$$

$\alpha$  为设置的概率模型的学习速率， $I_{ij}^k$  代表第  $k$  个个体的如下特征

$$I_{ij}^k = \begin{cases} 1, & \text{若工件 } j \text{ 在解的第 } i \text{ 位上或之前出现} \\ 0, & \text{其他} \end{cases}$$

#### 3.4. 局部搜索

对概率模型采样即可得到新一代的个体，对个体进行局部搜索可以提高 EDA 的性能[11]，为了减少计算开销，本文将对较好的个体进行局部搜索，分别有如下三种搜索方法：

插入：选择一个工件并随机插入到某一位置

交换：随机选择两个工件并交换其所在位置

倒置：随机选择两个工件，将这两个工件之间的序列反转

#### 3.5. 交叉算子

本文采用的交叉算子为次序保留交叉(order preserved crossover)。例如，亲代个体为 {2, 3, 5, 1, 4, 9, 8, 6, 7, 10} 和 {1, 2, 4, 5, 6, 7, 8, 3, 9, 10}，在交叉过程中保留的片段为 {4, 9, 8, 6}，则生成的子代个体为 {1, 2, 5,

7, 4, 9, 8, 6, 3, 10} 和 {2, 3, 4, 5, 6, 1, 8, 7, 9, 10}, 如图 1 所示。

### 3.6. 变异算子

本文选取的变异算子为移码变异(shift move mutation), 例如, 变异前的个体为{6, 8, 9, 10, 7, 4, 3, 1, 2, 5}, 选择 7, 8 这两个位点进行变异, 变异后个体为{6, 9, 10, 7, 8, 4, 3, 1, 2, 5}, 如图 2 所示。

### 3.7. 模糊逻辑控制

混合算法中 EDA 和 GA 生成个体的比例是影响算法性能的关键, 传统的算法比例混合方法主要包括固定比例和动态比例两种。使用固定比例时, 比例值将在整个算法搜索过程中保持不变。这种方法需要进行试验来确定合适的比例值, 其缺点在于为寻找到最佳比例值所需进行的试验多, 耗时长。而动态比例调节则只需预先确定一个比例的初始值, 而在运行过程中会根据搜索情况调节比例。调节方式又可以分为两种: 一种是应用传统的启发式规则控制算法生成个体的比例, 这些规则可以用确定的数学公式表示; 而另一种是用人工智能技术自适应调整生成个体的比例, 最常见的是将模糊逻辑应用于比例调节, 能根据算法性能的变化来实现比例控制[12]。为了使混合算法具有优良的适应性, 本文采用模糊逻辑控制来进行比例调节: 在 EDA 表现良好时提高 EDA 生成个体的比例发挥其全局搜索的优势, 在 EDA 求得解的质量下降时提高 GA 生成个体的比例, 以避免出现局部最优。

模糊逻辑控制[8] [13] [14]的步骤如下:

1) 计算出种群中全部个体的总完工时间平均值  $average(t)$ , 令  $\Delta aver(t) = average(t) - average(t-1)$ ,  $\Delta aver(t-1) = average(t-1) - average(t-2)$ 。

2) 将  $\Delta aver(t-1)$ ,  $\Delta aver(t)$  标准化至  $[-1.0, 1.0]$ , 所使用的公式如下:

$$-1 + \frac{\Delta aver(t) - \min(\Delta aver(t))}{\max(\Delta aver(t)) - \min(\Delta aver(t))} \times 2$$

3) 根据隶属度函数(membership function, 见图 3)将标准化后的  $\Delta aver(t-1)$ ,  $\Delta aver(t)$  进行模糊处理。

(NR 是 negative larger 缩写, NL 为 negative large, NM 为 negative medium, NS 为 negative small, ZS 为 zero, PS 为 positive small, PM 为 positive medium, PL 为 positive large, PR 为 positive larger)。

4) 得到  $\Delta aver(t-1)$ ,  $\Delta aver(t)$  的模糊值后结合模糊判断规则(fuzzy decision table, 见表 1)确定模糊输出量  $Z(i, j)$ 。

5) 利用输出量  $Z(i, j)$  的隶属度函数(见图 4)和去模糊化规则(defuzzification table, 见表 2)确定最终 EDA 生成个体的比例的变化量  $d\_rate$ , 则下一代 EDA 所生成个体的比例为  $rate(t+1) = rate(t) + d\_rate$ 。

(NR 是 negative larger 缩写, NL 为 negative large, NM 为 negative medium, NS 为 negative small, ZS 为 zero, PS 为 positive small, PM 为 positive medium, PL 为 positive large, PR 为 positive larger)。

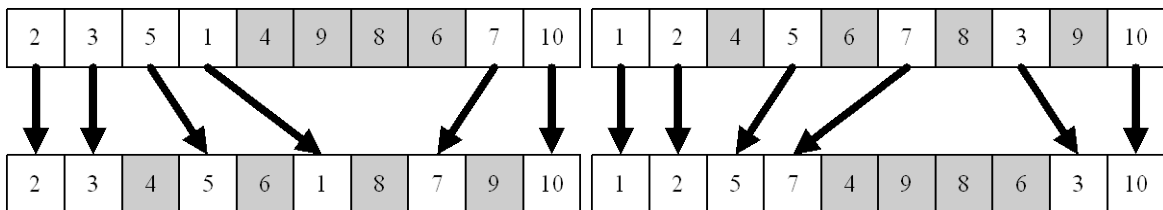


Figure 1. Order preserved crossover  
图 1. 次序保留交叉

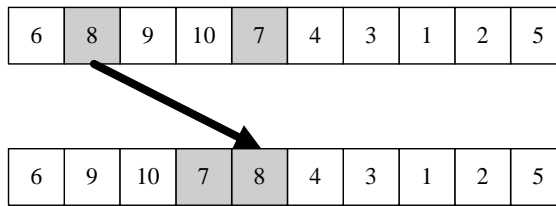


Figure 2. Shift move mutation  
图 2. 移码变异

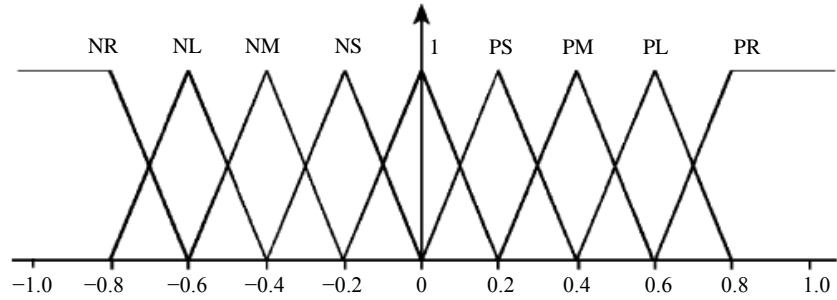


Figure 3. Membership function for inputs  
图 3. 输入量的隶属度函数

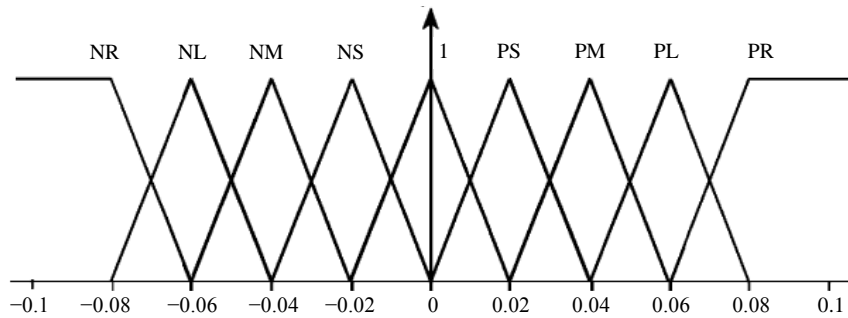


Figure 4. Membership function for outputs  
图 4. 输出量的隶属度函数

Table 1. Fuzzy decision table  
表 1. 模糊判断规则

|                     |    | $\Delta a_{ver}(t-1)$ |    |    |    |    |    |    |    |    |
|---------------------|----|-----------------------|----|----|----|----|----|----|----|----|
|                     |    | NR                    | NL | NM | NS | ZE | PS | PM | PL | PR |
| $\Delta a_{ver}(t)$ | NR | NR                    | NL | NL | NM | NM | NS | NS | ZE | ZE |
|                     | NL | NL                    | NL | NM | NM | NS | NS | ZE | ZE | PS |
|                     | NM | NL                    | NM | NM | NS | NS | ZE | ZE | PS | PS |
|                     | NS | NM                    | NM | NS | NS | ZE | ZE | PS | PS | PM |
|                     | ZE | NM                    | NS | NS | ZE | ZE | PS | PS | PM | PM |
|                     | PS | NS                    | NS | ZE | ZE | PS | PS | PM | PM | PL |
|                     | PM | NS                    | ZE | ZE | PS | PS | PM | PM | PL | PL |
|                     | PL | ZE                    | ZE | PS | PS | PM | PM | PL | PL | PR |
|                     | PR | ZE                    | PS | PS | PM | PM | PL | PL | PR | PR |

Table 2. Defuzzification table  
表 2. 去模糊化规则

|          |    | <i>j</i> |    |    |    |    |    |    |   |   |
|----------|----|----------|----|----|----|----|----|----|---|---|
|          |    | -4       | -3 | -2 | -1 | 0  | 1  | 2  | 3 | 4 |
| <i>i</i> | -4 | -4       | -3 | -3 | -2 | -2 | -1 | -1 | 0 | 0 |
|          | -3 | -3       | -3 | -2 | -2 | -1 | -1 | 0  | 0 | 1 |
|          | -2 | -3       | -2 | -2 | -1 | -1 | 0  | 0  | 1 | 1 |
|          | -1 | -2       | -2 | -1 | -1 | 0  | 0  | 1  | 1 | 2 |
|          | 0  | -2       | -1 | -1 | 0  | 0  | 1  | 1  | 2 | 2 |
|          | 1  | -1       | -1 | 0  | 0  | 1  | 1  | 2  | 2 | 3 |
|          | 2  | -1       | 0  | 0  | 1  | 1  | 2  | 2  | 3 | 3 |
|          | 3  | 0        | 0  | 1  | 1  | 2  | 2  | 3  | 3 | 4 |
|          | 4  | 0        | 1  | 1  | 2  | 2  | 3  | 3  | 4 | 4 |

#### 4. EDA-GA 混合算法

EDA-GA 混合算法步骤如下：

1) 种群和概率模型的初始化

产生初始种群，迭代次数  $t = 1$ ，概率模型  $P$  中  $p_{ij} = 1/n$

2) 对种群个体进行评价并选择出优势个体

以轮盘赌法选择出用以更新 EDA 概率模型的优势个体和待进行交叉变异操作的父代个体

3) 更新概率模型并对概率模型取样生成个体

对优势个体进行统计学习完成概率模型的更新，然后对概率模型抽样产生  $PS$  个个体，局部搜索后，把最好的  $rate(t) * PS$  个个体加入到下一代种群， $rate(t)$  为当前 EDA 所生成个体的比例

4) 交叉操作和变异操作

对父代分别进行交叉操作和变异操作，共产生  $(1 - rate(t)) * PS$  个个体，将这些个体加入到下一代种群中

5) 模糊逻辑控制调节比例

新一代种群生成后，将种群平均完工时间与上一代进行比较，得到模糊输入量，根据模糊判断规则确定下一次迭代时 EDA 所生成个体的比例  $rate(t+1)$

6) 终止条件判断

若满足终止条件，输出此时得到的最优解；否则， $t = t + 1$ ，进入步骤 2)。

### 5. 实验结果

#### 5.1. 参数设置

将 EDA-GA 混合算法的参数设为种群大小  $PS = 200$ ，迭代次数  $iteration\_times = 300$ ，优势个体所占比例为  $superior\_rate = 0.2$ ，GA 变异比例  $mutation\_rate = 0.1$ ，EDA 初始生成个体的比例  $rate = 1$ ，概率模型学习速率  $\alpha = 0.2$ 。

#### 5.2. 结果分析

为验证混合算法的效果，本文选用 OR-library 中的 Car1~Car8，Rec01~Rec41，共 29 个算例来进行测试，每个算例用 matlab 在计算机上独立运行 10 次，并与 GA，EDA 的结果进行比较。测试结果如表 3

**Table 3.** Computational results for benchmarks  
**表 3.** 算例仿真结果

| 算例    | n m   | EDA  |      | GA   |      | EDA-GA |      |
|-------|-------|------|------|------|------|--------|------|
|       |       | BRE  | ARE  | BRE  | ARE  | BRE    | ARE  |
| Car1  | 11 5  | 0    | 0    | 0    | 0    | 0      | 0    |
| Car2  | 13 4  | 0    | 0    | 0    | 0.59 | 0      | 0    |
| Car3  | 12 5  | 0    | 0    | 0    | 0    | 0      | 0    |
| Car4  | 14 4  | 0    | 0    | 0    | 0    | 0      | 0    |
| Car5  | 10 6  | 0    | 0.12 | 0    | 0.79 | 0      | 0    |
| Car6  | 8 9   | 0    | 0.54 | 0    | 1.63 | 0      | 0.38 |
| Car7  | 7 7   | 0    | 0    | 0    | 0    | 0      | 0    |
| Car8  | 8 8   | 0    | 0    | 0    | 0    | 0      | 0    |
| Rec01 | 20 5  | 0    | 0.47 | 0    | 0.58 | 0      | 0.16 |
| Rec03 | 20 5  | 0    | 0.09 | 0.18 | 0.22 | 0      | 0.16 |
| Rec05 | 20 5  | 0.24 | 0.52 | 0.24 | 0.27 | 0.24   | 0.24 |
| Rec07 | 20 10 | 1.15 | 1.34 | 0    | 2.27 | 0      | 0.07 |
| Rec09 | 20 10 | 0.72 | 2.06 | 2.6  | 2.62 | 0.33   | 1.67 |
| Rec11 | 20 10 | 0    | 1.3  | 0    | 1.55 | 0      | 0.65 |
| Rec13 | 20 15 | 0.93 | 1.69 | 1.81 | 1.96 | 0      | 1.52 |
| Rec15 | 20 15 | 1.32 | 2    | 0.05 | 0.85 | 0      | 0.85 |
| Rec17 | 20 15 | 1.5  | 2.59 | 2.12 | 2.2  | 0      | 1.59 |
| Rec19 | 30 10 | 1.37 | 1.94 | 0.42 | 1.14 | 0      | 0.75 |
| Rec21 | 30 10 | 1.12 | 1.77 | 0.05 | 1.35 | 0      | 1.11 |
| Rec23 | 30 10 | 1.97 | 2.87 | 0.44 | 1.91 | 0      | 1.44 |
| Rec25 | 30 15 | 1.48 | 2.24 | 1.33 | 1.98 | 0      | 1.29 |
| Rec27 | 30 15 | 1.12 | 1.48 | 0.67 | 1.71 | 0      | 1.14 |
| Rec29 | 30 15 | 0.77 | 1.02 | 1.07 | 1.07 | 0      | 0.83 |
| Rec31 | 50 10 | 1.5  | 1.97 | 0.45 | 1.57 | 0      | 1    |
| Rec33 | 50 10 | 0.93 | 1.62 | 0.32 | 1.3  | 0      | 0.77 |
| Rec35 | 50 10 | 0.37 | 0.47 | 0.49 | 0.49 | 0      | 0.19 |
| Rec37 | 75 20 | 1.15 | 1.41 | 0.52 | 0.78 | 0      | 0.61 |
| Rec39 | 75 20 | 1.47 | 1.62 | 0    | 0.75 | 0.31   | 0.67 |
| Rec41 | 75 20 | 0.92 | 1.2  | 0    | 0.58 | 0.25   | 0.68 |

所示。其中， $BRE = \frac{C_{\max} - C^*}{C^*} \times 100$ 、 $ARE = \frac{\sum_{i=1}^{10} (C_{\max} - C^*)}{10 \times C^*} \times 100$  为每种算法求得的最优解  $C_{\max}$  与三种算法测试所得的最好解  $C^*$  的相对偏差百分比的最小值和平均值。

从表 3 的实验结果可以看出，对测试问题 Car1~Car8 和 Rec01~Rec37，本文设计的 EDA-GA 混合算法 ARE 与 BRE 均优于 EDA 和 GA，说明 GA 的引入使得 EDA 的优化性能有了很大的改进。对于 Rec39、Rec41，混合算法 BRE 不如 GA，说明优化性能稍弱于 GA，但相比 EDA 解的质量有显著提高。因此总体而言，EDA-GA 混合算法的性能是要强于 GA 和 EDA。

## 6. 结论

本文设计了一种 EDA-GA 混合算法对以最小化总完工时间为优化目标的 PFSP 问题进行了求解。针对 EDA 容易过早收敛的缺点, 将 EDA 和 GA 各自生成一定比例的个体进行混合, 并用模糊逻辑控制的方法对比例进行自适应调节。模糊输入量为种群个体总加工时间的平均值的变化, 模糊输出量为 EDA 在下次迭代中所生成个体的比例。混合算法既利用了 EDA 全局搜索能力强的优势, 又弥补了 EDA 局部搜索能力弱的缺陷, 相比遗传算法也提高了搜索能力。算例仿真的结果证明上述 EDA-GA 混合算法是有效的。

## 致 谢

本工作受国家自然科学基金项目(71301124)、教育部人文社会科学一般项目(13YJC630165), 博士点新教师基金项目(20130141120071)的资助。

## 参考文献 (References)

- [1] Pan, Q.-K., Suganhan, P.N., Tasgetiren, M.F. and Chua, T.J. (2011) A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, **181**, 2455-2468.
- [2] Johnson, S.M. (1954) Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, **1**, 61-68.
- [3] Zhang, Y. and Li, X. (2011) Estimation of distribution algorithm for permutation flow shops with total flow time minimization. *Computers & Industrial Engineering*, **60**, 706-718.
- [4] 周驰, 高亮, 高海兵 (2006) 基于 PSO 的置换流水车间调度算法. *电子学报*, **34**, 2008-2011.
- [5] Larranaga, P. and Lozano, J.A. (2002) Estimation of distribution algorithms: A new tool for evolutionary computation. Springer, Berlin.
- [6] 叶宝林, 高慧敏, 王筱萍, 等 (2011) 基于分布估计算法的二阶段置换流水车间调度算法. *计算机应用研究*, **10**, 3702-3706.
- [7] Chen, S.H., Chen, M.C., Chang, P.C., et al. (2010) Guidelines for developing effective estimation of distribution algorithms in solving single machine scheduling problems. *Expert Systems with Applications*, **37**, 6441-6451.
- [8] Chan, F.T.S., Prakash, A. and Mishra, N. (2013) Priority-based scheduling in flexible system using AIS with FLC approach. *International Journal of Production Research*, **51**, 4880-4895.
- [9] Nawaz, M., Ensco Jr., E.E. and Ham, I. (1983) A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, **11**, 91-95.
- [10] 王圣尧, 王凌, 许焯, 等 (2012) 求解混合流水车间调度问题的分布估计算法. *自动化学报*, **3**, 437-443.
- [11] Wang, S., Wang, L., Liu, M., et al. (2013) An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. *International Journal of Production Economics*, **145**, 387-396.
- [12] 何宏, 钱锋 (2006) 遗传算法参数自适应控制的新方法. *华东理工大学学报(自然科学版)*, **5**, 601-606.
- [13] Kim, K.W., Gen, M. and Yamazaki, G. (2003) Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling. *Applied Soft Computing*, **2**, 174-188.
- [14] Kim, K.W., Yun, Y.S., Yoon, J.M., et al. (2005) Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. *Computers in Industry*, **56**, 143-160.



汉斯出版社为全球科研工作者搭建开放的网络学术中文交流平台。自2011年创办以来，汉斯一直保持着稳健快速发展。随着国内外知名高校学者的陆续加入，汉斯电子期刊已被450多所大中华地区高校图书馆的电子资源采用，并被中国知网全文收录，被学术界广为认同。

汉斯出版社是国内开源（Open Access）电子期刊模式的先行者，其创办的所有期刊全部开放阅读，即读者可以通过互联网免费获取期刊内容，在非商业性使用的前提下，读者不支付任何费用就可引用、复制、传播期刊的部分或全部内容。

