

# 控制与决策

Control and Decision

## 自适应Jaya算法求解多目标柔性车间绿色调度问题

王建华, 潘宇杰, 孙瑞

引用本文:

王建华, 潘宇杰, 孙瑞. 自适应Jaya算法求解多目标柔性车间绿色调度问题[J]. *控制与决策*, 2021, 36(7): 1714–1722.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.1773>

---

### 您可能感兴趣的其他文章

Articles you may be interested in

#### [超启发式交叉熵算法求解模糊分布式流水线绿色调度问题](#)

Hyper-heuristic cross-entropy algorithm for green distributed permutation flow-shop scheduling problem with fuzzy processing time  
*控制与决策*. 2021, 36(6): 1387–1396 <https://doi.org/10.13195/j.kzyjc.2019.1681>

#### [基于正态云模型的状态转移算法求解多目标柔性作业车间调度问题](#)

State transition algorithm based on normal cloud model for solving multi-objective flexible job shop scheduling problem  
*控制与决策*. 2021, 36(5): 1181–1190 <https://doi.org/10.13195/j.kzyjc.2019.1233>

#### [基于分解的多目标多因子进化算法](#)

A multiobjective multifactorial evolutionary algorithm based on decomposition  
*控制与决策*. 2021, 36(3): 637–644 <https://doi.org/10.13195/j.kzyjc.2019.0525>

#### [顺序依赖的调整时间和拖期的无缝钢管热轧批量调度算法](#)

Hot-rolled batch scheduling algorithm for seamless steel tube with sequence-dependent setup times and tardiness  
*控制与决策*. 2021, 36(2): 505–512 <https://doi.org/10.13195/j.kzyjc.2019.0723>

#### [基于改进多目标优化算法的分布式数据中心负载调度](#)

Multi-objective optimization of energy and performance management in distributed data centers  
*控制与决策*. 2021, 36(1): 159–165 <https://doi.org/10.13195/j.kzyjc.2019.0702>

# 自适应 Jaya 算法求解多目标柔性车间绿色调度问题

王建华<sup>†</sup>, 潘宇杰, 孙 瑞

(江苏大学 管理学院, 江苏 镇江 212013)

**摘要:** 针对多目标柔性作业车间绿色调度问题(MO-FJGSP), 建立优化目标为最大完工时间、机器总负荷和能耗最小的多目标数学模型, 并设计一种基于 Pareto 最优解的自适应多目标 Jaya 算法(SAMO-Jaya)对该问题进行优化求解. 算法采用两级实数编码方式实现工序排序与机器分配的编码表示, 并设计一种转换机制实现将 Jaya 连续解空间映射至 FJSP 离散解空间; 然后设计一种混沌序列与均匀分布相结合的混合策略以提高初始种群的质量与全局分散性; 此外, 在 Jaya 算法中嵌入自适应调整种群规模的方法以提高算法求解速度. 通过 10 个单目标与 3 个多目标基准算例测试, 并与 7 个已有算法进行对比分析, 结果表明 SAMO-Jaya 算法能够对 MO-FJGSP 进行有效求解.

**关键词:** 多目标优化; 柔性作业车间绿色调度; 自适应多目标 Jaya 算法; Pareto 最优解; 混合策略

中图分类号: TP301

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.1773

开放科学(资源服务)标识码(OSID):



**引用格式:** 王建华, 潘宇杰, 孙瑞. 自适应 Jaya 算法求解多目标柔性车间绿色调度问题[J]. 控制与决策, 2021, 36(7): 1714-1722.

## Multi-objective flexible job shop green scheduling problem with self-adaptive Jaya algorithm

WANG Jian-hua<sup>†</sup>, PAN Yu-jie, SUN Rui

(College of Management, Jiangsu University, Zhenjiang 212013, China)

**Abstract:** A mathematical model aiming at minimizing the makespan, total machine utilization and energy consumption is established according to the multi-objective flexible job shop green scheduling problem (MO-FJGSP). A self-adaptive multi-objective Jaya algorithm (SAMO-Jaya) based on Pareto optimal solution is designed to optimize the model. And two-level real number encoding is adopted to implement the coding scheme of processes' sequences and machines' assignment, then a transformation mechanism is designed to create a mapping between the continuous solution space of Jaya and the discrete solution space of flexible job-shop scheduling problem (FJSP). And then a hybrid strategy combining chaotic sequence and uniform distribution is raised to improve the quality and diversity of the initial populations. In addition, a self-adaptive population size adjusting method is embedded to improve the optimizing speed of the algorithm. By analyzing the solutions of 10 single-objective benchmarks and 3 multi-objective benchmarks solved by SAMO-Jaya and other 7 existing algorithms, the results show that SAMO-Jaya can solve the MO-FJGSP effectively.

**Keywords:** multi-objective optimization; flexible job shop green scheduling; self-adaptive multi-objective Jaya algorithm; Pareto optimal; mixed strategy

## 0 引言

生产力的快速发展, 导致能源的消耗速度日渐增长. 近些年来, 世界各国已经意识到了能源利用效率不高导致的浪费以及带来的环境污染问题. 绿色制造正是在这个环境下成为学术界与企业界所热议的话题<sup>[1]</sup>. 而绿色车间调度作为绿色制造的关键环节之一, 也引起了许多学者的兴趣与关注<sup>[2]</sup>.

多目标柔性作业车间调度问题 (multi-objective

flexible job-shop scheduling problem, MO-FJSP) 是对传统车间调度问题的扩展, 是较为经典的 NP-hard 问题, 具有多指标、工艺路线柔性、加工机器不确定等特性, 更加符合实际的生产状态. 智能优化算法是求解 MO-FJSP 极为有效的一种方法, 并且已有部分学者运用这些算法取得了研究成果, 例如杂草算法、粒子群算法、萤火虫算法等. 黄霞等<sup>[3]</sup> 建立以最大完工时间、总流经时间和总延迟时间为目标的置换流水

收稿日期: 2019-12-19; 修回日期: 2020-03-14.

基金项目: 国家自然科学基金项目(71673118).

责任编委: 王凌.

<sup>†</sup>通讯作者. E-mail: jiannywang@163.com.

车间调度问题模型,并提出一种改进的混沌杂草算法进行求解.吴锐等<sup>[4]</sup>提出了一种基于三维向量编码方式与局部搜索算子的改进人工蜂群算法用于解决分布式柔性车间调度问题.Dai等<sup>[5]</sup>提出了一种改进遗传算法用于解决带有运输约束<sup>[13]</sup>的多目标柔性作业车间调度问题.Amiri等<sup>[6]</sup>考虑资源分配和作业顺序不确定,建立多目标柔性作业车间模型,并用组合优化策略进行求解.Zhong等<sup>[7]</sup>以最小化最大模糊完成时间和最小化最大模糊机器负荷为调度目标建立多目标模糊柔性作业车间调度问题模型,并设计了一种改进人工蜂群算法对问题进行求解.韩玉艳等<sup>[8]</sup>采用离散NSGA-II算法求解带有限缓冲区的多目标流水线调度问题.朱光宇等<sup>[9]</sup>设计了一种基于直觉模糊集相似度的遗传算法用于解决考虑机器质量的多目标柔性作业车间调度模型.在上述研究中,研究者大多只关注最大完工时间、拖期率等经济指标,少有关注能耗等绿色指标.近些年来,开始有学者研究绿色车间调度问题:艾子义等<sup>[10]</sup>以总碳排放为优化目标,建立相应低碳柔性车间调度问题,并提出一种新型蛙跳算法进行求解;Gong等<sup>[11]</sup>提出了一种混合遗传算法求解具有工人和机器柔性的双柔性作业车间调度问题;吴秀丽等<sup>[12]</sup>考虑机器多转速的因素,通过调节机器转速来设计综合考虑完工时间和能源消耗的低碳调度算法;Zheng等<sup>[13]</sup>提出了一种协同多目标果蝇优化算法用于求解资源约束无关并联机床绿色制造调度问题.

Jaya算法是一种基于群体的新兴智能优化算法<sup>[14]</sup>,该算法区别于其他优化算法,具有无参数运行、求解速度快、不易陷入局部最优解等优点.自Jaya算法被提出以来,已经被广泛应用于诸多领域,并取得了卓越的成果,例如图像识别、机械设计、机器学习等<sup>[15-17]</sup>.该算法在许多领域表现出良好的性能,因此在近些年来被引入车间调度问题进行求解,但由于Jaya算法提出时间较晚,在车间调度方面的应用还较少,且仅有的研究也大多集中在流水车间调度问题领域<sup>[18-19]</sup>,少有应用于多目标柔性作业车间调度问题研究.综上所述,本文以最小化最大完工时间、最小化机器总负荷、最小化能耗为优化目标,构建多目标柔性作业车间绿色调度问题模型(multi-objective flexible job-shop green scheduling problem, MO-FJGSP);并设计了一种基于Pareto最优解的自适应多目标Jaya算法(self-adaptive multi-objective jaya algorithm, SAMO-Jaya).SAMO-Jaya采用两级实数编码方式表示工件信息,针对基本Jaya算法只能解决连

续函数问题的局限性,设计了一种转换机制用于建立Jaya连续空间与FJSP离散空间的映射.为了提高初始种群的质量与全局分布性,设计一种混沌序列和均匀分布相结合的混合策略用于生成初始种群;然后,计算种群的Pareto等级和拥挤度用于选择最优解与最劣解,根据位置更新公式、最优解和最劣解对其余候选解进行更新;最后,设计一种根据解集质量自适应调整种群规模的方法,提高了求解速度,避免算法过早收敛.通过基准算例仿真并与近期具有代表性的多个算法进行性能对比,验证了所提出算法在求解MO-FJGSP上的有效性.

## 1 多目标柔性作业车间绿色调度问题

### 1.1 问题描述

MO-FJGSP描述如下: $n$ 个工件在 $m$ 台机器上加工,工件 $i$ ( $i = 1, 2, \dots, n$ )包含 $q_i$ 道工序,其中不同工件的工序数量可以不等;每道工序可以在多台机器上选择加工,且不同机器的加工时间可以不等, $M_{ij}$ 表示工件 $i$ 的第 $j$ 道工序的可选机器集, $M_{ij} \in \{1, 2, \dots, m\}$ ;  $S_{ij}$ 为工件 $i$ 的第 $j$ 道工序的开始加工时间;  $T_{ij}^k$ 为工件 $i$ 的第 $j$ 道工序在机器 $k$ 上的加工时间;  $F_{ij}$ 为工件 $i$ 的第 $j$ 道工序的结束加工时间;优化目标包括最大完工时间 $C_T$ 、机器总负荷 $T_M$ 和能耗 $E$ .

在对问题建立模型之前,需要依据实际情况作一些合理的假设:

- 1) 每台机器在同一时刻只能加工一个工件;
- 2) 机器在运行过程中不会出现故障;
- 3) 工件一旦进入机器加工就不能被打断;
- 4) 每道工序仅能被一台机器加工一次;
- 5) 在0时刻,所有工件材料已经准备完毕;
- 6) 同一工件的不同工序存在紧前紧后关系.

为了方便读者对本文的理解,本文出现的符号定义如下: $n$ 为工件数量; $m$ 为机器数量; $i, h$ 为工件号; $j, v$ 为工序号; $k$ 为机器号; $q_i$ 为工件 $i$ 的工序数; $O_{ij}$ 为工件 $i$ 的第 $j$ 道工序; $M_{ij}$ 为工序 $O_{ij}$ 的可选机器集; $X_{ij}^k$ 为工序 $O_{ij}$ 是否在机器 $k$ 上加工,为0-1变量; $Y_{hvi j}$ 为工序 $O_{hv}$ 与工序 $O_{ij}$ 的加工顺序,为0-1变量; $S_{ij}$ 为工序 $O_{ij}$ 的开始加工时间; $T_{ij}^k$ 为工序 $O_{ij}$ 在机器 $k$ 上的加工时间; $F_{ij}$ 为工序 $O_{ij}$ 的结束加工时间; $a_{ij}^k$ 为工序 $O_{ij}$ 在机器 $k$ 上的负载单位能耗; $b^k$ 为机器 $k$ 的空载单位能耗; $M$ 为一个大的正实数; $C_T$ 为最终完工时间; $T_M$ 为机器总负荷; $E$ 为能耗.

### 1.2 模型建立

在实际生产中,管理者往往要依据多个目标作出决策,本文的调度模型权衡了经济指标和绿色

指标两个方面,考虑了最大完工时间( $C_T$ )、机器总负荷( $T_M$ )和能耗( $E$ )3个优化目标,可以表示为 $\min(C_T, T_M, E)$ ,其中最大完工时间为经济指标.采用机器总负荷这一指标是为了衡量设备的使用寿命,而能耗则是作为绿色指标,具体的计算公式如下:

$$\min C_T = \max(F_{ij}) = \max(S_{ij} + T_{ij}^k); \quad (1)$$

$$\min T_M = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^m T_{ij}^k; \quad (2)$$

$$\min E = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^m [a_{ij}^k X_{ij}^k T_{ij}^k + b^k (\max(X_{ij}^k F_{ij}) - \min(X_{ij}^k S_{ij}) - X_{ij}^k T_{ij}^k)]. \quad (3)$$

$$\text{s.t. } X_{hv}^k F_{hv} \leq X_{ij}^k S_{ij} + M(1 - Y_{hvij}), \forall i, j, k; \quad (4)$$

$$F_{ij} - S_{ij} = T_{ij}^k, \forall i, j, k; \quad (5)$$

$$\sum_{k=1}^m X_{ij}^k = 1, \forall i, j; \quad (6)$$

$$S_{i(j+1)} \geq F_{ij}, \forall i, j; \quad (7)$$

$$X_{ij}^k \in \{0, 1\}, \forall i, j, k; \quad (8)$$

$$Y_{hvij} \in \{0, 1\}, \forall h, v, i, j. \quad (9)$$

其中:式(1)~(3)为目标函数,分别表示最小化最大完工时间、最小化机器总负荷、最小化能耗;式(4)表示每台机器在同一时刻只能加工一个工件;式(5)表示工件一旦开始加工中途不能被打断;式(6)表示每道工序仅能被一台机器加工一次;式(7)表示同一工件的工序之间存在紧前紧后关系;式(8)为0-1变量,若工序 $O_{ij}$ 在机器 $k$ 上加工,则 $X_{ij}^k = 1$ ,否则 $X_{ij}^k = 0$ ;式(9)为0-1变量,若工序 $O_{hv}$ 在工序 $O_{ij}$ 之前加工,则 $Y_{hvij} = 1$ ,否则 $Y_{hvij} = 0$ .

## 2 自适应多目标Jaya算法

基本Jaya算法仅应用于求解连续函数问题,而MO-FJGSP为离散函数问题,为求解该问题,需要解决连续函数与离散函数之间的转换,因此本文设计了一种转换机制用于连续与离散问题的相互转换.为了提升初始种群的质量,设计了一种混沌序列与均匀分布相结合的混合策略.Jaya算法的核心在于位置更新策略,该策略需要通过获取每一代种群的最优解与最劣解实现,而MO-FJGSP为多目标问题,不易直接得到两个极端解,因此本文通过计算所有个体非支配等级来进行分类,再通过计算拥挤度从第1类与最后1类中选择出最优解与最劣解.此外,在算法中嵌入了一种自适应调整种群规模的方法用于加快算法的求解速度.区别于其他基于群体的智能优化算法,SAMO-Jaya不需要特定的参数控制种群的迭代,

例如遗传算法的变异率、交叉率等,而合适的参数对于算法的寻优能力有着很大的影响,因此运用SAMO-Jaya对问题进行求解时不容易陷入局部最优解,且寻优能力更强.

### 2.1 编解码与转换机制

编码方式的合适与否直接决定算法运行的性能,对最终的求解结果有着相当重要的影响.MO-FJGSP可以看成两个子问题,即工序排序和机器分配.针对该特点,本文的编码方式采用两级实数编码,第1级采用基于工序的编码方式,第2级采用基于机器的编码方式,分别对应工序排序和机器分配.为方便理解,现有一个简单 $3 \times 4$ 的MO-FJGSP例子,工序集为 $O_1 = \{O_{11}, O_{12}, O_{13}\}$ , $O_2 = \{O_{21}, O_{22}\}$ , $O_3 = \{O_{31}, O_{32}, O_{33}\}$ ,机器集为 $m = \{m_1, m_2, m_3, m_4\}$ .图1为一个可行解的编码,第1行为工序编号,第2行为机器分配,该编码表示工件1的第1道工序在机器1上加工,工件2的第1道工序在机器3上加工,以此类推.其中第1行的数字表示工件号,相同的工件号表示同一工件的前后工序,第2行的数字表示机器号,这种两级实数编码方式既能满足工艺路线的约束,又能满足加工机器柔性的约束.

工序编号	1	2	3	1	3	1	3	2
机器分配	1	3	2	3	2	1	4	4

图1 编码示例

由图1可以看出,MO-FJGSP中的解都为离散的实数,而Jaya算法的个体位置更新都为连续值,因此为了实现算法解集与调度解空间的相互映射,本文在这里针对工序编号与机器分配设计了一种转换机制.

#### 1) 工序编号的转换.

将个体位置的连续值范围限定为 $[\alpha, \beta]$ ,依据下式:

$$Z(i) = \frac{(\beta - \alpha)i}{n} + \alpha \quad (10)$$

可以将工件编码转化为相应的位置向量,其中 $Z(i)$ 为工件 $i$ 对应的位置向量, $n$ 为工件数.图1中的工件编码一行可以转换为图2所示的位置元素,范围为 $[1, 2]$ .

工序编号	1	2	3	1	3	1	3	2
位置向量	1.33	1.67	2	1.33	2	1.33	2	1.67

图2 工序编号转换

#### 2) 机器编号的转换.

采用下式:

$$Z(ij) = \frac{(\beta - \alpha)\text{pos}(m_{ij})}{M_{ij}} + \alpha, \quad (11)$$

可以将可选机器集中的机器位置转换为个体位置向量,连续值仍然限定为 $[\alpha, \beta]$ . 其中: $Z(ij)$ 为工序 $O_{ij}$ 所选机器 $m_{ij}$ 的位置向量, $\text{pos}(m_{ij})$ 表示所选机器在工序可选机器集中的位置, $M_{ij}$ 为工序 $O_{ij}$ 的可选机器数量. 假设上例中的可选机器集为 $M_1 = \{3, 2, 4\}$ ,  $M_2 = \{1, 3\}$ ,  $M_3 = \{4, 2, 4\}$ ,  $M_1$ 表示工件1的3道工序可选机器集数量为3, 2, 4, 以此类推. 图1中的机器分配行可以转换为图3所示的位置元素.

机器编号	1	3	2	3	2	1	4	4
可选机器集	3	1	4	2	2	4	4	3
位置号	1	1	2	2	1	1	4	3
位置向量	1.33	2	1.5	2	1.5	1.25	2	2

图3 机器编号转换

3) 个体位置向调度方案的逆向转换.

$$i \approx \frac{[Z(i) - \alpha]n}{\beta - \alpha}; \quad (12)$$

$$\text{pos}(m_{ij}) \approx \begin{cases} \frac{[Z(ij) - \alpha]M_{ij}}{\beta - \alpha}, & \text{pos}(m_{ij}) \neq 0; \\ 1, & \text{pos}(m_{ij}) = 0. \end{cases} \quad (13)$$

个体位置向调度方案的转换,即将位置向量的连续值转换为调度方案的离散实数. 可以通过式(10)和(11)的逆运算式(12)和(13)实现,位置向量转为工序编号可依据式(12),位置向量转为机器编号可依据式(13),然后根据求得的 $\text{pos}(m_{ij})$ 在可选机器集中确定机器编号,结果均取整数.

需要注意的是,在用式(13)进行运算时, $\text{pos}(m_{ij})$ 有可能会出现0,因此当 $\text{pos}(m_{ij})$ 为0时,将 $\text{pos}(m_{ij})$ 默认为1,即选取 $M_{ij}$ 中的第1个机器.

### 2.2 基于混合策略的种群初始化

初始种群的质量将对最终求得的结果产生很大的影响,为了保证初始种群具有较高的质量,同时保存种群的多样性,本文设计一种混沌序列与均匀分布相结合的混合策略产生初始种群. 混沌序列是一种确定系统中出现的无规则的运动,但是值得注意的是混沌不是混乱无序,而是有着精致内在结构且在一定范围内按其自身的“规律”不重复地遍历所有状态,混沌行为的起源在于非线性系统对初值的敏感依赖性<sup>[20]</sup>. 混沌序列的核心logistic方程为

$$x_{n+1} = vx_n(1 - x_n), \quad 2 < v < 4, \quad (14)$$

其中 $x_n$ 为一数列,满足式(14)的递推关系.

该操作主要分为两个阶段,即工序排序和机器分配. 在工序排序阶段,采用随机打乱的方法对所有工序的先后顺序进行排序;在机器分配阶段,每一道工序的机器都采用混沌序列的方法从可选机器集中产生,在此过程中实时统计所有机器的被使用次数,若某台机器的被使用次数到达平均值,则该工序优先使用其他机器,保证所有机器的负载均衡. 不断重复上述步骤,直至生成完整的初始种群. 生成初始种群的具体步骤如下.

step 1: 定义调度方案的编码长度为 $\sum_{i=1}^n q_i$ ,按顺序分别产生 $q_i$ 个 $i$ ,其中 $i = 1, 2, \dots, n$ ,见图4.

工序编号	1	1	1	2	2	3	3	3
rand值	0.74	0.52	0.81	0.13	0.63	0.46	0.91	0.32
工序编号	2	3	3	1	2	1	1	3

图4 工序排序

step 2: 对每个工序随机产生一个rand值,其中 $\text{rand} \in [0, 1]$ ,根据rand值的大小对所有工序进行升序排序,可得到一组工序的加工序列,见图4.

step 3: 机器分配. 随机产生一组随机值 $Z(ij)$ ,其中 $Z(ij) \in [\alpha, \beta]$ ,根据式(14)的logistic方程可将 $Z(ij)$ 转为具有混沌序列特性的 $Z(ij)$ ,再根据式(13)在可选机器集中进行选择.

step 4: 均匀分配. 统计所有机器的被使用次数,若某一机器已达到被使用次数的平均值,则该工序优先使用其他机器.

### 2.3 计算个体非支配等级和拥挤度

对MO-FJGSP而言,不能通过单一的目标来评价解的优良性,为了选择出种群中的最优解与最劣解用于SAMO-Jaya的位置更新,需要一种好的适应度评价策略来保证种群将表现优异的解遗传至下一代. 现有的研究中对于多目标问题的求解大多采用Pareto占优的方法,求得非劣解集. 其中Pareto占优可以理解为对于 $n$ 个目标分量 $f_i(x)$ , $i = 1, 2, \dots, n$ ,任意给定两个决策变量 $x_1$ 、 $x_2$ ,若满足如下两个条件: 1)  $\forall i \in 1, 2, \dots, n$ ,都有 $f_i(x_1) \leq f_i(x_2)$ 成立; 2)  $\exists i \in 1, 2, \dots, n$ ,使得 $f_i(x_1) < f_i(x_2)$ 成立. 则称 $x_1$ 支配 $x_2$ . 因此,本文首先采用Pareto占优计算个体的非支配等级,在计算完个体等级之后,还需考虑同一等级的Pareto解的优先级关系. 相关文献表明,对于相同等级的多个解,需要采用一定的策略来区分

解之间的优先级关系,目前较为常用的策略有聚类分析、信息熵和拥挤度等<sup>[21]</sup>.对于SAMO-Jaya算法而言,对个体进行排序的主要目的是为了求得种群中最坏与最好的两个解用于其余个体位置的更新,因此个体之间的拥挤度更加适用于本文算法.拥挤度 $n_d$ 的计算公式如下:

$$n_d = \sum_{m=1}^M \frac{f_m(i+1) - f_m(i-1)}{f_m^{\max} - f_m^{\min}}. \quad (15)$$

其中: $M$ 为目标函数的个数, $f_m^{\max}$ 为个体目标函数值 $f_m$ 的最大值, $f_m^{\min}$ 为个体目标函数值 $f_m$ 的最小值, $f_m(i-1)$ 、 $f_m(i+1)$ 分别为该个体排序后的前后两位的目标函数值.

#### 2.4 个体位置更新策略

在SAMO-Jaya算法的搜索过程中,候选个体 $A_k(i)$ 由当前种群中最优解 $A_b(i)$ 和最劣解 $A_w(i)$ 引导进行个体位置的更新,其中最优解与最劣解通过2.3节计算个体非支配等级和拥挤度来获得,整个引导的过程是将候选个体引向最优解个体.由于在更新过程中存在候选个体偏离最优解个体的情况,在更新完个体后,需要将新个体与旧个体的目标函数值进行比较,选取表现优异的个体放入下一代.个体位置更新公式如下:

$$A_k(i+1) = A_k(i) + r_1(A_b(i) - |A_k(i)|) - r_2(A_w(i) - |A_k(i)|), \quad (16)$$

其中: $r_1$ 、 $r_2$ 为 $[0,1]$ 范围内随机产生的数字.

#### 2.5 自适应调整种群规模

MO-FJGSP作为一个实际生产中的问题,所涉及的工件和机器数量非常庞大,因此种群的规模也十分巨大,运用算法进行求解的速度相对较慢.而在企业界,非常看重算法的求解速度,有时为了提高求解速度而放弃那些求解结果好但速度慢的算法.因此,为了提高SAMO-Jaya算法的求解速度,本文嵌入了一种自适应调整种群规模的方法,在算法时间复杂度不变的情况下,通过有选择地缩小种群规模来达到提高求解速度的目的.

相关文献表明,自Jaya算法被提出以来,尚未有相关研究将其与自适应调整种群规模结合应用.自适应调整种群规模的公式如下所示:

$$n_{\text{new}} \approx n_{\text{old}} + r n_{\text{old}}. \quad (17)$$

其中: $n_{\text{new}}$ 为下一代的种群规模,结果取整数; $n_{\text{old}}$ 为当前的种群规模; $r$ 为 $[-0.3, 0.3]$ 之间的随机值,将作为控制种群数量的参数.在当代的种群质量好于上一代时, $r \in (0, 0.3]$ ,种群规模将扩大,当代的所有

个体都进入下一代,并且按照2.3节所计算的Pareto等级和拥挤度对所有个体进行排名,将当代种群前 $n_{\text{new}} - n_{\text{old}}$ 个体赋予剩余个体;在当代的种群质量劣于上一代时, $r \in [-0.3, 0)$ ,种群规模将缩小,实行精英主义,同样按照2.3节计算所得种群个体排序,选取前 $n_{\text{old}} - n_{\text{new}}$ 个体进入下一代;在当代的种群质量与上一代相同时, $r = 0$ ,种群规模不变.

本文对于评价种群质量的方法如下:多目标问题中的单个目标之间不能直接进行对比,不同代种群的所有个体目标值也无法直接进行对比,因此文本评价种群质量的指标为2.3节计算所得的非劣解的数量,非劣解的数量是评价一个算法寻优性能的体现,非劣解数量越多,种群的质量越好,当非劣解数量相同时,视为两代种群质量相同.

#### 2.6 算法流程

SAMO-Jaya算法的具体实现步骤如下.

step 1: 种群初始化.根据2.2节的方法进行种群初始化,并设置参数迭代次数 $n = 0$ 和终止条件.

step 2: 寻找最优和最劣解.根据2.3节的方法计算所有个体的非支配等级和拥挤度,并进行排序选出其中的最优解和最劣解.

step 3: 更新种群.根据2.4节的个体位置更新公式,对所有候选解进行更新.

step 4: 判断新旧解.新解是否好于旧解,若是,则将新解放入下一代;否则将旧解放入下一代.

step 5: 判断终止条件.终止条件是否满足,若是,则输出最优解方案;否则执行step 6.

step 6: 自适应调整种群规模.根据2.5节的调整种群规模的方法对种群大小进行调整,返回step 2,  $n = n + 1$ .

### 3 仿真测试与分析

本文采用Matlab2014a进行编程,并在Windows7, Intel(R) Core(TM) i5-3337U, CPU1.8 GHz, 内存4GB, 64位操作系统的计算机上运行.算法的参数设置为:种群大小为100,迭代次数为200代.

#### 3.1 算法性能测试

为了验证SAMO-Jaya算法的可行性和有效性,本文从单目标和多目标两个方面对本文算法进行测试.在单目标方面,本文将其应用于Benchmark算例中的MK01-MK10基准算例,并用SAMO-Jaya算法分别运行20次,取其中的最优解与具有相同优化理念的改进算法:Caldeira等<sup>[22]</sup>提出的改进Jaya算法(improver jaya algorithm, IJaya)、Prasert等<sup>[23]</sup>提出的改进微分进化算法(improved differential evolution, IDE)

和姜天华<sup>[24-25]</sup>提出的灰狼优化算法 (grey wolf optimization, GWO) 以及混合灰狼优化算法 (hybrid grey wolf optimization, HGWO) 进行对比. 表1为各算法结果比较, 粗体表示相同算例中取得的最优解.

表1 Benchmark算例对比

算例	$n \times m$	IJaya	IDE	GWO	HGWO	SAMO-Jaya
MK01	10×6	40	40	42	40	<b>36</b>
MK02	10×6	<b>28</b>	<b>28</b>	31	29	<b>28</b>
MK03	15×8	<b>204</b>	<b>204</b>	<b>204</b>	<b>204</b>	<b>204</b>
MK04	15×8	60	71	68	65	<b>59</b>
MK05	15×4	172	179	179	175	<b>171</b>
MK06	10×15	<b>57</b>	73	68	79	75
MK07	20×5	<b>139</b>	146	155	149	149
MK08	20×10	<b>523</b>	528	<b>523</b>	<b>523</b>	<b>523</b>
MK09	20×10	—	<b>321</b>	—	325	325
MK10	20×15	—	<b>235</b>	—	253	272

由表1可以发现, 在Benchmark的10个算例中, 本文算法在算例MK01~MK05和算例MK08下都达到最优或优于其他4种算法, 能获得最优解, 只在MK06/07/09/10四个算例下未取得最优解, 但除了MK10的结果不理想外, 其余3个算例的结果都是接近于最优解. 此外, 数据显示SAMO-Jaya算法能获得最优解的个数为6个, 为5种算法中表现最好的算法. 图5为MK01算例下Makespan = 36的调度方案. 其次, 与现有的IJaya算法相比, SAMO-Jaya在MK01~MK08八个算例中, 有6个算例表现最好, 略优于IJaya的5个.

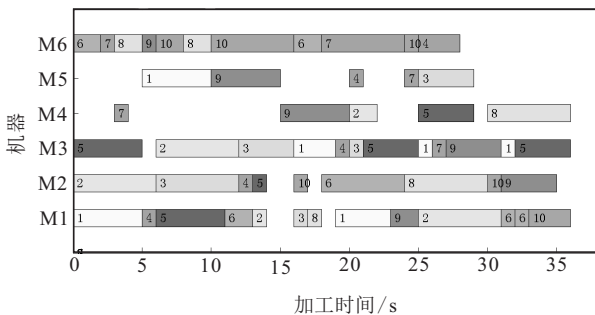


图5 MK01 (Makespan=36)调度方案

为了探究各改进部分对算法的影响, 以MK04算例为例, 在此基础上分析改进部分对Jaya算法性能的影响. 图6为引入混合策略前后Jaya算法在求解MK04算例时的收敛曲线, 可以看出SAMO-Jaya算法在早期的收敛速度和收敛效果要明显优于Jaya算法, 且SAMO-Jaya算法在125代左右时已达到最优, 最优解为59, 而Jaya算法直到150代左右才求得最优解65, 由此可以看出本文设计的混合策略对提高种群质量的有效性. 为了验证在迭代过程嵌入自适应调

整种群规模的方法之后对提高求解速度的有效性, 将SAMO-Jaya算法与Jaya算法同时求解MK04算例20次, 得到如表2所示的结果, 其中RT为系统运行时间, VAR为20次结果的方差, 可以看出SAMO-Jaya的求解速度明显快于Jaya, 但在稳定性上二者并没有差别. 因此, 在算法中嵌入自适应调整种群规模的方法后能有效提升求解速度, 但不会影响求解结果的稳定性.

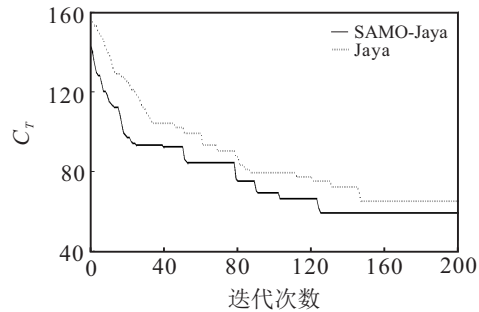


图6 引入混合策略前后Jaya算法MK04收敛曲线

表2 Jaya算法与SAMO-Jaya算法求解速度对比

指标	RT / min		VAR	
	SAMO-Jaya	Jaya	SAMO-Jaya	Jaya
结果	1.12	1.48	0.107	0.132

为了进一步验证本文算法的性能, 用SAMO-Jaya算法求解经典算例Kacem中的8×8, 10×10, 15×10三个基准算例, 并与Alzahrani<sup>[26]</sup>提出的抢占式约束规则 (pre-emptive constraint procedure, PCP)、曹磊等<sup>[27]</sup>提出的变邻域杂草优化算法 (variable neighbourhood invasive weed optimization, VNIWO)、Soto等<sup>[28]</sup>提出的多目标分支界定算法 (multi-objective branch and bound, MBB) 进行对比, 实验结果如表3所示. 其中C<sub>m</sub>为最大完工时间, W<sub>m</sub>为瓶颈机器负载, W<sub>t</sub>为机器总负载.

由表3的实验结果可以看出, SAMO-Jaya求解8×8问题所得的非劣解为4个, 仅次于VNIWO, 多于PCP和MBB, 将SAMO-Jaya与VNIWO的非劣解集放在一起进行Pareto占优操作后可以发现, VNIWO还剩[16, 13, 73]一组解, 而SAMO-Jaya还剩[15, 11, 77], [16, 12, 75], [15, 12, 76]三组解, 因此SAMO-Jaya从解集质量来看要优于VNIWO; 求解10×10问题所得非支配解为4个, 多于其他余3个算法, 且SAMO-Jaya的性能指标更占优势; 求解15×10问题所得非支配解为2个, 少于PCP而多于MBB. 因此, SAMO-Jaya能够有效求解不同规模的多目标问题, 且能保证解的质量.

表3 Kacem基准算例结果对比

算例	目标	PCP			VNIWO				MBB			SAMO-Jaya				
8×8	$C_m$	16	15	16	15	16	17	17	18	15	17	14	15	16	15	17
	$W_m$	12	13	11	13	13	11	12	12	13	11	10	11	12	12	11
	$W_t$	78	76	82	75	73	78	76	75	73	75	78	77	75	76	75
10×10	$C_m$	8	9		9	8	7			8	7		7	8	7	7
	$W_m$	7	5		7	5	6			6	7		6	6	5	6
	$W_t$	44	43		41	43	42			41	43		42	41	43	42
15×10	$C_m$	13	11	11		11	12				12			12	11	
	$W_m$	10	11	10		11	10				11			10	11	
	$W_t$	93	95	98		91	93				91			91	93	

3.2 MO-FJGSP求解与分析

求解本文问题模型所采用的算例为MK08,目标函数为最大完工时间( $C_T$ )、机器总负荷( $T_M$ )和能耗( $E$ ),其中 $C_T$ 和 $T_M$ 均可由算例中的数据直接求得, $E$ 的值无法直接求得,因此添加各工件在机器上加工能耗以及各机器空载能耗的数据集,由计算机在[0.5, 2]和[0.1, 0.3]范围内随机产生,14台机器的单位加工能耗为[1.3, 0.9, 1.7, 1.1, 1.2, 1.9, 0.7, 0.5, 2.0, 1.5, 1.8, 0.9, 1.1, 1.3], 14台机器的单位空载能耗为[0.2, 0.1, 0.1, 0.3, 0.2, 0.3, 0.2, 0.2, 0.2, 0.1, 0.2, 0.3, 0.1, 0.2]. 算法最终求得如表3所示具有18个非支配解的Pareto解集. 表4所示的18个解为本文问题模型提供了18种相对较优的调度方案. 在实际生产中,生产

计划员需要根据自身企业的需求,通过层次分析法(analytic hierarchy process, AHP)为不同的目标确定相应的权重,从中选出最适用的调度方案. 假设 $C_T$ 、 $T_M$ 、 $E$ 的权重分别为0.5、0.3、0.2,可以计算出上述18

表4 最终代Pareto最优解集

序号	$C_T$	$T_M$	$E$	序号	$C_T$	$T_M$	$E$
1	536	2916	621.8	10	530	2815	697.5
2	547	2877	631.8	11	534	2567	647.3
3	527	2741	733.4	12	540	2713	535.3
4	543	2566	737.1	13	523	2884	712.5
5	533	2683	724.7	14	535	2654	637.9
6	550	2661	617.6	15	548	2742	621.2
7	530	2783	707.1	16	536	2739	636.5
8	534	2572	640.2	17	543	2563	740.2
9	539	2645	637.4	18	534	2765	718.6

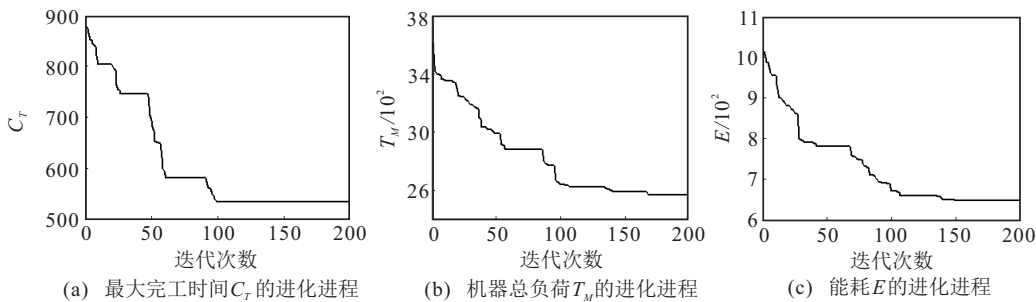


图7 各目标函数最优值的进化过程

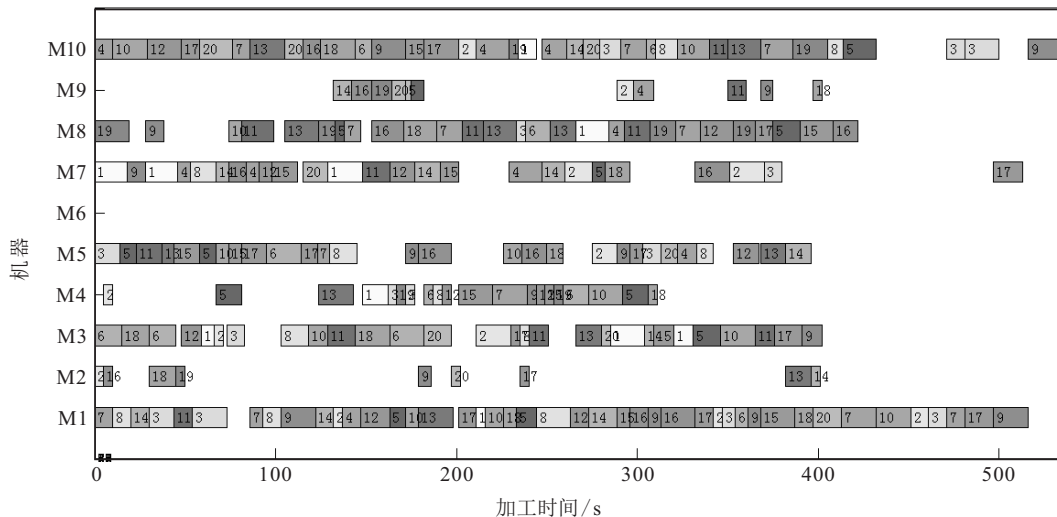


图8 各目标函数最优值的进化过程

个非支配解的对应适应度值,通过排序最终得到第11组为最优解.最大完工时间为534,机器总负荷为2567,能耗为647.3,在迭代过程中各目标值的收敛曲线如图7所示,第11组解的甘特图如图8所示.

## 4 结 论

针对能源消耗过大的现状,建立了以最大完工时间、机器总负荷、能耗为目标的多目标柔性作业车间绿色调度问题模型,并根据问题特点,对基本Jaya算法进行了一系列改进:

1) 设计了一种转换机制用于将问题从离散分布转换为连续分布,便于Jaya算法的求解;采用混沌序列与均匀分布相结合的混合策略产生初始种群,提高了初始种群的质量,同时降低了陷入局部最优解的概率;将自适应调整种群规模方法与Jaya算法相结合,加快了算法的求解速度.

2) 从单目标和多目标两个方面求解了Kacem和Benchmark的13个基准算例,对算法性能进行了验证,证明了算法的有效性和稳定性,同时求解了MO-FJGSP并对其进行了分析.

笔者将在本文研究基础上,进一步探究Jaya算法应用于动态重调度等更为复杂的问题.

## 参考文献(References)

- [1] “中国工程院绿色制造发展战略研究”课题组. 推进绿色制造建设生态文明——中国绿色制造战略研究[J]. 中国工程科学, 2017, 19(3): 53-60.  
(The research group for research on green manufacturing strategy. Advancing green manufacturing for ecological civilization: Research on green manufacturing strategies in china[J]. Chinese Engineering Science, 2017, 19(3): 53-60.)
- [2] 王凌, 王晶晶, 吴楚格. 绿色车间调度优化研究进展[J]. 控制与决策, 2018, 33(3): 385-391.  
(Wang L, Wang J J, Wu C G. Advances in green shop scheduling and optimization[J]. Control and Decision, 2018, 33(3): 385-391.)
- [3] 黄霞, 叶春明, 曹磊. 多目标置换流水车间调度的混沌杂草优化算法[J]. 系统工程理论与实践, 2017, 37(1): 253-262.  
(Huang X, Ye C M, Cao L. Chaos invasive weed optimization algorithm for multi-objective permutation flow shop scheduling problem[J]. Systems Engineering—Theory & Practice, 2017, 37(1): 253-262.)
- [4] 吴锐, 郭顺生, 李益兵, 等. 改进人工蜂群算法求解分布式柔性作业车间调度问题[J]. 控制与决策, 2019, 34(12): 2527-2536.  
(Wu R, Guo S S, Li Y B, et al. Improved artificial bee colony algorithm for distributed and flexible job shop scheduling problem[J]. Control and Decision, 2019, 34(12): 2527-2536.)
- [5] Dai M, Tang D B, Giret A S, et al. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints[J]. Robotics and Computer-Integrated Manufacturing, 2019, 59: 143-157.
- [6] Amiri F, Babak S, Ali T. Multi-objective simulation optimization for uncertain resource assignment and job sequence in automated flexible job shop[J]. Applied Soft Computing Journal, 2019, 75: 190-202.
- [7] Zhong Y G, Yang F, Liu F. Solving multi-objective fuzzy flexible job shop scheduling problem using MABC algorithm[J]. Journal of Intelligent & Fuzzy Systems, 2019, 36(2): 1455-1473.
- [8] 韩玉艳, 李俊青, 桑红燕, 等. 离散NSGA-II求解带有限缓冲区的多目标批量流水线调度问题[J]. 聊城大学学报: 自然科学版, 2018, 31(1): 89-96.  
(Han Y Y, Li J Q, Sang H Y, et al. Discrete NSGA-II for multi-objective lot-streaming flow shop scheduling problem with limited buffers[J]. Journal of Liaocheng University: Natural Science, 2018, 31(1): 89-96.)
- [9] 朱光宇, 徐文婕. 考虑能耗与质量的机床构件生产线多目标柔性作业车间调度方法[J]. 控制与决策, 2019, 34(2): 252-260.  
(Zhu G Y, Xu W J. Multi-objective flexible job shop scheduling method for machine tool component production line considering energy consumption and quality[J]. Control and Decision, 2019, 34(2): 252-260.)
- [10] 艾子义, 雷德明. 基于新型蛙跳算法的低碳柔性作业车间调度[J]. 控制理论与应用, 2017, 34(10):104-111.  
(Ai Z Y, Lei D M. A novel shuffled frog leaping algorithm for low carbon flexible job shop scheduling[J]. Control Theory & Applications, 2017, 34(10): 104-111.)
- [11] Gong G L, Deng Q W, Gong X R, et al. A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators[J]. Journal of Cleaner Production, 2018, 174: 560-576.
- [12] 吴秀丽, 孙阳君. 机器多转速的柔性作业车间绿色调度问题[J]. 计算机集成制造系统, 2018, 24(4): 54-67.  
(Wu X L, Sun Y J. Flexible job shop green scheduling problem with multi-speed machine[J]. Computer Integrated Manufacturing Systems, 2018, 24(4): 54-67.)
- [13] Zheng X L, Wang L. A collaborative multi-objective fruit fly optimization algorithm for the resource constrained unrelated parallel machine green scheduling problem[J].

- IEEE Transactions on Systems, Man & Cybernetics, 2018, 48(5): 790-800.
- [14] Rao R V, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems[J]. International Journal of Industrial Engineering Computations, 2016(7): 19-34.
- [15] Zhang Y D, Yang X J, Cattani C, et al. Tea category identification using a novel fractional Fourier entropy and Jaya algorithm[J]. Entropy, 2016, 18(3): 77-93.
- [16] Rao R V, Waghmare G G. A new optimization algorithm for solving complex constrained design optimization problems[J]. Engineering Optimization, 2017, 49(1): 60-83.
- [17] Sinha R K, Ghosh S. Jaya based ANFIS for monitoring of two class motor imagery task[J]. IEEE Access, 2016(4): 9273-9282.
- [18] Aseem M, Divya S. A TLBO and a Jaya heuristics for permutation flow shop scheduling to minimize the sum of inventory holding and batch delay costs[J]. Computers & Industrial Engineering, 2018, 124: 509-522.
- [19] Buddala R, Mahapatra S S. Improved teaching-learning-based and JAYA optimization algorithms for solving flexible flow shop scheduling problems[J]. Journal of Industrial Engineering International, 2018, 14: 555-570.
- [20] 王宁, 蔚承建, 盛昭瀚. 基于嵌入混沌序列的遗传算法[J]. 系统工程理论与实践, 1999, 19(11): 1-8.  
(Wang N, Wei C J, Sheng Z H. Genetic algorithm combined with a chaotic sequence[J]. Systems Engineering—Theory & Practice, 1999, 19(11): 1-8.)
- [21] 肖海宁, 楼佩煌, 严伟国, 等. 柔性作业车间中机床与自动导引车在线调度方法[J]. 农业机械学报, 2013, 44(4): 280-286.  
(Xiao H N, Lou P H, Yan W G, et al. On-line scheduling method for simultaneous scheduling of machines and automated guided vehicles in flexible job shop[J]. Transactions of the Chinese Society for Agricultural Machinery, 2013, 44(4): 280-286.)
- [22] Caldeira R H, Gnanavelbabu A. Solving the flexible job shop scheduling problem using an improved Jaya algorithm[J]. Computers and Industrial Engineering, 2019, 137: 106064.
- [23] Prasert S, Nuchara K, Preecha K. Improved differential evolution algorithm for flexible job shop scheduling problems[J]. Mathematical and Computational Applications, 2019, 24(3): 80-98.
- [24] 姜天华. 基于灰狼优化算法的低碳车间调度问题[J]. 计算机集成制造系统, 2018, 24(10): 56-63.  
(Jiang T H. Low-carbon workshop scheduling problem based on grey wolf optimization[J]. Computer Integrated Manufacturing Systems, 2018, 24(10): 56-63.)
- [25] 姜天华. 混合灰狼优化算法求解柔性作业车间调度问题[J]. 控制与决策, 2018, 33(3): 503-508.  
(Jiang T H. Flexible job shop scheduling problem with hybrid grey wolf optimization algorithm[J]. Control and Decision, 2018, 33(3): 503-508.)
- [26] Alzahrani J S. Multi-objective job shop scheduling using pre-emptive constraint procedure[J]. American Journal of Modeling and Optimization, 2019, 7(1): 8-13.
- [27] 曹磊, 叶春明, 黄霞. 基于员工学习行为的多目标柔性车间调度[J]. 计算机集成制造系统, 2018, 24(8): 2023-2034.  
(Cao L, Ye C M, Huang X. Multi-objective flexible job-shop scheduling based on learning effect[J]. Computer Integrated Manufacturing Systems, 2018, 24(8): 2023-2034.)
- [28] Soto C, Dorronsoro B, Fraire H. Solving the multi-objective flexible job shop scheduling problem with a novel parallel branch and bound algorithm[J]. Swarm and Evolutionary Computation, 2020, 53: 100632.

### 作者简介

王建华(1977—), 男, 副教授, 博士, 从事工业工程、智能调度等研究, E-mail: jiannywang@163.com;

潘宇杰(1995—), 男, 硕士生, 从事算法优化、智能调度的研究, E-mail: wzupanyujie@163.com;

孙瑞(1995—), 男, 硕士生, 从事运筹学与智能算法的研究, E-mail: 772615743@qq.com.

(责任编辑: 孙艺红)