

6-2-2020

Improved Shuffled Frog-Leaping Algorithm for Solving Flexible Job Shop Scheduling Problem

Xiaoxing Zhang

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Wang Yan

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Dahu Yan

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Zhicheng Ji

Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Improved Shuffled Frog-Leaping Algorithm for Solving Flexible Job Shop Scheduling Problem

Abstract

Abstract: Aiming at the characteristics of flexible job shop scheduling problem, a *multi-objective scheduling model with maximum completion time and minimum energy consumption* was proposed. An improved shuffled frog leaping algorithm was developed. By designing the local updating strategy based on crossover operation of maximum preserved crossover (MPX) and shifting operation of single parent gene algorithm (PGA), it avoided the illegal solution and trimming of the algorithm. Additionally, it accelerated optimization rate of the algorithm. And the optimal solution of the group was optimized by the simplified neighborhood optimization strategy to prevent the algorithm from falling into the local optimum. Then, by solving an example of an enterprise production workshop, scheduling schemes under different weights were obtained. Compared with completion time and processing energy consumption obtained by the classical shuffled frog-leaping algorithm, it proves the effectiveness of the algorithm.

Keywords

flexible job shop scheduling, shuffled frog-leaping algorithm, multi-objective optimization, energy consumption

Recommended Citation

Zhang Xiaoxing, Wang Yan, Yan Dahu, Ji Zhicheng. Improved Shuffled Frog-Leaping Algorithm for Solving Flexible Job Shop Scheduling Problem[J]. Journal of System Simulation, 2017, 29(9): 2093-2099.

改进的混合蛙跳算法求解柔性作业车间调度

张晓星, 王艳, 严大虎, 纪志成

(江南大学物联网技术应用教育部工程研究中心, 无锡 214122)

摘要: 针对柔性作业车间调度问题, 构建了以最大完工时间和总加工能耗最小为优化目标的多目标调度模型。发展了改进的混合蛙跳算法, 通过设计基于 MPX(Maximum Preserved Crossover)操作和单亲遗传算法基因移位操作的局部更新策略, 避免了算法的非法解产生和修整, 加快了算法寻优速率。并通过简化邻域寻优策略对组内最优解进行优化, 防止算法陷入局部最优。通过求解某企业生产车间实例, 得出了不同权重下的调度方案, 并对比标准混合蛙跳算法下的最大完工时间和加工能耗, 证明了算法的有效性。

关键词: 柔性作业车间调度; 混合蛙跳算法; 多目标优化; 能耗

中图分类号: TP278

文献标识码: A

文章编号: 1004-731X (2017) 09-2093-07

DOI: 10.16182/j.issn1004731x.joss.201709029

Improved Shuffled Frog-Leaping Algorithm for Solving Flexible Job Shop Scheduling Problem

Zhang Xiaoxing, Wang Yan, Yan Dahu, Ji Zhicheng

(Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China)

Abstract: Aiming at the characteristics of flexible job shop scheduling problem, a multi-objective scheduling model with maximum completion time and minimum energy consumption was proposed. An improved shuffled frog leaping algorithm was developed. By designing the local updating strategy based on crossover operation of maximum preserved crossover (MPX) and shifting operation of single parent gene algorithm (PGA), it avoided the illegal solution and trimming of the algorithm. Additionally, it accelerated optimization rate of the algorithm. And the optimal solution of the group was optimized by the simplified neighborhood optimization strategy to prevent the algorithm from falling into the local optimum. Then, by solving an example of an enterprise production workshop, scheduling schemes under different weights were obtained. Compared with completion time and processing energy consumption obtained by the classical shuffled frog-leaping algorithm, it proves the effectiveness of the algorithm.

Keywords: flexible job shop scheduling; shuffled frog-leaping algorithm; multi-objective optimization; energy consumption

引言

随着以“强化智能应用、促进制造业产业升级”



收稿日期: 2017-05-17 修回日期: 2017-07-10;
基金项目: 国家自然科学基金(61572238), 江苏省杰出青年基金(20160001), 江苏省产学研联合创新资金-前瞻性联合研究项目(201602224);
作者简介: 张晓星(1994-), 女, 山东潍坊, 硕士生, 研究方向为车间调度; 王艳(1978-), 女, 江苏盐城, 博士后, 教授, 博导, 研究方向为制造系统能效优化。

为主题的中国制造 2025 的提出, 绿色制造、节能减排受到广泛的关注。车间生产制造过程中一直未摆脱高能耗、高排放、低效率的格局。生产柔性作业车间调度作为制造业的基石和核心管理技术, 即是要根据车间生产的实际情况, 尽量发展采用物料和能源消耗少、环境污染小的工艺方案和生产途径。因此, 对于制造车间的工艺方案、能效优化技术研究与应用有着迫切的需求。

<http://www.china-simulation.com>

• 2093 •

柔性作业车间调度问题(Flexible Job Shop Scheduling Problem, FJSP)相对于传统的作业车间调度问题更接近实际的生产环境,减少了机器约束,并扩大了可行解的优化组合方案。柔性作业车间调度问题是 NP-Hard 问题,具有复杂性、非线性、大规模性等特点。传统的 FJSP 的优化目标已经不能满足当前生产的需求,如吴秀丽等^[1]考虑最大完工时间、交货期满意度、生产成本以及设备利用率等目标。张国辉等^[2]分析柔性作业车间调度问题的特点,考虑各个机器的负荷平衡、所有机器总负荷以及最大完工时间为优化目标。随着低碳环保成为社会的主调,能耗、低碳排放也成为调度的重要优化目标,如 Liu^[3]等提出的流水作业车间的混合整数非线性规划能耗模型。蒋增强等^[4]深入分析了柔性作业车间调度问题,运用基于血缘变异的改进的非支配排序遗传算法,考虑了设备状态—能耗的模型。

多类智能算法用来求解 FJSP,如遗传算法、粒子群算法、蚁群算法等。混合蛙跳算法^[5-6]是由 Muzaffar Eusuff 和 Kevin Lansey 在 2003 年提出的一种后启发式群体进化算法,具有参数少,全局搜索寻优能力强等优点。混合蛙跳算法的基本思想是模拟现实自然环境中青蛙群体在寻找食物的过程中所体现的信息交流和协同合作行为,从而实现问题的求解。自从该算法提出,已有学者对算法进行改进并应用于 FJSP。Li 等^[7]通过引入局部寻优算子对混合蛙跳算法进行改进,增强了算法的局部寻优能力,解决了柔性作业车间调度问题。Xu 等^[8]基于关键路径的局部寻优策略,采用实验设计的方法进行参数设置,有效的改进了混合蛙跳算法。

综上所述,本文考虑设备能耗和资源约束的基础上,根据现实生产环境,构建以完工时间和加工能耗为指标的多目标优化调度模型。采用改进的混合蛙跳算法,通过 MPX 交叉操作和单亲遗传算法基因移位操作优化局部更新策略,并通过简化邻域寻优策略对组内最优解进行优化,避免了蛙跳算法的非解产生和早熟的缺点,加快算法寻优速率,

验证了改进的算法在求解生产调度车间的可行性和有效性。

1 问题描述

1.1 柔性作业车间调度的模型

FJSP 可以描述为:设有 m 台机器 $\{M_1, M_2, \dots, M_m\}$, 可以加工 n 个待加工工件 $\{J_1, J_2, \dots, J_n\}$, 每个工件有多道工序,且工序数量不一定相同,每道工序可以从可选机器集中选择一台机器加工,不同的加工顺序和机器选择将导致不同的最大完工时间和能耗。且要满足以下约束条件:

- 1) 每个工件的一道工序在同一时刻只能由一台机器进行加工。
- 2) 不同工件之间没有先后约束,同一工件的工序之间存在先后约束。
- 3) 工件加工过程不能中止。

为了方便讨论,定义以下变量。

m : 机器总数

n : 工件总数

i : 工件序号

j : 工序序号

k : 机器序号

O_{ij}^k : 第 i 个工件的第 j 道工序且加工机器为 k ;

t_{ijk} : 第 i 个工件的第 j 道工序在机器 k 上加工时间;

C_{\max} : 最大完工时间;

E : 表示加工能耗;

e_{ijk} : 工件 i 的第 j 道工序在机器 k 上的加工能耗;

a_{ijk} : 工件 i 的第 j 道工序能否由机器 k 加工,当 $a_{ijk}=1$, 则可以加工, $a_{ijk}=0$ 则表示不可以加工。

1.2 FJSP 的调度目标

FJSP 的调度目标是找出 n 个工件所有工序的加工顺序,并分配相应的机器,使最大完工时间以及加工能耗最小。以下给出了 MOFJSP 问题的具体数学模型。

(1) 最小最大完工时间

$$C_{\max} = \max C_i, \quad i = 1, 2, \dots, n \quad (1)$$

(2) 最小加工能耗

$$E = \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{k=1}^m t_{ijk} a_{ijk} e_{ijk} \quad (2)$$

(3) 基于决策者偏好调度方案下目标函数值表示如下, 其中, λ_1 和 λ_2 分别表示最大完工时间和加工能耗的权重。

$$\min f(x) = \lambda_1 C_{\max} + \lambda_2 E \quad (3)$$

2 蛙跳算法的设计与改进

2.1 标准混合蛙跳算法

混合蛙跳算法模拟自然界青蛙种群的觅食和进化过程。初始化种群后, 按照目标函数值的好坏进行降序排列, 并将整个种群分成 p 个模因组, 每个组中有 q 个青蛙个体。具体的分组规则可描述为: 序列第一个个体进入第一组, 第二个个体进入第二组, 依次下去, 第 $p+1$ 个青蛙个体再次进入第一组, 直至最后一个个体进入第 q 组。随着模因组的进化而实现整个种群的进化, 且记录每次进化过程中的组内最好解 p_b 和最差解 p_w , 以及整个种群的最好解 p_g 。随后, 运用一定的局部更新策略来对最差解进行位置调整。每个模因组达到进化次数之后, 将所有的模因组混合成新的种群, 实现所有模因组的信息交互, 算法达到最大进化代数后停止。局部更新策略公式如下:

青蛙移动步长:

$$D_s = rand \times (p_b - p_w) \quad (4)$$

最差青蛙更新位置:

$$p'_w = p_w + D_s, \quad D_{s\min} \leq D_s \leq D_{s\max} \quad (5)$$

其中: $rand$ 表示 0~1 之间的均匀任意数; p_b 表示进化过程中最好青蛙的位置, 即组内最优解; p_w 表示进化过程中最差青蛙的位置, 即组内最差解; D_s 表示青蛙移动步长; $D_{s\max}$ 与 $D_{s\min}$ 分别表示移动步长的最大值和最小值。

若运用局部更新策略后的 p_w 效果不理想, 则将公式中的 p_b 替换为 p_g 。

青蛙移动步长:

$$D_s = rand \times (p_g - p_w) \quad (6)$$

最差青蛙更新位置:

$$p'_w = p_w + D_s, \quad D_{s\min} \leq D_s \leq D_{s\max} \quad (7)$$

式中: p_g 表示整个种群中最好青蛙的位置, 即全局最优解。若更新后的 p_w 仍然不理想, 则随机产生一个新解来代替最差解。

2.2 编码和解码

本文采用双层式编码方式, 一层基于工序的编码, 确定工序的加工顺序; 一层是基于机器的编码, 给工序分配加工机器。例如对 3 工件 3 机器示例的编码如下。由图 1 可知, 工件的加工工序为 $O_{21}^1, O_{11}^3, O_{31}^1, O_{12}^2, O_{22}^2, O_{13}^3, O_{32}^2, O_{33}^1$ 。

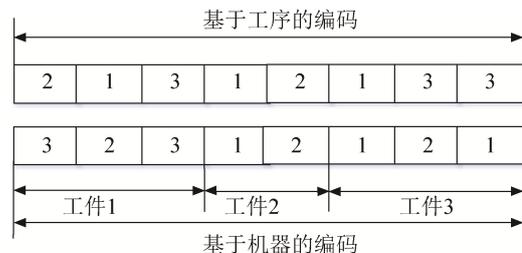


图 1 编码方式
Fig. 1 Coding method

解码是根据工序的编码的加工顺序, 并根据个体基于机器的编码找到工序对应加工机器的过程。如图 1, 表示第 1 个工件的第 1 道工序可以由第 3 台机器来加工, 第 2 个工件的第一道工序可以由第 1 台机器进行加工, 以此类推。

2.3 混合蛙跳算法的改进

混合蛙跳算法具有参数少, 全局搜索寻优能力强等优点。但是, 与其他智能算法一样, 混合蛙跳算法也存在缺陷, 比如直接用来求解离散问题存在不足, 易产生无效解、易早熟。本文通过改进蛙跳算法的局部更新策略, 对最差解更新的同时, 兼顾最优解进行了更新, 可以有效地解决标准蛙跳算法易早熟的缺点。

针对局部更新策略中最差值 p_w , 本文主要采用基于组内最优解 p_b 进行 MPX 交叉操作, 若更新

后的 p'_w 效果并不起作用, 则将交叉操作对象变更为 p_g , 若效果依然不理想, 则采用单亲遗传算法基因移位操作优化局部更新策略。更新过程示意图如图 2 所示。

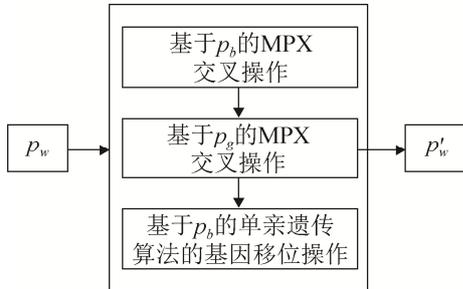


图 2 p_w 更新过程示意图
Fig. 2 Update process of p_w

针对局部更新策略中组内最优解 p_b , 基本的混合蛙跳算法并没有对其进行更新, 本文主要采用基于组内最优解 p_b 来搜索邻域解 p'_b 。更新过程示意图如图 3 所示。



图 3 p_b 更新过程示意图
Fig. 3 Update process of p_b

2.3.1 局部更新策略的改进

对于柔性作业车间调度问题, 其可行解代表的是工序和加工机器, 具有离散性。如果直接使用标准混合蛙跳算法的步长更新和位置更新公式, 会产生大量的不可行解。且经过分析发现, 混合蛙跳算法的局部更新策略是通过最优解的同化作用, 使最差解不断向最优解靠近来实现种群的更新。其中 $rand$ 是同化作用的强弱程度。若 $rand$ 越接近于 1, 则 p_b 对 p_w 影响越大; 反之, 则影响越小。因此, 可以借鉴遗传算法中的 MPX 交叉方式的思想来对混合蛙跳算法进行改进, 如图 7 所示。随机产生一组与基于工件编码长度相同的 0-1 数组^[8]。对应数组为 1 时, 则 p_w 中对应位置的数据保留; 对应数组为 0 时, 则用 p_b 中的对应位置数据替换 p_w 中对应位置的数据。在该过程中只更新 p_w , 并不对 p_b 进行改进。

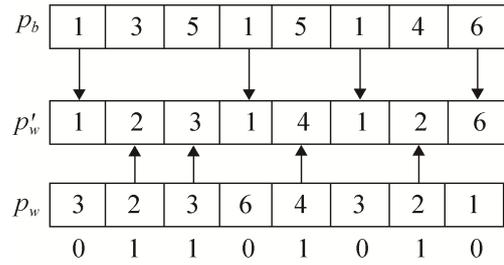


图 4 最差解 p_w 的更新

Fig. 4 Updating process of frog in each memplex with worst fitness value p_w

与标准混合蛙跳算法流程相同, 若交叉后产生的新的 p_w 效果不如之前的 p_w 效果理想, 则采用 p_g 与 p_w 进行 MPX 交叉操作。相比第 3 种情况标准混合蛙跳算法随机产生一个新解的操作, 进化方向不确定性增大, 对算法的寻优能力有较大的影响。本文采用单亲遗传算法(Single Parent Genetic Algorithm, PGA)中的基因移位思想来重新定义更新策略。首先定义一个变量常数 R , 表示 p_w 与 p_b 完全一致所要经过的基因移位次数, 则可设计每次更新 p_w 进取整 $0.5 \times R$ 次基因移位, 例如 $R=4$, 则对 p_w 进行 2 次基因移位。基因移位是基于最优解 p_b 进行的, 旨在让最差解向最优解靠拢。

假设 $R=4$, 则进行 2 次基因移动。取例子 $p_b = [1, 3, 5, 4, 3, 6, 2, 1]$, $p_w = [1, 3, 3, 1, 4, 5, 6, 2]$ 来简单说明优化过程。由于 p_w 与 p_b 前面两个元素相同, 则从第 3 个元素开始查找, p_b 的第三个元素是 5, 所以从 p_w 中选取第一个是 5 的元素, 并移到第三个元素的位置。 p_w 中其余的元素依次往后移动一位, 第一次单点基因移位后, $p_w = [1, 3, 5, 3, 1, 4, 6, 2]$ 。同理, 第四个元素 4 也按照相同的步骤移到 p_w 的第 4 个位置, 则第二次单点基因移位操作后, $p_w = [1, 3, 5, 4, 3, 1, 6, 2]$, 两次基因移位已经完成, 得到最后更新的解 p_w 。计算更新后 p_w 的目标函数值, 若优于原值, 则取代原来模因组中的青蛙; 若没有优化, 则用 p_g 取代 p_b , 则按照上述步骤更新 p_w 。经过多次的优化, 一般情况下 p_w 的值相比前解会得到较好的改进。具体优化过程如下图所示。

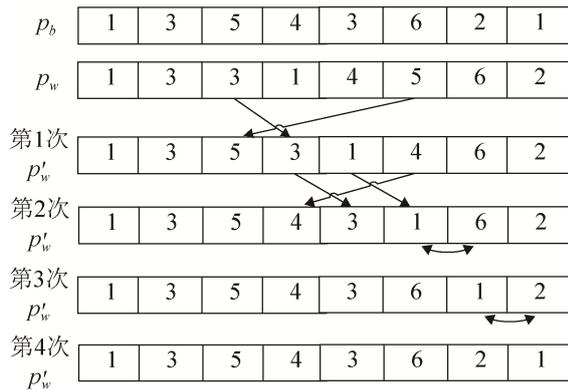


图 5 基于单亲遗传算法的混合蛙跳算法更新策略
Fig. 5 Updating strategy for shuffled frog-leaping algorithm based on PGA

2.3.2 局部最优 p_b 的更新

标准的混合蛙跳算法只更新最差青蛙的位置 p_w , 而对组内最好青蛙的位置 p_b 不作处理, 这导致标准混合蛙跳算法易陷入局部最优。与文献[7-8]不同, 为了使混合蛙跳算法跳出局部最优, 本文对 p_b 进行简化邻域搜索方法来更新。从组内最好青蛙位置 p_b 开始, 通过邻域动作, 产生其邻居解, 计算其邻居解的目标函数值, 以一定的概率来确定是否来选择接受邻居解。若邻居解 p'_b 被接受, 则更新最内最好青蛙位置 p_b , 且比较更新后的 p'_b 和全局最优青蛙位置 p_g , 若 p'_b 优于全局最优 p_g , 则将 p_b 代替更新 p_g 。可将局部最优更新 p_b 的主要方法伪代码表示如下:

```

DE =  $p'_b - p_b$ ;
If  $DE < 0$ 
     $p_b = p'_b$ ;
Else If  $\exp(-DE / \text{Maxgen}) \geq \text{rand}$ 
     $p_b = p'_b$ ;
Else
     $p_b = p_b$ ;
End

```

其中, DE 代表更新后的组内最优解与更新前最优解的差, Maxgen 表示种群最大进化代数, rand 表示 0~1 之间的随机数。

2.4 算法解决 FJSP 流程

改进的混合蛙跳算法解决 FJSP 问题的主要流

程如下:

Step 1: 初始化种群, 确定算法各参数值。

Step 2: 按一定的规则形成模因组, 并记录整个种群的最优解 p_g 和组内最优解 p_b 、最差解 p_w 。

Step 3: 每个模因组按照 2.3.1 节和 2.3.2 节的更新策略执行组内搜索。

Step 4: 每个模因组达到进化次数之后, 将所有的模因组混合成新的种群, 实现所有模因组的信息交互。

Step 5: 若没有达到终止条件, 则转到 Step 2。达到终止条件, 则算法结束。

3 仿真实验

3.1 实验设置

某企业生产车间制造过程可以简化为 6 台机器加工 6 个工件的柔性作业车间调度问题。加工工件和机器的约束如表 1 所示。表中第一列表示 6 个工件, 第二列表示每个工件的工序, 第 3~8 列表示可以选择加工的机器, 分别对应着加工时间和加工能耗, “-”表示不可以在该机器上进行加工。

混合蛙跳算法的主要参数设置如下: 种群规模 $F=80$, 整个种群进化代数 $\text{Maxgen}=200$, 子种群进化代数 $N_e=50$, 最大移动步长 $D_s=2$ 。

3.2 结果分析

基于改进的混合蛙跳算法和标准混合蛙跳算法, 分别对 5 种情况最大完工时间和能耗权重值进行优化求解, 取 10 次运行的结果平均值如表 2 所示。黑色加粗表示所计算的较好的值。

如表 2 所示, 分别给出了标准混合蛙跳算法和改进的混合蛙跳算法的不同调度方案下的结果。由相同权重下的调度方案可见, 改进的混合蛙跳算法求得的最大完工时间和能耗的值都更小, 如在权重为 0.5, 0.5 时, 改进的混合蛙跳算法求得的最大完工时间值为 73, 明显优于标准的混合蛙跳算法值 91.7; 改进的混合蛙跳算法的能耗的值为 118, 也是优于标准的混合蛙跳算法的值 120.5。

表 1 FJSP 数据
Tab. 1 Data of FJSP

工件	工序	加工时间/单位时间加工能耗					
		M_1	M_2	M_3	M_4	M_5	M_6
J_1	O_{11}	10/3	15/2.5	-	14/2.7	-	14/1.5
	O_{12}	-	6/3	4/2.1	16/2.5	15/3	-
	O_{13}	-	5/2.4	-	-	16/2.5	8/1.7
	O_{14}	12/2	-	13/2	13/2	-	-
	O_{15}	6/3.1	6/3.1	8/1.5	8/2	4/2.1	9/1.4
	O_{16}	-	-	16/1.8	-	13/2	16/1.6
J_2	O_{21}	15/2.5	-	6/1	16.5/2.2	11/2.5	-
	O_{22}	-	15/2	10/1.5	7/2.1	-	12/1.5
	O_{23}	5/2.3	-	16/1.8	10/2.3	14/2.3	-
J_3	O_{31}	14/2.9	15/2.7	6/1.7	5/2.8	4/3.1	-
	O_{32}	-	5/2	6/2	-	16/2.2	-
	O_{33}	5/2.8	8/2.3	-	11/2.9	-	15/2
	O_{34}	-	6/2.5	17/2.1	14/2.5	12/2	-
	O_{35}	17/2.5	-	14/2.4	7/2	7/2.8	11/2.1
J_4	O_{41}	20/3	-	19/2	13/2.5	15/2.3	-
	O_{42}	-	10/2.5	7/1	14/2.8	7/3	15/2
	O_{43}	4/2.8	8/2	-	-	-	16/2
	O_{44}	9/3.2	-	6/1.6	-	6/2.5	-
	O_{45}	16/2.9	9/2.3	16/2.1	13/2	-	-
J_5	O_{51}	-	6/2.2	-	7/2.8	12/2	8/0.9
	O_{52}	8/3	-	12/2.7	16/2.7	-	6/1
	O_{53}	13/2.5	12/2.3	-	-	16/1.8	8/0.7
	O_{54}	-	4/2	6/2	5/3	12/1.5	-
	O_{55}	13/3.2	-	-	8/2.5	-	9/1.2
	O_{56}	11/2.8	3/2.5	10/2.5	12/2.4	16/2	5/0.9
J_6	O_{61}	-	11/1.2	-	-	7/1.9	8/0.8
	O_{62}	-	-	8/2	12/3	-	6/1
	O_{63}	10/2.4	5/1	-	13/2.1	6/2	-
	O_{64}	16/2.2	-	8/1.7	-	-	12/1.1

表 2 决策者不同偏好下的调度方案结果
Tab. 2 Scheduling schemes under different weights of decision makers

调度方案	完工时间权重	能耗权重	最大完工时间	加工能耗
1	1	0	61 \85	193.2 \196.9
2	0.8	0.2	70 \76	164.1 \165.5
3	0.5	0.5	73 \91.7	118 \120.5
4	0.3	0.7	86 \106	105 \114.7
5	0	1	110 \141.5	102.4 \112.2

注：表格中第 4 列和第 5 列的*，前一个*代表改进的混合蛙跳算法下的计算结果，后一个*代表基本混合蛙跳算法的计算结果。

且由表 2 可以看出，在 5 个不同的权重值下，

得出来的最大完工时间和能耗的值是不同的，且与相应的权重值成反比，即权重值越大，得到的相应目标的值就越小，结果越好。同时，决策者的偏好至关重要，不同的偏好可以得到不同的目标值和不同的调度方案。因此，决策者可根据实际的生产状况，选取合适的调度方案。

对权重为 0.5, 0.5 的调度方案分别给出了算法进化过程中目标函数值的变化，如图 6 和图 7 所示。横坐标代表进化代数，纵坐标表示目标函数值。由图 6 可见，随着代数增加，目标函数值在不断减小，而图 7 曲线存在波动现象，说明改进的混合蛙跳算法在 130 代已经达到了最优，其收敛性更好。

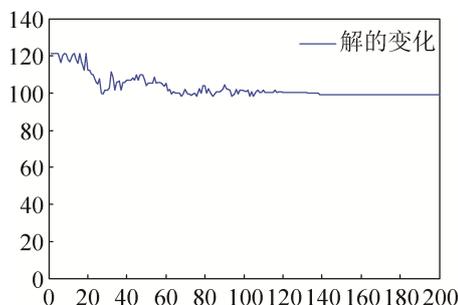


图 6 基于改进混合蛙跳算法解的变化

Fig. 6 Change of solution based on improved shuffled frog-leaping algorithm

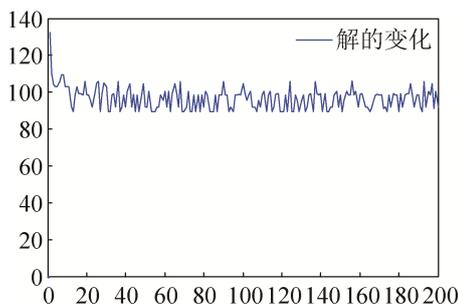


图 7 基于标准混合蛙跳算法解的变化

Fig. 7 Change of solution based on classical shuffled frog-leaping algorithm

本文就改进的混合蛙跳算法和标准混合蛙跳算法下的完工时间和能耗权重为 0.5, 0.5 分别给出完工时间的甘特图见图 8~9。其中，纵坐标表示加工机器，横坐标表示时间。由甘特图可见，改进的算法在机器上的完工时间更小，且在各台机器上加工分布更为均匀，不会出现一台机器拥堵的情况。

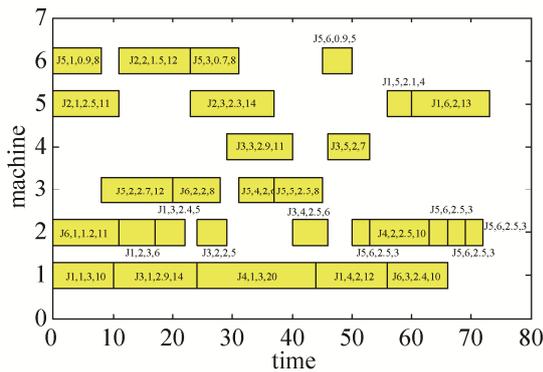


图 8 改进混合蛙跳算法的权重 0.5,0.5 甘特图

Fig. 8 Gantt chart on weight of 0.5,0.5 adopting improved shuffled frog-leaping algorithm

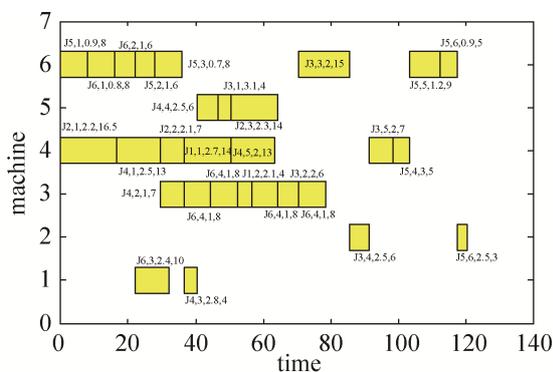


图 9 标准混合蛙跳算法的权重 0.5,0.5 甘特图

Fig. 9 Gantt chart on weight of 0.5,0.5 adopting classical shuffled frog-leaping algorithm

4 结论

通过对柔性作业车间调度问题的分析和研究,构建了以最大完工时间和能耗最小为目标函数的调度模型。针对标准混合蛙跳算法直接用来求解离散问题存在易产生无效解、易早熟的缺陷,采用基于 MPX 交叉操作和单亲遗传基因移位操作,对局部更新策略进行了改进。使用 MATLAB 对简化的实际柔性作业车间调度进行了仿真,给出了基于决策者不同偏好下的调度方案,并给出了对应方案的调度甘特图,得出改进的混合蛙跳算法,相比标准混合蛙跳算法,具有更好的收敛性,也证明了其解决柔性作业车间调度问题的可行性。

参考文献:

[1] 吴秀丽, 孙树栋, 余建军, 等. 多目标柔性作业车间调度优化研究 [J]. 计算机集成制造系统, 2006, 12(5):

731-736. (Wu X L, Sun S D, Yu J J, et al. Research on multi-objective optimization for flexible job shop scheduling [J]. Computer Integrated Manufacturing Systems, 2006, 12(5): 731-736.)

- [2] 张国辉, 高亮, 李培根, 等. 改进遗传算法求解柔性作业车间调度问题 [J]. 机械工程学报, 2009, 45(7): 145-151. (Zhang G, Gao L, Li P, et al. Improved Genetic Algorithm for the Flexible Job-shop Scheduling Problem [J]. Journal of Mechanical Engineering, 2009, 45(7): 145-151.)
- [3] Liu X, Zou F, Zhang X. Mathematical model and genetic optimization for shuffled flow shop scheduling problem based on energy consumption [C]// Control and Decision Conference, 2008. USA: IEEE, 2008: 1002-1007.
- [4] 蒋增强, 左乐. 低碳策略下的多目标柔性作业车间调度 [J]. 计算机集成制造系统, 2015, 21(4): 1023-1031. (Jiang Zengqiang, Zuo Le. Multi-objective flexible job-shop scheduling based on low-carbon strategy [J]. Computer Integrated Manufacturing Systems, 2015, 21(4): 1023-1031.)
- [5] Muzaffar Eusuff, Kevin Lansey, Fayzul Pasha. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization [J]. Engineering Optimization (S0305-215X), 2006, 38(2): 129-154.
- [6] Eusuff M M, Lansey K E. Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm [J]. Journal of Water Resources Planning & Management (S0733-9496), 2003, 129(3): 210-225.
- [7] Li J, Pan Q, Xie S. An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems [J]. Applied Mathematics & Computation (S0096-3003), 2012, 218(18): 9353-9371.
- [8] Xu Y, Wang L, Liu M, et al. An effective shuffled frog-leaping algorithm for shuffled flow-shop scheduling with multiprocessor tasks [J]. The International Journal of Advanced Manufacturing Technology (S0268-3768), 2013, 68(5): 1529-1537.
- [9] 张国辉, 党世杰. 考虑机器速度的低碳柔性作业车间调度问题研究 [J]. 计算机应用研究, 2017, 33(4): 1072-1075. (Zhang Guo-hui, Dang Shi-jie. Research on low carbon flexible job shop scheduling problem considering machine speed [J]. Application Research of Computers, 2017, 33(4): 1072-1075.)
- [10] Lei D, Guo X. A shuffled frog-leaping algorithm for job shop scheduling with outsourcing options [J]. International Journal of Production Research (S0020-7543), 2016, 54(16): 1-12.