

DOI: 10.11992/tis.201801041

网络出版地址: <http://kns.cnki.net/kcms/detail/23.1538.tp.20180503.1008.003.html>

应用改进区块遗传算法求解置换流水车间调度问题

裴小兵, 张春花

(天津理工大学管理学院, 天津 300384)

摘要: 针对最小化最大完工时间的置换流水车间调度问题, 提出一种将遗传算法与蚁群算法相结合的改进区块遗传算法。算法利用随机机制和改进反向学习机制相结合的方式产生初始解, 以兼顾初始种群的多样性和质量。通过若干代简单遗传算法操作产生精英群体, 借鉴蚁群算法中利用蚂蚁信息度浓度统计路径和节点信息的思想, 对精英群体所携带信息进行统计分析并建立位置信息素矩阵和相依信息素矩阵, 根据两矩阵挖掘区块并将区块与非区块组合形成染色体。将染色体进行切段与重组, 以提高染色体的质量, 使用二元竞赛法保留适应度较高的染色体。算法通过 Reeves 实例和 Taillard 实例进行测试, 并将结果与其他算法进行比较, 验证了该算法的有效性。

关键词: 生产调度; 组合优化; 遗传算法; 蚁群优化算法; 构建区块; 人工染色体

中图分类号: TP18 **文献标志码:** A **文章编号:** 1673-4785(2019)03-0541-10

中文引用格式: 裴小兵, 张春花. 应用改进区块遗传算法求解置换流水车间调度问题[J]. 智能系统学报, 2019, 14(3): 541-550.

英文引用格式: PEI Xiaobing, ZHANG Chunhua. An improved puzzle-based genetic algorithm for solving permutation flow-shop scheduling problems[J]. CAAI transactions on intelligent systems, 2019, 14(3): 541-550.

An improved puzzle-based genetic algorithm for solving permutation flow-shop scheduling problems

PEI Xiaobing, ZHANG Chunhua

(School of Management, Tianjin University of Technology, Tianjin 300384, China)

Abstract: Targeting the permutation flow-shop scheduling problem that minimizes maximum completion times, an improved block genetic algorithm combined with an ant colony algorithm is proposed. The algorithm uses a random mechanism and the improved opposition-based learning mechanism to generate the initial solution. It takes into account the diversity and quality of the initial population. Elite populations are generated through several generations of simple genetic algorithm operations. Based on the idea of using ant information density concentration's statistical path and node information in the ant colony algorithm, the information carried by elite groups was counted. Position pheromone and dependent pheromone matrices were established. Mining blocks according to two matrices were developed and blocks were combined with non-blocks to form artificial chromosomes. Chromosomes were cut and recombined to increase chromosome quality. The binary race method was used to retain chromosomes with higher fitness. The algorithm was tested through the Reeves and Taillard instances, and the results were compared with other algorithms to verify effectiveness of the algorithm.

Keywords: production scheduling; combinatorial optimization; genetic algorithms; ant colony optimization; building block; artificial chromosome

收稿日期: 2018-01-23. 网络出版日期: 2018-05-03.

基金项目: 国家创新方法工作专项项目 (2017IM060200); 天津市哲学社会科学规划项目 (TJYY17-013).

通信作者: 张春花. E-mail: Zhang_chunhua1203@126.com.

置换流水车间调度问题 (permutation flow-shop scheduling problem, PFSP) 是广泛受到研究的一种组合优化问题, 属于 NP-hard 问题。PFSP 的搜索

空间中存在诸多局部最优解,且问题复杂度越高,越容易局部最优化,因此许多学者针对 PFSP 特性不断对求解 PFSP 的算法进行优化。求解 PFSP 的算法可分为精确算法和近似算法,精确算法虽能求出最优解,但时间复杂性较高,随着问题的规模增大将变得不可行。近似算法对于较大规模的问题能在可行时间内求出近优解,是研究 PFSP 问题的重要方法。近似算法包括遗传算法^[1]、蚁群算法^[2]、模拟退火算法^[3]等。

遗传算法 (genetic algorithm, GA) 通过选择、交叉和变异等操作搜索解空间的最优解,已经成功应用于解决旅行商问题、车辆路径问题和车间调度问题等不同领域的组合问题。但是遗传算法的局部搜索能力较差,在实际应用中容易产生早熟收敛的问题。怎么运用遗传算法既能保留优良个体,又能维持群体的多样性,一直是遗传算法中较难解决的问题。近年来针对 PFSP,不少学者为提高遗传算法的有效性,将遗传算法与其他算法相结合对该问题进行求解,取得了不错的效果。Benbouzid 等^[4]针对 PFSP 提出 VacGA(introduces vaccination into the field of GAs),将遗传算法和免疫算法相结合,遗传算法执行全局搜索,人工免疫系统执行局部搜索,以克服演化过程中遗传算法局部搜索能力不足的问题。Chen 等^[5]将遗传算法与广义邻域搜索相结合,提出一种新型混合遗传算法求解 PFSP。Bessedik 等^[6]提出一种基于免疫的遗传算法,这种算法分为两种形式:1) 将接种引入到基于免疫理论的遗传算法;2) 从免疫网络理论得到启发,并将其运用到遗传算法中。Ganguly 等^[7]将贪婪局部搜索算子引入到遗传算法,有效地解决大范围的排序和调度离散优化问题。齐学梅等^[8]为求解 PFSP 问题,将变邻域搜索算法与遗传算法相结合,加快了收敛速度。吴秀丽等^[9]将变邻域搜索算法和遗传算法相结合提出了一种变邻域改进遗传算法求解混合流水车间调度问题,增强了遗传算法的局部搜索能力。

Chang 等^[10]针对置换流水车间调度问题,提出一种基于区块的遗传算法 (puzzle-based artificial chromosome genetic algorithm, p-ACGA),将遗传算法和蚁群算法相结合,通过简单遗传算法的交叉、突变等操作产生精英群体,针对这些精英群体,利用蚁群算法的信息素浓度对工件与工件的关系进行统计分析,进而建立信息素相依矩阵并根据矩阵挖掘区块,利用区块和非区块重组形

成人工染色体,然后将其注入到进化过程中。虽然 p-ACGA 能够避免局部最优问题,一定程度上改善了解的质量,加快了进化速度,但是算法在初始化、信息素分布统计等方面还存在一些不足。因此本文提出一种改进区块遗传算法 (puzzle-based improved artificial chromosome genetic algorithm, p-IACGA)。该算法在种群初始化时引入反向学习的思想,并对其进行了改进,将其与随机机制相结合产生初始解,大大提高了初始解的多样性和质量;为进一步描述信息素在路径上的分布情况,算法考虑了信息素在节点上的分布情况,即工件与所在解序列位置的关系,并将其与 p-ACGA 中的统计分析工件与工件的相邻关系相结合,以全面地描述解的空间分布情况。然后,根据这两种关系建立相依信息素矩阵和位置信息素矩阵,进而挖掘区块以组合人工染色体。最后,引入染色体重组机制,进一步提高解的质量。

1 置换流水车间生产调度问题描述

PFSP 生产调度问题可以描述为:有 n 个作业以相同的顺序在 m 台机器上加工。同时,要满足的约束条件包括:1) 操作是独立的,可以在零时间开始;2) 每台机器在同一时间最多加工一个工件;3) 每一个工件在同一时间只能在一台机器上加工。用 $P(i, j)$ 表示工件 i 在机器 j 上的处理时间,用 $(\pi_1, \pi_2, \dots, \pi_n)$ 表示工件序列,用 $C(\pi_i, j)$ 表示完成时间的计算,即

$$C(\pi_1, 1) = P(\pi_1, 1)$$

$$C(\pi_i, 1) = C(\pi_{i-1}, 1) + P(\pi_i, 1)$$

$$C(\pi_1, j) = C(\pi_1, j-1) + P(\pi_1, j)$$

$$C(\pi_i, j) = \max\{C(\pi_{i-1}, j), C(\pi_i, j-1)\} + p(\pi_i, j)$$

式中: $i = 2, 3, \dots, n; j = 2, 3, \dots, m$ 。

最大完工时间: $C_{\max}(\pi) = C(\pi_i, m)$ 。

适应度为: $f(x) = 1/C_{\max}(\pi)$ 。

2 求解置换流水车间调度问题的改进区块遗传算法

p-IACGA 主要分为 5 部分:产生初始解、挖掘区块、构建人工染色体、染色体重组和保留优势解。p-IACGA 流程如图 1 所示,因为染色体携带的较优信息是逐代积累的,所以为了挖掘出较优人工染色体,每执行 n (给定的一个数) 代遗传算法后挖掘一次区块,这样每一轮执行的总代数数是 $n+1$ 。

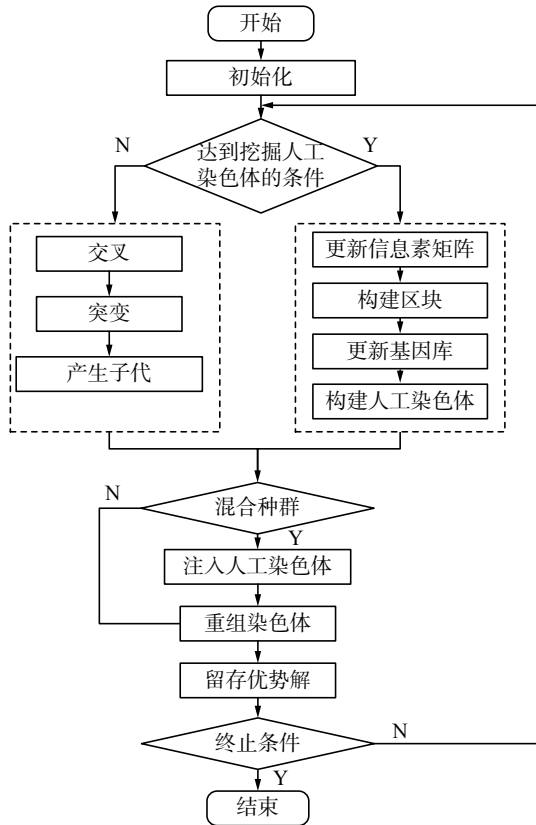


图1 p-IACGA 流程

Fig. 1 Flow chart of p-IACGA

2.1 改进反向学习法初始化种群

1) 反向学习法

反向学习法 (opposition-based learning, OBL)^[11] 的主要思想是同时考虑原始解序列和它的反向解序列, 以获得当前候选解中的最优解。假设 $X = (x_1, x_2, \dots, x_n)$ 是 n 维空间中的一个解, 其中 $x_i \in \mathbf{R}$ 和 $x_i \in [l_i, u_i]$, l_i, u_i 是搜索空间的上、下界。 $\forall i \in \{1, 2, \dots, n\}$, 那么它的反向解是 $X' = (x'_1, x'_2, \dots, x'_n)$, 其中 $x'_i = l_i + u_i - x_i$ 。OBL 优化过程为:

- ①在 n 维搜索空间中生成初始解 $X = (x_1, x_2, \dots, x_n)$ 和反向解 $X' = (x'_1, x'_2, \dots, x'_n)$;
- ②求出初始解的适应度 $f(x)$ 和其反向解的适应度 $f(x')$, 评估两个解的适应度;
- ③如果 $f(x') \geq f(x)$, 用 x' 替换 x ; 否则, 继续使用 x 。

通过同时评估这两个解可以获得更好的结果。

2) 应用改进反向学习法进行种群初始化

初始化过程采用随机机制和改进反向学习机制相结合的方式, 以兼顾初始解的多样性和质量。本研究对反向学习法作了改进以减少优秀解的丢失, 基本思路就是将所有原始解和所有反向解混合在一起形成混合群体, 然后从混合群体中选出适应度较高的解, 如图2所示。首先利用随机方式产生 N (N 为给定的初始种群规模) 条初始

解并对每条初始解求出它的反向解; 然后, 将两种群混合在一起形成规模为 $2N$ 的种群, 计算每个解的适应度, 并按照适应度大小对解降序排列; 最后, 选择前 N 条适应度较大的解作为进化的初始种群。

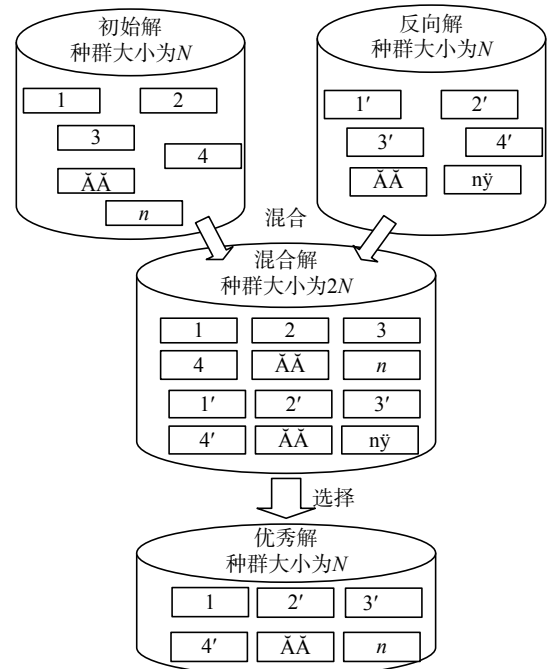


图2 改进反向学习法

Fig. 2 Improved opposition-based learning

2.2 挖掘区块

在遗传算法中, 进化若干代后, 大部分染色体都会携带与较优解相似的一些信息, 这些信息具有相似性并且这些相似性在大规模的种群中具有一定的可靠性^[12]。在 p-IACGA 中, 使用蚁群优化算法 (ant colony optimization, ACO) 中的蚂蚁信息素浓度来识别染色体中较好的相似序列并通过区块的方式将它们挖掘出来。

1) 建立信息素矩阵

本研究使用相依信息素矩阵和位置信息素矩阵来记录蚂蚁经过的路径信息, 其中位置信息素矩阵记录每个节点 (工件在解序列中的位置) 的信息素浓度信息, 相依信息素矩阵记录两节点间的路径 (两工件间的相邻关系) 的信息素浓度信息。矩阵建立过程如下:

①对信息素矩阵进行初始化

从第 n 代中选择一条最优染色体进行矩阵初始化。图3为初始相依信息素矩阵生成过程, 两个工件间的初始信息素表示为 τ_0 (见式 (1))。同样的方法生成初始位置信息素矩阵如图4。

$$\tau_0 = \frac{1}{L} \quad (1)$$

式中 L 表示完工时间。

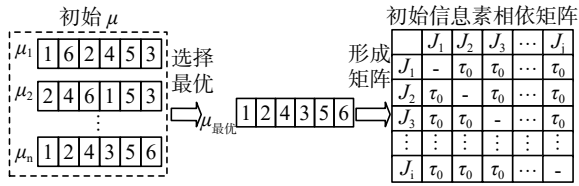


图 3 初始相依信息素矩阵

Fig. 3 Initially dependent pheromone matrix

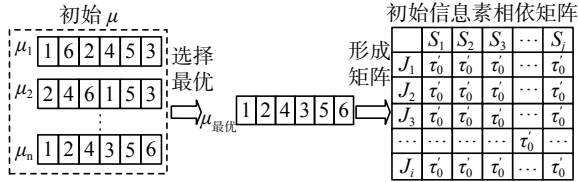


图 4 初始位置信息素矩阵

Fig. 4 The initial position pheromone matrix

②更新信息素矩阵

由遗传算法产生的最新一代的前 30% 适应度较高的染色体更新信息素矩阵, 更新过程如图 5, 更新公式如式 (2):

$$\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \rho \Delta \tau_{ij}(t+1)$$

$$\Delta \tau_{ij}(t+1) = \begin{cases} 1/L, & \text{若蚂蚁经过路径}(i,j) \\ 0, & \text{其他} \end{cases} \quad (2)$$

式中: τ_{ij} 表示每对相邻工件 (i,j) 间的信息素浓度; $\tau_{ij}(t+1)$ 表示更新的 τ_{ij} ; ρ 表示信息素的蒸发速率^[12]。



图 5 更新相依信息素矩阵

Fig. 5 Update dependent pheromone matrix

同理, 得到更新的位置信息素矩阵如图 6。更新公式如式 (3):

$$\tau'_{mi}(t+1) = (1-\rho) \times \tau'_{mi}(t) + \rho \Delta \tau'_{mi}(t+1)$$

$$\Delta \tau'_{mi}(t+1) = \begin{cases} 1/L, & \text{若蚂蚁经过的第 } m \text{ 个节点为节点 } i \\ 0, & \text{其他} \end{cases} \quad (3)$$

式中: τ'_{mi} 为蚂蚁经过的第 m 个节点为节点 i 时在节点 i 处释放的信息素。

	S_1	S_2	S_3	...	S_m
J_1	τ'_{11}	τ'_{12}	τ'_{13}	...	τ'_{1m}
J_2	τ'_{21}	τ'_{22}	τ'_{23}	...	τ'_{2m}
J_3	τ'_{31}	τ'_{32}	τ'_{33}	...	τ'_{3m}
...
J_i	τ'_{i1}	τ'_{i2}	τ'_{i3}	...	τ'_{im}

图 6 更新的位置信息素矩阵

Fig. 6 Updated positional information matrix

2) 构建区块

区块是简短的染色体片段, 本研究通过学习链接关系^[13]构建区块。首先确定区块的最小长度, 然后随机选择起始位置, 最后为各位置选择工序。在区块最小长度内, 起始位置采用位置信息素矩阵中的信息选择工序, 如图 7, 假设 $P'_{11} > P'_{21} > \dots > P'_{i1}$, 选择工件 J_1 放入起始位置 S_1 。其他位置采用位置信息素矩阵和相依信息素矩阵的合并信息, 以轮盘赌选择法 (roulette wheel section, RWS)^[13]对工序进行筛选, 已经入选的工件不再进行筛选, 如图 8, 假设 CP_2 最大, 选择工件 J_2 放入位置 S_2 。

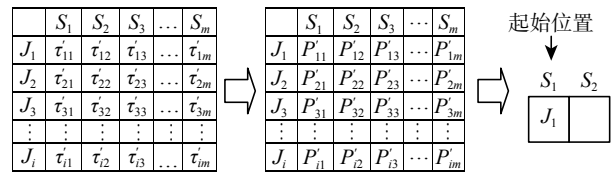


图 7 起始位置工序挑选

Fig. 7 Starting position process selection

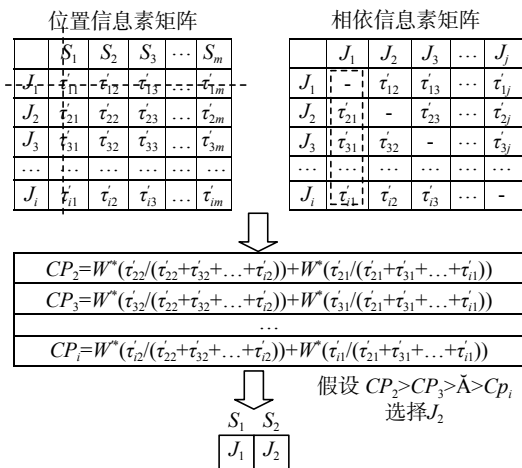


图 8 区块最小长度内其他位置工件选择

Fig. 8 Workpiece selection at other positions within the block's minimum length

当出现两个或多个较大值时随机选择。位置信息素浓度概率计算公式为

$$P'_{im} = \frac{\tau'_{mi}}{\sum_{i \in uns} \tau'_{mi}}, \quad m = 1, 2, \dots, n \quad (4)$$

相依信息素浓度概率计算公式为

$$P_{ij} = \frac{\tau_{ij}}{\sum_{i \in uns} \tau_{ij}}, \quad i, j = 1, 2, \dots, n \quad (5)$$

合并概率计算公式

$$CP_i = (W \times P_{ij}) + (W' \times P'_{im}), \quad i, j, m = 1, 2, \dots, n \quad (6)$$

式中: i 表示工件号; j 表示 i 所连接的上一个工件号; m 表示染色体上的工件的位置; n 表示染色体长度; P'_{im} 表示工件 i 与位置 m 在位置矩阵的概率; P_{ij} 表示相依矩阵中工件 j 相连于工件 i 之后

的概率; CP_i 表示工件 i 的合并概率; W' 与 W 分别表示位置矩阵与相依矩阵的权重值, 权重值会随着进化代数的增加不断改变^[14]。

对最小长度外的位置进行工序选择时设立合并概率最小阈值^[15], 作为筛选条件。当工件的合并概率大于最小阈值时, 选择该工件, 否则停止选择工件, 本区块完成挖掘, 如图9所示。因为当区块包含的工件越多总体概率越低, 组合错误的概率越大, 区块阈值将会保证区块的质量, 同时也将会导致挖掘出的区块长度也有所不同。挖掘的区块均存放在区块资料库中。

3) 区块竞争

区块竞争^[16]就是去除有重复信息的区块。将区块资料库中的区块的工序与位置进行比对, 如果区块之间出现重复的工序或涵盖的位置重复, 则通过比较平均概率的方式进行选择, 平均概率较大者保留至区块资料库, 较小者则删除, 举例如图10。平均概率的计算公式为

$$P_{B_i}^{AVG} = \frac{P_{B_i}^{dom} + \sum_{i=2}^n CP_{B_i}}{n}, \quad l = 1, 2, \dots, n \quad (7)$$

式中: i 表示区块号; l 表示区块的第 l 个位置; n

表示区块的长度; $P_{B_i}^{dom}$ 表示第 i 条区块的第 l 个工件的位置概率; CP_{B_i} 表示第 i 个区块的第 l 个工件的合并概率。

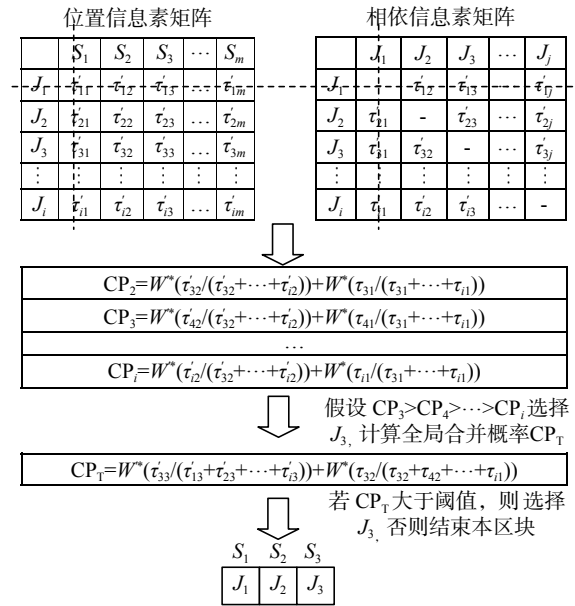


图9 区块最小长度外的工件选择

Fig. 9 Selection of workpieces outside the block minimum length

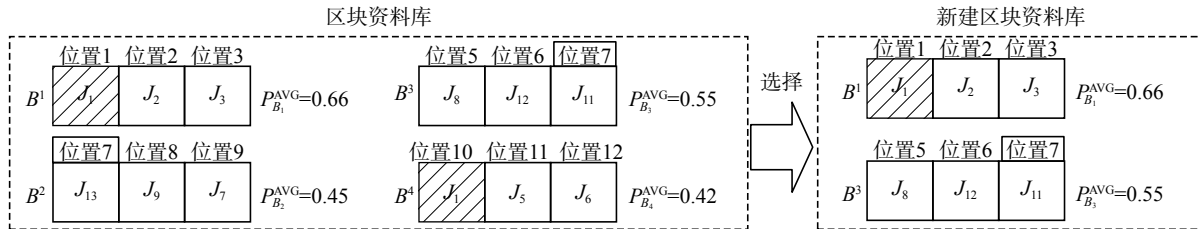


图10 区块竞争

Fig. 10 Block competition

2.3 组合染色体

本研究利用新建区块资料库中的区块和非区块资料库中的工件组合出较优染色体。先将区块资料库中的所有区块复制到确定长度的空白染色体对应位置, 对于尚未放入工序的空位置依照其

位置顺序 (由左到右), 利用 RWS 方法根据合并概率从剩余的工件中选择合适工件放入其中。例如图11 区块资料库中有2个区块, 非区块资料库有2个工件, 合并概率 CP_6 大于 CP_5 , 选择 J_6 放至位置5, 剩下 J_5 放至位置9。

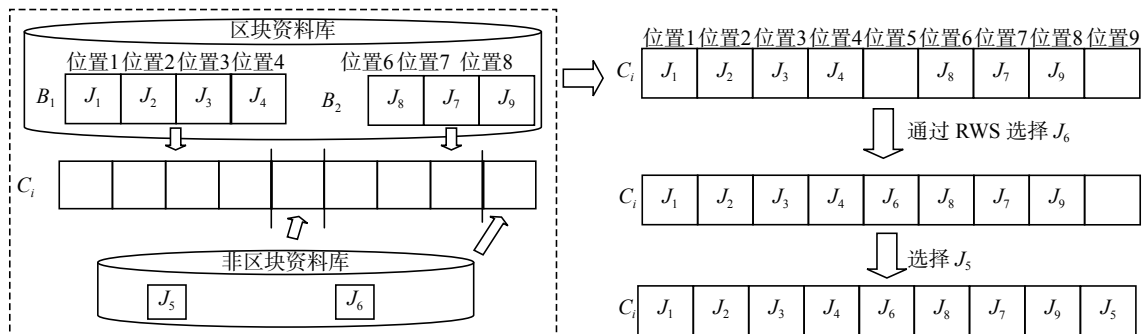


图11 染色体组合

Fig. 11 Chromosome combinations

2.4 染色体重组

为了进一步提高算法搜索到最优解的机会,对每一代染色体进行重组。首先随机选择 N 个切点,将一条完整的染色体分割成 $N+1$ 个片段,再从这些片段中选择出长度最长的片段进行重组。以工件数为10切点数为2为例,如图12,切割完成后的段数为3段,分别为{5, 2}、{2, 6, 10,

9, 1, 7}及{7, 4, 3, 8},选择长度较长的片段进行重组,过程如图13所示。

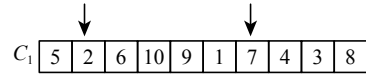


图12 切割人工染色体

Fig. 12 Cutting an artificial chromosome

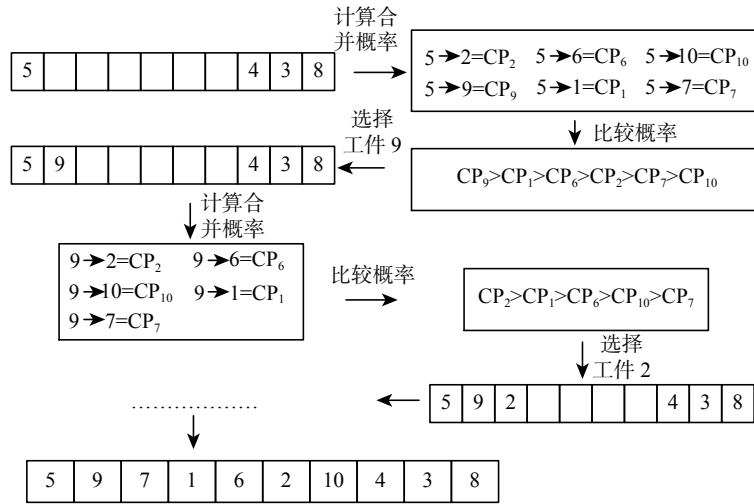


图13 人工染色体重组

Fig. 13 Artificial chromosome reorganization

2.5 留存优势解

将重组后的GA最新子代 μ_i 和人工染色体 C_i 放入选择池,使用二元竞赛法^[17]选择出优秀染色体作为子代进入下一代进化。首先,从选择池中随

机选择两条染色体,比较适应度,选择适应度较大的染色体放入染色体库,适应度较小的染色体放回选择池继续筛选,反复执行上述步骤,直到染色体库中染色体的数量满足设定的群体大小,如图14。

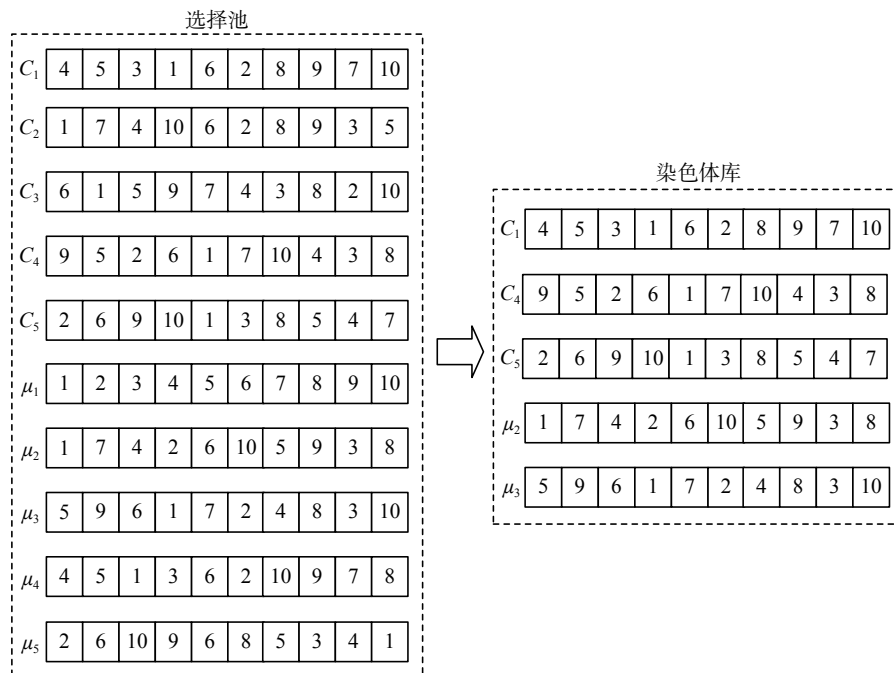


图14 留存优势解

Fig. 14 Retention of advantage solution

3 实验结果分析

为了评估所提出的算法的性能,本研究将通过对不同实例进行测试,使用误差率(ER)作为对比指标,与其他算法进行比较。误差率是指测试中算法所得最优解与实例已知最优解的差值占实例已知最优解的比例,ER计算公式为

$$ER_i = \frac{C_{\max} - U_i}{U_i} \times 100\% \quad (8)$$

式中: U_i 是实例的已知最优解; C_{\max} 是测试算法的最优解。

本文所提算法由C++语言编写,程序的运行环境为:处理器 Intel(R) Core(TM) i5-4005U CPU @ 3.40 GHz,内存为 4.0 GB 的计算机。在本文中,选择了 29 个调度问题进行比较,其中 21 个 Reeves 实例和 8 个 Taillard 实例,每个实例独立运行 30 次,并求出 ER 的平均值。为了达到性能评价的目的,在 SGA^[18]、ACGA^[19]、p-ACGA^[10]、BBEDA^[18]、LMBBEA^[20]、p-IACGA 的参数设置上

选择了最佳的、相同的参数配置以保证算法的可比性。

以 SGA、ACGA、p-ACGA、BBEDA、LMBBEA 和 p-IACGA 计算 Taillard 实例,结果如表 1 所示。从表 1 可以看出,ta005、ta010、ta020、ta030、ta070 和 ta080 这 6 个实例中的 ER 值均为 0,即找到了 6 个与实例已知解相同的解,多于其他算法所能找到的个数。尽管在 ta050 实例中,BBEDA 和 LMBBEA 的 ER 值为 0.85, p-IACGA 的 ER 值为 0.97,前两者的结果优于 p-IACGA,但是 p-IACGA 的平均值 ER 为 0.32%,均低于其他算法,表明 p-IACGA 的整体求解问题的性能优于其他算法。同理,用这些算法计算 Reeves 实例,结果如表 2。通过对 Reeves 的 21 个实例测试,发现 p-IACGA 在 Rec01、Rec03、Rec09、Rec11 和 Rec35 实例中 ER 的值为 0,找到了 5 个与实例已知解相同的解,多于其他算法,且 p-IACGA 的平均误差率为 0.46,均小于其他算法,充分表明 p-IACGA 求解 PFSP 的优良性能。

表 1 Taillard 实例测试结果比较

Table 1 Comparison of Taillard instance test results

算例	问题规模	U_i	BBEDA		LMBBEA		SGA		ACGA		p-ACGA		p-IACGA	
			C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER
ta005	20×5	1 235	1 235	0.00	1 235.00	0.00	2 630.55	1.13	1 840.15	0.49	1 235.00	0.00	1 235.00	0.00
ta010	20×5	1 108	1 108	0.00	1 108.00	0.00	2 925.12	1.64	1 894.68	0.71	1 163.40	0.05	1 108.00	0.00
ta020	20×10	1 591	1 989	0.25	1 591.00	0.00	5 727.60	2.60	4 597.99	1.89	1 781.92	0.12	1 591.00	0.00
ta030	20×20	2 178	2 287	0.05	2 286.90	0.05	8 951.58	3.11	5 466.78	1.51	2 221.56	0.02	2 178.00	0.00
ta050	50×10	3 065	5 670	0.85	5 670.25	0.85	19 156.25	5.25	15 416.95	4.03	6 773.65	1.21	4 444.25	0.45
ta060	50×20	3 696	17 704	3.79	17 703.84	3.79	34 298.88	8.28	28 274.40	6.65	16 114.56	3.36	11 568.48	2.13
ta070	100×5	5 322	5 322	0.00	5 322.00	0.00	7 823.34	0.47	7 078.26	0.33	5 428.44	0.02	5 322.00	0.00
ta080	100×10	5 845	5 845	0.00	5 845.00	0.00	13 501.95	1.31	11 748.45	1.01	6 604.85	0.13	5 845.00	0.00
平均值				0.62		0.59		2.97		2.08		0.61		0.32

表 2 Reeves 实例测试结果比较

Table 2 Comparison of Reeves instance test results

算例	问题规模	U_i	BBEDA		LMBBEA		SGA		ACGA		p-ACGA		p-IACGA	
			C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER
Rec01	20×5	1 247	1 259	0.01	1 284.41	0.03	9 926.12	6.96	1 446.52	0.16	1 384.17	0.11	1 247.00	0.00
Rec03	20×5	1 109	1 242	0.12	1 131.18	0.02	6 044.05	4.45	1 208.81	0.09	1 186.63	0.07	1 109.00	0.00
Rec05	20×5	1 242	1 689	0.36	1 502.82	0.21	5 986.44	3.82	1 540.08	0.24	1 589.76	0.28	1 242.00	0.00
Rec07	20×10	1 566	1 566	0.00	1 566.00	0.00	9 881.46	5.31	2 364.66	0.51	2 286.36	0.46	1 566.00	0.00
Rec09	20×10	137	171	0.25	137.00	0.00	785.01	4.73	289.07	1.11	146.59	0.07	137.00	0.00
Rec11	20×10	1 431	1 431	0.00	1 431.00	0.00	12 778.83	7.93	1 631.34	0.14	1 545.48	0.08	1 431.00	0.00
Rec13	20×15	1 930	2 586	0.34	2 277.40	0.18	13 452.10	5.97	3 531.90	0.83	2 644.10	0.37	2 219.50	0.15
Rec15	20×15	1 950	2 204	0.13	2 008.50	0.03	10 315.50	4.29	3 354.00	0.72	2 983.50	0.53	1 989.00	0.02
Rec17	20×15	1 902	2 358	0.24	1 921.02	0.01	13 466.16	6.08	4 907.16	1.58	3 899.10	1.05	2 415.54	0.27

续表 2

算例	问题规模	U_i	BBEDA		LMBBEA		SGA		ACGA		p-ACGA		p-IACGA	
			C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER	C_{\max}/s	ER
Rec19	30×10	2 093	3 746	0.79	3 202.29	0.53	14 797.51	6.07	5 211.57	1.49	4 018.56	0.92	2 741.83	0.31
Rec21	30×10	2 017	4 942	1.45	4 659.27	1.31	14 260.19	6.07	5 324.88	1.64	5 022.33	1.49	3 388.56	0.68
Rec23	30×10	2 011	3 157	0.57	2 936.06	0.46	17 013.06	7.46	5 610.69	1.79	4 122.55	1.05	2 493.64	0.24
Rec25	30×15	2 513	5 328	1.12	4 498.27	0.79	20 606.60	7.20	7 815.43	2.11	6 584.06	1.62	3 342.29	0.33
Rec27	30×15	2 373	4 366	0.84	3 701.88	0.56	18 628.05	6.85	5 885.04	1.48	5 362.98	1.26	3 488.31	0.47
Rec29	30×15	2 287	4 254	0.86	3 499.11	0.53	21 680.76	8.48	7 181.18	2.14	4 985.66	1.18	3 201.80	0.40
Rec31	50×10	3 045	5 633	0.85	4 841.55	0.59	27 465.90	8.02	12 149.55	2.99	7 886.55	1.59	4 019.40	0.32
Rec33	50×10	3 114	3 893	0.25	4 079.34	0.31	19 057.68	5.12	6 010.02	0.93	5 169.24	0.66	3 456.54	0.11
Rec35	50×10	3 277	3 277	0.00	3 277.00	0.00	14 091.10	3.30	3 473.62	0.06	3 309.77	0.01	3 277.00	0.00
Rec37	75×20	4 951	24 260	3.90	20 645.67	3.17	54 807.57	10.07	30 349.63	5.13	21 437.83	3.33	15 447.12	2.12
Rec39	75×20	5 087	19 432	2.82	15 159.26	1.98	48 377.37	8.51	24 214.12	3.76	17 295.80	2.40	14 396.21	1.83
Rec41	75×20	960	4 714	3.91	3 955.20	3.12	10 588.80	10.03	5 932.80	5.18	4 233.60	3.41	3 369.60	2.51
平均值				0.90		0.66		6.51		1.62		1.04		0.46

为了更直观地比较算法性能,图15给出了关于 p-ACGA 和 p-IACGA 两个算法测试部分 Tailard 实例的收敛图。从图15中可以看出,在相同的最大迭代次数下, p-IACGA 的收敛速度比 p-ACGA 更快。在进化初期,算法通过优化初始方法提高初始解的质量,保证进化过程中解的质量;采用节点信息浓度和路径信息度相结合的方

式生成矩阵,提高了区块质量。在进化的后期,应用二元竞赛法来寻找具有更优适应值的解,加快了进化速度。

通过实验数据分析,对于求解 PFSP,本文提出的 p-IACGA 相较于 SGA、ACGA、p-ACGA、BBEDA、LMBBEA 和 p-IACGA 具有较优的求解性能。

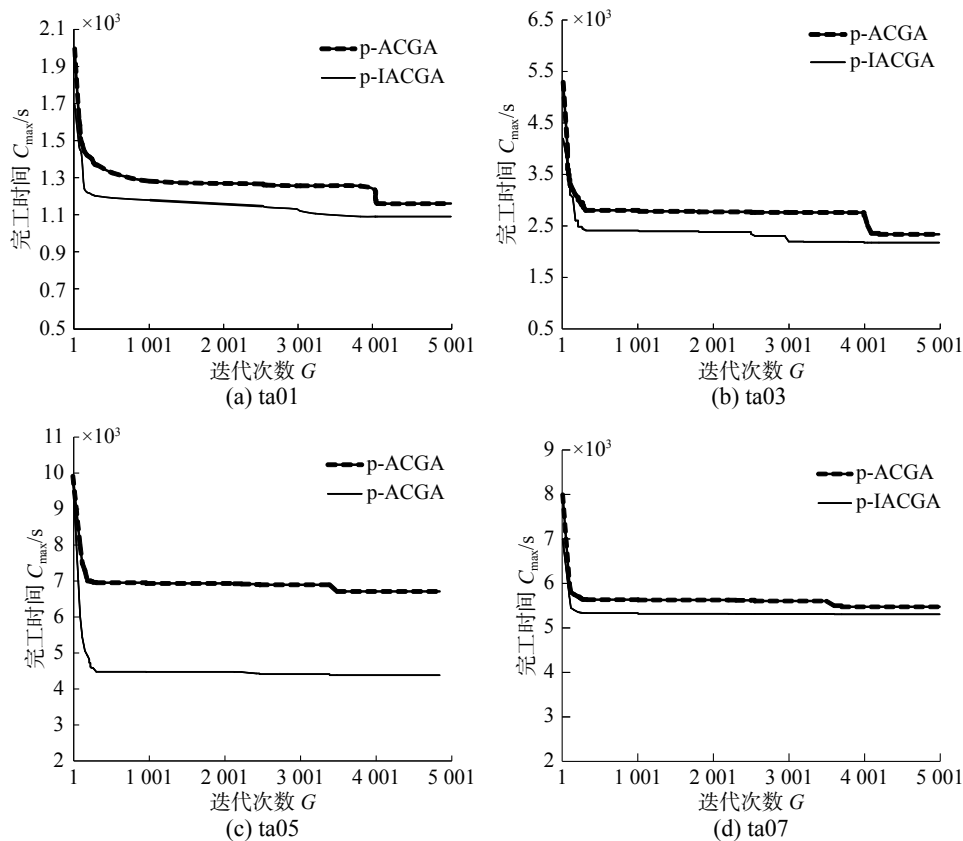


图15 算例收敛图

Fig. 15 Instance convergence diagram

4 结束语

本文针对置换流水车间调度问题,通过对p-ACGA改进,提出了p-IACGA。该算法初始化采用随机机制和改进反向学习机制相结合的方式,提高了初始种群的多样性和质量。进一步考虑了工件与所在解序列的位置关系,对节点信息素浓度进行了分析,通过位置信息素矩阵和相依信息素矩阵挖掘区块加快了收敛速度,提高了最终解的质量。该算法保留了传统GA的交叉、变异等操作,通过若干代的GA进化过程产生精英群体,不断更新区块,提高了人工染色体的质量和更新速度。实验中,通过Taillard和Reeves实例进行测试,以误差率作为比较指标,将结果与其他算法比较,验证了p-IACGA的有效性。

未来研究方向可以将所提出的p-IACGA应用于其他组合优化问题,如旅行商问题、车辆路径问题等。也可以通过改进该算法对PFSP的最小化流程时间、最小化延迟时间等其他一个或多个目标进行研究。

参考文献:

- [1] 周明,孙树栋.遗传算法原理及应用[M].北京:国防工业出版社,1999:6.
- [2] 夏小云,周育人.蚁群优化算法的理论研究进展[J].智能系统学报,2016,11(1):27-36.
XIA Xiaoyun, ZHOU Yuren. Advances in theoretical research of ant colony optimization[J]. CAAI transactions on intelligent systems, 2016, 11(1): 27-36.
- [3] 李金忠,夏洁武,曾小芸,等.多目标模拟退火算法及其应用研究进展[J].计算机工程与科学,2013,35(8):77-88.
LI Jinzhong, XIA Jiewu, ZENG Xiaohui, et al. Survey of multi-objective simulated annealing algorithm and its applications[J]. Computer engineering and science, 2013, 35(8): 77-88.
- [4] TAYEB F B S, BESSEDIK M, BENBOUZID M, et al. Research on permutation flow-shop scheduling problem based on improved genetic immune algorithm with vaccinated offspring[J]. Procedia computer science, 2017, 112: 427-436.
- [5] CHEN Rongchang, CHEN J, CHEN T S, et al. Synergy of genetic algorithm with extensive neighborhood search for the permutation flowshop scheduling problem[J]. Mathematical problems in engineering, 2017, 2017: 3630869.
- [6] BESSEDIK M, TAYEB F B S, CHEURFI H, et al. An immunity-based hybrid genetic algorithms for permutation flowshop scheduling problems[J]. International journal of advanced manufacturing technology, 2016, 85(9/10/11/12): 2459-2469.
- [7] GANGULY S, MUKHERJEE S, BASU D, et al. A novel strategy adaptive genetic algorithm with greedy local search for the permutation flowshop scheduling problem[M]//Swarm, Evolutionary, and Memetic Computing. Berlin, Heidelberg: Springer, 2012: 687-696.
- [8] 齐学梅,王宏涛,陈付龙,等.求解多目标PFSP的改进遗传算法[J].计算机工程与应用,2015,51(11):242-247.
QI Xuemei, WANG Hongtao, CHEN Fulong, et al. Improved genetic algorithm for multi-objective of PFSP[J]. Computer engineering and applications, 2015, 51(11): 242-247.
- [9] 崔琪,吴秀丽,余建军.变邻域改进遗传算法求解混合流水车间调度问题[J].计算机集成制造系统,2017,23(9):1917-1927.
CUI Qi, WU Xiuli, YU Jianjun. Improved genetic algorithm variable neighborhood search for solving hybrid flow shop scheduling problem[J]. Computer integrated manufacturing systems, 2017, 23(9): 1917-1927.
- [10] CHANG P C, HUANG W H, WU J L, et al. A block mining and re-combination enhanced genetic algorithm for the permutation flowshop scheduling problem[J]. International journal of production economics, 2013, 141(1): 45-55.
- [11] JIN Jian. A hybrid discrete biogeography-based optimization for the permutation flow shop scheduling problem[J]. International journal of production research, 2016, 54(16): 4805-4814.
- [12] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE transactions on evolutionary computation, 1997, 1(1): 53-66.
- [13] 陈慧芬.基于链结学习的子群体进化算法求解多目标调度问题[D].天津:天津理工大学,2017.
CHEN Huifen. Sub-population evolutionary algorithm based on linkage learning for multi-objective scheduling problem[D]. Tianjin: Tianjin University of Technology, 2017.
- [14] 裴小兵,陈慧芬,张百栈,等.改善式BVEDA求解多目标调度问题[J].山东大学学报(工学版),2017,47(4):25-30.
PEI Xiaobing, CHEN Huifen, ZHANG Baizhan, et al. Improved bi-variables estimation of distribution algorithms for multi-objective permutation flow shop scheduling problem[J]. Journal of Shandong University (engineering science), 2017, 47(4): 25-30.
- [15] 裴小兵,赵衡.基于二元分布估计算法的置换流水车间调度方法[J].中国机械工程,2017,28(22):2752-2759.

- PEI Xiaobing, ZHAO Heng. Permutation flow shop scheduling problem based on hybrid binary distribution estimation algorithm[J]. *China mechanical engineering*, 2017, 28(22): 2752–2759.
- [16] 张敏, 汪洋, 方侃. 基于改进区块进化算法求解置换流水车间问题[J]. 计算机集成制造系统, 2018, 24(5): 1207–1216.
- ZHANG Min, WANG Yang, FANG Kan. Improved block-based evolutionary algorithm for solving permutation flowshop scheduling problem[J]. *Computer integrated manufacturing systems*, 2018, 24(5): 1207–1216.
- [17] CHANG P C, CHEN Menghui. A block based estimation of distribution algorithm using bivariate model for scheduling problems[J]. *Soft computing*, 2014, 18(6): 1177–1188.
- [18] MILLER B L, GOLDBERG D E. Genetic algorithms, tournament selection, and the effects of noise[J]. *Complex systems*, 1995, 9(3): 193–212.
- [19] CHANG P C, CHEN S H, FAN C Y, et al. Generating artificial chromosomes with probability control in genetic algorithm for machine scheduling problems[J]. *Annals of operations research*, 2010, 180(1): 197–211.
- [20] HSU C Y, CHANG P C, CHEN M H. A linkage mining in block-based evolutionary algorithm for permutation flowshop scheduling problem[J]. *Computers & industrial engineering*, 2015, 83: 159–171.

作者简介:



裴小兵, 男, 1965 生, 教授, 博士, 主要研究方向为生产调度、精益生产。发表学术论文 14 篇。



张春花, 女, 1992 生, 硕士研究生, 主要研究方向为生产调度、智能算法。