

• 计算机工程与应用 •



改进激素算法求解置换流水车间调度问题

郑 堃^{1*}, 练志伟¹, 王玉国¹, 朱长建¹, 顾新艳¹, 刘 轩²

(1. 南京工程学院汽车与轨道交通学院 南京 211167; 2. 中德智能制造研究院 南京 211800)

【摘要】遗传算法中由于激素调节的选择、交叉以及变异算子存在较大目标函数值失调的问题,提出了基于改进激素浓度计算法的自适应遗传算法(IHCCM-IAGA)。IHCCM-IAGA采用基于工件排列的编码方式,并利用反向学习法初始化种群,提高了初始解的质量;针对两点交叉(TPX)算子存在冗余度高、效率低等问题,提出了改进型TPX(ITPX),并引入优良基因库及免疫因子,实现两种交叉方式,同时监控整个进化过程,避免了优质染色体的丢失;设计了多种扰动保持丰富的多样性结构以及相关的局部搜索算法组合成变异算子,建立种群湮灭算子,并设置湮灭因子来引导变异算子中的局部搜索。将IHCCM-IAGA应用于置换流水车间调度问题中,并进行该问题标准算例的各项测试,结果表明IHCCM-IAGA切实有效。

关键词 优良基因库; 激素浓度; 激素调节机制; 改进型TPX; 置换流水车间调度问题; 反向学习法
中图分类号 TP391 **文献标志码** A **doi**:10.12178/1001-0548.2021308

Improved Hormone Algorithm for Solving the Permutation Flow Shop Scheduling Problem

ZHENG Kun^{1*}, LIAN Zhiwei¹, WANG Yugu¹, ZHU Changjian¹, GU Xinyan¹, and LIU Xuan²

(1. School of Automotive and Rail Transit, Nanjing Institute of Technology Nanjing 211167;

2. Sino-German Intelligent Manufacturing Research Institute Nanjing 211800)

Abstract Based on the large objective function value imbalance problem of selection, crossover and mutation in hormone-regulation based genetic algorithm, an adaptive genetic algorithm based on improved hormone concentration calculation method (referred to as IHCCM-IAGA) is proposed. IHCCM-IAGA adopts the coding method based on the arrangement of work pieces, and uses the reverse learning method to initialize the population, which improves the quality of the initial solution. Aiming at the problems of high redundancy and low efficiency in the two-points crossover (TPX) operator, an improved TPX (improved two-points crossover, ITPX) is proposed, and the introduction of excellent gene pool and immune factors realizes two crossover methods and monitors the entire evolution process, avoiding the loss of high-quality chromosomes. A variety of perturbations are designed to maintain a rich diversity structure and related local search algorithms are combined into a mutation operator. A population annihilation operator is established and an annihilation factor is set to guide the local search in the mutation operator. IHCCM-IAGA is applied to the permutation flow shop scheduling problem, and various tests of the standard calculation example of the problem are performed. The results show that IHCCM-IAGA is effective.

Key words excellent gene pool; hormone concentration; hormone regulation mechanism; improved TPX; permutation flow shop scheduling problem; reverse learning method

置换流水车间调度问题(permutation flow shop scheduling problem, PFSP)是流水车间调度中经典的问题之一,也是实际工程应用最广泛的规划问题。目前,精确方法、启发式方法和元启发式方法是求解PFSP常用的3类方法。精确方法不适合求

解复杂度会呈指数增加的调度问题^[1]。启发式方法^[2]可快速给出确定的调度方案,但过分依赖局部调度规则。元启发式方法又称为智能优化算法,是目前求解PFSP的重点研究方法。

智能优化算法又可分为进化算法(evolution

收稿日期:2021-10-22;修回日期:2022-03-15

基金项目:国家自然科学基金(51805244);国家重点研发计划项目(2018YFE011700)

作者简介:郑堃(1984-),博士,副教授,主要从事智能制造、制造系统建模与优化等方面的研究。

*通信作者:郑堃, E-mail: kunzheng@njit.edu.cn

algorithm, EA)、群智能算法 (swarm intelligence, SI) 和局部搜索算法 (local search, LS) 等。EA 通过模仿自然界的进化过程抽象而来, 是 PFSP 的主要处理手段。EA 中应用最广泛与深入的是遗传算法 (genetic algorithm, GA)^[3], 文献 [4-5] 针对 GA 存在的缺点进行改进, 并用现有的 PFSP 标准算例进行验证, 证明其改进的有效性。SI 是受自然界种群的群体智能行为启发, 如共生生物搜索算法 (symbiotic organisms search, SOS)^[6] 是根据自然界的共生关系启发得出的。LS 通常根据 PFSP 的特点设计相应的邻域操作, 并利用该邻域操作对当前局部最优解进行深度挖掘, 如文献 [7] 研究了 3 个基本邻域操作求解 PFSP 的性能, 并与粒子群 (particle swarm optimization, PSO) 算法相结合, 取得了不错的效果。

GA 的深入研究使其在多个领域获得了广泛应用^[8-9]。GA 的操作主要包括选择、交叉以及变异这 3 个基本遗传算子, 其中交叉算子是 GA 区别于其他 EA 的重要特征, 是 GA 产生新个体的主要方法。文献 [10] 首次利用提出的两点交叉 (two-points crossover, TPX) 探索 PFSP 解空间, 取得了不错的效果, 被多次引用^[11-12], 但 TPX 的交叉点为随机选取, 所以存在冗余度高和算法中后期效率低等缺陷。文献 [4] 针对 GA 本身不具备记忆功能而导致优质解易丢失、收敛速度慢和最终解的质量低等缺点, 提出保优机制和选择策略, 按一定保优比例, 将优质解遗传到下一代种群中。文献 [5] 利用限优策略保持种群多样性, 解决 GA 陷入局部等问题。因此有学者提出混合算法, 即将智能算法中的各类算法进行混合, 作为适用域拓展和算法性能提高的重要手段, 如文献 [13] 将 SOS 与 LS 结合, 提出了混合共生生物搜索算法 (hybrid symbiotic organisms search, HSOS), 采用标准算例集证明其优越性和稳定性, 在优化过程中可同时满足探索 (exploitation) 和开发 (exploration)、集中性 (intensification) 和广泛性 (diversification), 但会增加算法的计算复杂度和搜索时间。

近年来, 学者们进一步提出了生物学和生产制造系统相结合的方法。文献 [14-15] 在这方面研究颇多。利用生物体的激素调节对体内各种激素的调控具有较好的自适应性和稳定性等优点, 文献 [16] 提出了基于激素调节规律的改进型自适应遗传算法 (improved adaptive genetic algorithm, IAGA), 并结合作业车间调度问题进行了验证; 文献 [17-18]

将激素调节规律应用至 PSO 的速度更新公式和惯性因子来求解 PFSP 和混合流水车间调度问题。目前基于激素调节的算法已被证实能有效克服种群进化缓慢、个体早熟等缺陷问题, 但受激素浓度的影响, 一般激素浓度的计算方法使激素调节规律在目标函数较大时, 出现了失调现象, 因而无法保证基于激素调节的算法能具有更好的优化结果。

综上所述, 在采用 GA 求解 PFSP 问题时, 为解决常用选择算子中 TPX 的缺陷、全局最优解收敛困难问题以及 IAGA 的局限性, 本文提出了基于改进激素浓度计算法的 IAGA(improved hormone concentration calculation method-iaGA, IHCCM-IAGA)。该算法主要将激素调节机制引入到选择算子中, 通过改进激素浓度的计算方法, 使基于激素调节规律的参数在目标函数值较大时仍有较好的灵敏度; 采用了精确取点的 ITPX 算子进行交叉, 提高算法的探索能力; 通过建立优良基因库及免疫因子, 保存并监控每代进化过程的优质解, 同时将多种扰动操作和局部搜索算法进行组合, 设置湮灭因子并利用免疫因子映射的影子标记优质染色体来共同引导局部搜索, 提高算法的局部开发特性。

1 置换流水车间调度问题描述

PFSP 描述为: 有 n 个相互独立工件 $\{J_1, J_2, \dots, J_n\}$, 使用 m 台加工机器 $\{M_1, M_2, \dots, M_m\}$, 每个工件在每台机器上的加工时间都不相同。调度目标为优化工件的加工顺序使得最后的完工时间最小, 此过程需满足如下约束。

- 1) 初始时刻, 所有工件和加工机器准备就绪;
- 2) 机器之间的顺序是固定的, 所有工件需以相同的顺序经过每台机器;
- 3) 任意时刻一台机器只可加工一个工件;
- 4) 机器一旦开始加工, 则不能中断;
- 5) 工件之间存在先后排序。

PFSP 的数学模型描述如下: 假设 $p(i, j)$ 表示工件 j 在机器 i 上的加工时间, 且工件之间的排序为集合 $O(O_1, O_2, \dots, O_n)$, 工件在机器上的加工顺序为机器 1~机器 m , 则 n 个工件在 m 台机器上的完工时间 $C(i, O_j)$ 分别为:

$$C(1, O_1) = p(1, O_1) \quad (1)$$

$$C(1, O_j) = C(1, O_{j-1}) + p(1, O_j) \quad j = 2, 3, \dots, n \quad (2)$$

$$C(i, O_1) = C(i-1, O_1) + p(i, O_1) \quad i = 2, 3, \dots, m \quad (3)$$

$$C(1, O_j) = \max\{C(i, O_{j-1}), C(i-1, O_j) + p(i, O_j)\} \\ i = 2, 3, \dots, m \quad j = 2, 3, \dots, n \quad (4)$$

$$C_{\max} = C(m, O_n) \quad (5)$$

本文求解 PFSP 就是要获得该工件集的最优排序 O^* , 使其他任何工件排序 O 都有:

$$C_{\max}(O^*) \leq C_{\max}(O) \quad (6)$$

因此, 求解 PFSP 的目标函数为 $\min\{C_{\max}(O)\}$ 。

2 IHCCM 在 IAGA 中的应用

2.1 IHCCM-IAGA 流程

IHCCM-IAGA 流程如图 1 所示, 该算法能很好地弥补传统 GA 的不足, 使得改进算法在求解的性能上拥有更大优势。

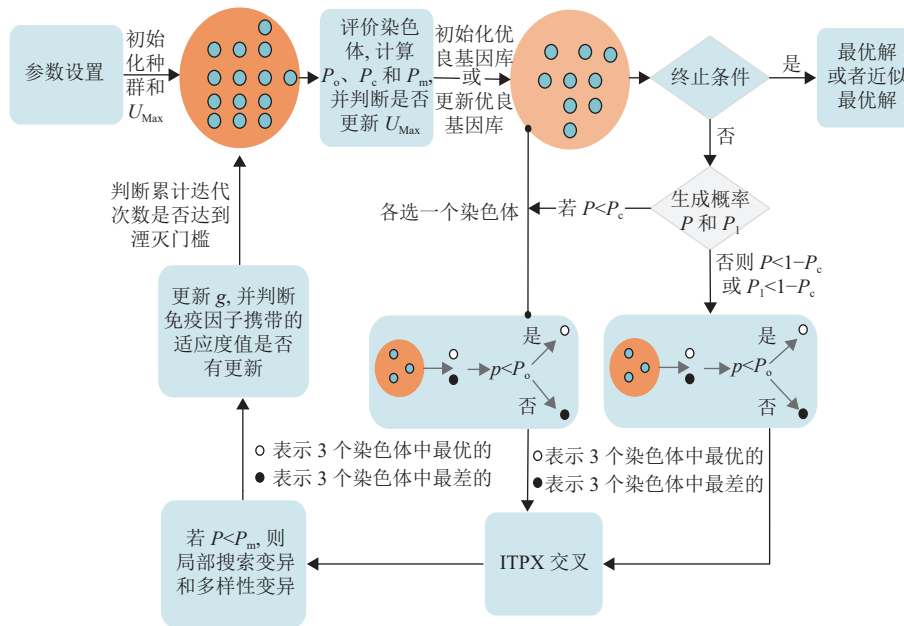


图 1 IHCCM-IAGA 流程框图

由图中可知, IHCCM-IAGA 算法较普通 GA 流程的主要差别为激素的应用、优良基因库和免疫因子的引入、两种交叉方式的实现、变异算子与局部搜索的融合以及种群湮灭算子和湮灭因子的建立。其具体执行步骤如下。

1) 确定参数。包括种群 N 和基因库规模 NP 、迭代次数 G 、种群湮灭因子 A_f 、自适应选择门槛 P_0 、交叉概率 P_c 和变异概率 P_m 等一些参数;

2) 利用反向学习法生成初始种群法和初始化 U_{\max} ;

3) 评价种群中每个染色体的适应度值, 计算自适应选择门槛 P_0 、交叉概率 P_c 和变异概率 P_m , 并判断是否更新 U_{\max} , 同时根据要求更新基因库或初始化基因库; 如果达到最大迭代次数则输出最优解或近似最优解, 其中输出解为免疫因子携带的染色体和其适应度值, 结束运行; 否则执行步骤 4);

4) 生成随机概率 P 或 P_1 , 若 $P < P_c$, 则在种群和优良基因库各选一个染色体进行 ITPX 交叉, 否则, $P < 1 - P_c$ 或 $P_1 < 1 - P_c$ 时, 在种群选出两个染色

体进行 ITPX 交叉, 其中如果从种群选取染色体, 则按选择操作选取个体, 否则随机选取;

5) 若随机概率 $P < P_c$, 则按变异规则进行局部搜索变异和多样性变异, 产生下一代种群;

6) 更新当前迭代次数 g , 并检查免疫因子携带的适应度值是否有改变, 若有改变, 则不对当前种群做任何变动, 且重置累计迭代次数的计算; 否则更新累计迭代次数, 并检测累计迭代次数是否达到湮灭门槛, 若达到湮灭门槛, 则对当前种群实行湮灭和更新湮灭因子 A_f , 同时重置累计迭代次数的计算;

7) 返回步骤 3)。

2.2 IHCCM-IAGA 原理

2.2.1 基于激素调节的 IAGA 参数设计

通常种群进化过程主要分为两个阶段: 1) 种群进化前期: 交叉概率大、变异概率小, 有利于种群的快速收敛, 较优解不易丢失的特点; 2) 种群进化后期: 交叉概率小、变异概率大, 有利于精细化搜索, 种群多样性的保持。但传统 GA 的交叉概率和

变异概率固定, 且参数选择困难, 而这极大影响 GA 的优化结果。因此, 借鉴内分泌激素调节规律的上升和下降函数, 文献 [16,19] 提出了基于激素调节的改进型自适应遗传算法 (IAGA)。

同时种群多样性还与选择算子息息相关, 因此, 有必要设置一个由激素调节的自适应选择门槛。

内分泌激素的基本规律由文献 [20] 提出, 它揭示激素具有单调性和非负性的变化特性, 激素调节的上升规律函数 $F_{up}(X)$ 和下降规律函数 $F_{down}(X)$ 遵循 Hill 函数规律, 其公式如下:

$$F_{up}(X) = \frac{X^n}{D^n + X^n} \quad (7)$$

$$F_{down}(X) = \frac{D^n}{D^n + X^n} \quad (8)$$

式中, X 为函数自变量; D 为阈值, 且 $D > 0$; n 为 Hill 系数, 且 $n \geq 1$; n 和 D 共同决定上升下降曲线斜率。

图 2 为参数 $D=1$, $n=1,2,3$ 情况下的 Hill 变化曲线。当参数 D 不变、 n 值由小到大时, Hill 曲线收敛的速度越来越快, 到达稳定状态所需时间也越来越少。

如果激素 x 受激素 y 调控, 则激素 x 的分泌速度 S_x 与激素 y 的浓度 C_y 的关系为:

$$S_x = \alpha F(C_y) + S_{x0} \quad (9)$$

式中, S_{x0} 为激素 x 的基础分泌速率; α 为常量系数。

将式 (7) 和式 (8) 代入式 (9), 得:

$$V_{up} = a \frac{C_y^n}{D^n + C_y^n} + S_{x0} = S_{x0} \left(1 + \frac{a}{S_{x0}} \times \frac{C_y^n}{D^n + C_y^n} \right) \quad (10)$$

$$V_{down} = a \frac{D^n}{D^n + C_y^n} + S_{x0} = S_{x0} \left(1 - \frac{a}{S_{x0}} \times \frac{C_y^n}{D^n + C_y^n} \right) + a \quad (11)$$

模仿式 (10) 和式 (11), 设计了与种群优劣程度相关的自适应选择门槛、交叉概率和变异概率因子函数, 具体如下。

1) 自适应交叉概率^[16]: 由式 (11) 设计出自适应交叉概率为:

$$P_c = 1 - P_c^{init} \left(1 + a \frac{(f_{av})^{n_c}}{(f_{max} - f_{min})^{n_c} + (f_{av})^{n_c}} \right) \quad (12)$$

式中, P_c^{init} 为初始交叉概率; P_c 为自适应交叉概率; f_{max} 为每一代个体适应度最大值; f_{min} 代表适应度最小值; f_{av} 为适应度平均值; α 、 n_c 为调节

系数。 P_c 随着 f_{av} 增大而减小; 随着 f_{av} 的减小而增大。

2) 自适应变异概率^[16]: 根据式 (10), 设计自适应变异概率为:

$$P_m = P_m^{init} \left(1 + \beta \frac{(f_{av})^{n_m}}{(f_{max} - f_{min})^{n_m} + (f_{av})^{n_m}} \right) \quad (13)$$

式中, P_m^{init} 为初始变异概率; P_m 为自适应变异概率; β 、 n_m 为调节系数。

3) 自适应选择门槛: 参照式 (11) 对锦标赛选择设置一个由激素调节的门槛, 有:

$$P_o = 1 - P_o^{init} \left(1 + \gamma \frac{(f_{av})^{n_o}}{(f_{max} - f_{min})^{n_o} + (f_{av})^{n_o}} \right) \quad (14)$$

式中, P_o^{init} 表示初始选择门槛; P_o 表示自适应选择门槛; γ 、 n_o 是调节系数。

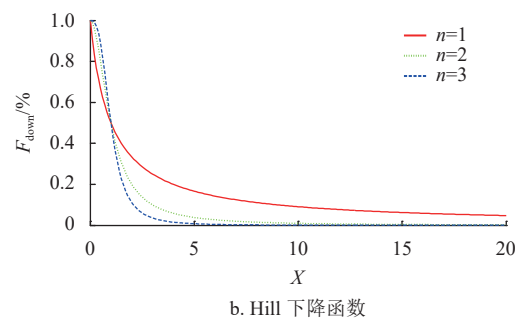
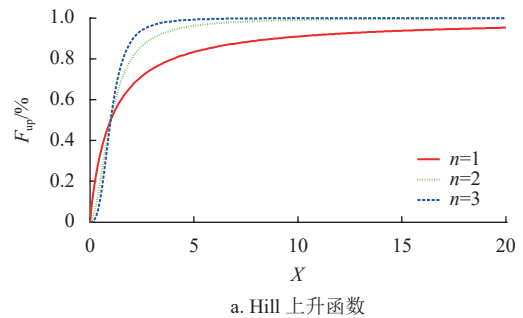


图 2 参数 n 对 Hill 曲线的影响

2.2.2 参数的选择范围讨论

1) α 、 β 和 γ

α 、 β 和 γ 的取值将影响自适应交叉概率、变异概率和选择门槛的增幅和减幅, 其取值范围由式 (12)~式 (14) 确定, 可得:

$$0 < \alpha \leq \frac{1}{P_c^{init}} - 1 \quad (15)$$

$$0 < \beta \leq \frac{1}{P_m^{init}} - 1 \quad (16)$$

$$0 < \gamma \leq \frac{1}{P_o^{init}} - 1 \quad (17)$$

2) n_c 、 n_m 和 n_o

同样, 参数 n_c 、 n_m 和 n_o 的取值会对自适应交叉概率、变异概率和选择门槛的进化速度产生影响。一般地, 取值范围为 1~4 之间的整数。

2.2.3 内分泌激素浓度计算法的改进

激素调节规律与激素的浓度有关, 而 GA 中的激素浓度由 f_{av} 决定, 一般 f_{av} 的适应度值主要由目标函数值取倒数 ($1/\text{Makespan}$) 获得, 从而使得激素浓度总是能够与 Hill 上升和下降函数呈正相关和负相关的联系, 即适应度值越大, 适应能力就越强。所以在 Makespan 较小时, 对激素调节有很好的引导作用, 但适应度由目标函数取倒数不仅会有参数的影响, 且随着 Makespan 越来越大, 其适应度值会以降低数量级的形式表达, 因此个体适应度值之间的变化量就越小, 此时受激素浓度影响的激素调节规律将会出现调节不明显、失调等现象, 即选择门槛、交叉概率和变异概率不再严格遵循 Hill 下降和上升函数的规律。为此, 本文将对计算激素浓度的方法进行改进, 以解决激素规律失调的问题。

对激素公式中适应度值的计算方法改进如下:

1) 评价初始种群中每个染色体, 记其中最差染色体的最大适应度值 $\max\{C_{\max}(O)\}$ 为 $U_{\max} = \max\{C_{\max}(O)\} + 50$;

2) 以 U_{\max} 为标准, 对种群中所有染色体计算标准化适应度值, 标准化公式为 $f_{Si} = U_{\max} - f_i$, $i = 1, 2, \dots, N$, 为种群中每个染色体离目前已知最差染色体的距离; 染色体越好, 离最差染色体就越远, 其 f_{Si} 越大, 相反, 离最差染色体就越近, 其 f_{Si} 就越小; 可见, 改进的计算方法可以与激素的调节规律和现实意义成正比, 具有可行性。

3) 用标准化适应度值计算激素公式中的 f_{av} 、 f_{\min} 和 f_{\max} , 在种群每次迭代后计算标准适应度值时, 如有 $f_{Si} \leq 0$, 则在计算完 f_{av} 、 f_{\min} 和 f_{\max} 后, 更新当前 U_{\max} , 更新方法为计算当代种群中最差染色体的最大适应度值 $\max\{C_{\max}(O)\}$ 为 $U_{\max} = \max\{C_{\max}(O)\} + 50$;

激素调节规律除受激素浓度 f_{av} 调控外, 还受激素震荡的影响, 激素震荡由 f_{\max} 与 f_{\min} 的差值决定。当差值过大时, 激素规律则认为种群染色体分布趋于两极化, 由激素震荡成为主控因子, 调节选择门槛、交叉概率快速增大和变异概率快速减小, 以确保种群染色体分布均匀, 保持种群的稳定。相反, 当差值过小时, 激素规律则认为种群染色体分布过于集中、多样性过少, 此时激素震荡仍是主控

因子, 调节选择门槛、交叉概率快速减小和变异概率快速增大, 以确保种群染色体分布均匀、增加多样性; 而差值过大或过小, 使得激素震荡成为主控因子, 无法正确反映种群的实际情况, 因此收敛速度不稳定、解的质量也难以保证。为此, 只有在差值合适时, 激素浓度才是主控因子, 而一般激素浓度的计算方法可以做到此点, 且受激素震荡的微调作用。为此, 本文对标准化的 f_{\min} 改为 $f_{\min} = f_{av}/2 + f_{\min}$, 使改进后的激素浓度计算法与一般激素浓度计算法具有相同的调节作用。

3 IHCCM-IAGA 求解 PFSP

3.1 初始化种群

3.1.1 染色体的编码

用集合 $O = \{O_1, O_2, \dots, O_n\}$ 表示工件在机器 1~机器 m 上的加工顺序, 集合 O 中的元素用工件的整数编号表示, 用集合 $J = \{J_1, J_2, \dots, J_n\}$ 表示工件集, 从而构造出基于工件编号的整数编码, 例如 7 个工件的集合 $O = \{1, 3, 4, 2, 6, 5, 7\}$ 所对应工件的加工顺序为 $J_1 \rightarrow J_3 \rightarrow J_4 \rightarrow J_2 \rightarrow J_6 \rightarrow J_5 \rightarrow J_7$ 。

3.1.2 初始化方法

本文将采用文献 [21] 提出的反向学习法 (opposition-based learning, OBL), 设工件编号的最小数为 1, 最大数为 n , 当前工序的工件编号为 O_i , 则 O_i 的反向学习 \bar{O}_i 的编号如为:

$$\bar{O}_i = 1 + n - O_i \quad (18)$$

因此本文的初始化过程如下。

1) 设置初始种群个体数为 N , 随机初始化 N 个染色体, 生成初始种群 $P1$; 然后对种群 $P1$ 中的所有染色体进行如图 3 所示的反向学习, 生成初始种群 $P2$;



图 3 染色体的反向学习

2) 将初始种群 $P1$ 和 $P2$ 合并, 计算合并后所有染色体的目标值并从小到大进行排列, 选取前 N 个染色体作为最后的初始种群。

3.2 适应度函数

根据每个染色体的集合 O 编码计算其目标函数值 Makespan, 并作为其适应度值, 即 $\text{fitness} = \min\{C_{\max}(O)\}$ 。

3.3 选择操作

本文采用文献 [22] 提出的锦标赛选择方法,

并加以改进, 确保在种群进化后期能够更好的保持种群多样性。

每次从种群中选择若干个体进行适应度比较, 选出最差的个体和最优的个体, 若随机值 (在 $0\sim 1$ 之间随机产生) 小于阈值 (受激素调节的值), 则选择最优的一个, 否则就选择最差的一个。这样可保证在种群进化前期, 高性能个体能最大限度的遗传至下一代, 以提高种群的全局收敛速度; 在种群进化后期, 提高较差个体进入下一代种群的概率, 以确保种群多样性, 从而更好地克服个体早熟和局部收敛等现象。

3.4 交叉操作

3.4.1 ITPX 算子

对基于工件编号整数排列的编码方式, 一般采用 TPX^[10] 来产生子代, 如图 4 所示, 此方法容易出现冗余、效率低等问题, 无法高效地对解空间进行搜索, 因此最后解的质量得不到保障。

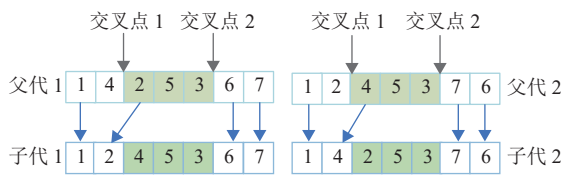


图 4 TPX 交叉算子

本文将在此交叉算子的基础上进行改进, 提出一种精确的取点方式 ITPX, 具体步骤如下。

1) 对两个父代染色体从左向右依次等基因位对比, 标注等基因位值不同的基因位, 记作集合 $T=\{T_1, T_2, \dots, T_s\}$, 其中 T_1 为两父代染色体第一不同的基因位, T_s 为两父代染色体最后不同的基因位, T_1 至 T_s 的基因位距离称为最大不同基因位区间;

2) 设置两个交叉点分别为 n_1 、 n_2 , 且 $n_2 > n_1$ 。 n_1 和 n_2 分两次从集合 T 中选取基因位作为其交叉点。当 n_1 从集合 T 选取交叉点时, n_1 第一个交叉点必须从染色体的第一个基因位开始选取 (不管染色体第一基因位是否为不同基因位), 记为 a_0 ; n_1 其余交叉点从集合 T 中按 $1\sim s$ 的顺序选取, 记为 a_1 , 同时若从集合中选取的第一个基因位 T_1 为染色体第一基因位, 则按顺序重新选取, 直至集合 T 选取完毕。 n_2 则从左向右依次选取交叉点, n_2 开始交叉点 b_0 选取规则为: 若 $n_1=a_0$, 且 $a_0 \neq T_1$, 则 n_2 开始交叉点为 T_1 , 若 $a_0=T_1$, 则 n_2 开始交叉点为 a_0+1 ; 若 n_1 选取的交叉点为 a_1 , 则 n_2 开始交叉点为 a_1+1 ; n_2 结束交叉点均为 T_s-2 。每当 n_1 选取一个交叉点, n_2 则从 b_0 开始, 依次加

1, 直至结束点 T_s-2 ; n_1 只能从集合 T 中选取 T_s-2 之前的点;

3) 当 n_2 从集合 T 选取交叉点时, n_2 第一个交叉点必须从染色体的最后一个基因位开始选取, 记为 a_{n-1} ; n_2 其余交叉点从集合 T 中按 $1\sim s$ 的顺序选取, 记为 a_2 , 直至集合 T 选取完毕, 其中如果从集合中选取的最后基因位 T_s 为染色体最后基因位, 则提前结束选取。 n_1 则从右向左依次选取交叉点, n_1 开始交叉点 b_1 选取规则为: 若 $n_2=a_{n-1}$, 且 $a_{n-1} \neq T_s$, 则 n_1 开始交叉点为 T_s , 若 $a_{n-1}=T_s$, 则 n_1 开始交叉点为 $a_{n-1}-1$; 若 n_2 选取的交叉点为 a_2 , 则 n_1 开始交叉点为 a_2-1 ; n_1 结束交叉点均为 T_1+2 。每当 n_2 选取一个交叉点, n_1 则从 b_1 开始, 依次减 1, 直至结束点 T_1+2 。以上过程, 如图 5 所示。值得注意的是, n_2 只能从集合 T 中选取 T_1+2 之后的点。

通过 ITPX 方法生成两父代所有子代, 从各自子代集中选取最优的染色体作为下一代, 且免疫因子实时监测两父代产生的子代集, 以保证优质解的不丢失。同时本文引入优良基因库, 并采用两个交叉概率来实现两种交叉方式, 一种是分别从优良基因库随机选择染色体, 按选择操作从种群中选择染色体以 P_c 概率进行交叉; 另一种是两个染色体均按选择操作从种群中选取以 $1-P_c$ 的概率进行交叉。

3.4.2 优良基因库

优良基因库主要分为以下几个部分。

1) 免疫因子: 免疫因子主要监控每代种群在交叉算子中所有的子代染色体和变异算子中局部搜索部分, 实时更新优于免疫因子所携带的染色体;

2) 免疫因子映射的影子: 影子只映射免疫因子携带的值, 利用影子标记基因库中适应度值相同, 但两两个体 H 距离均不同的染色体 (两个染色体中不同基因位的个数称为海明距离, 简称 H 距离), 并对这些染色体从零开始标号, 记最大标号为 M , 利用映射的影子标记优质染色体来引导下一代种群的局部搜索;

3) 优良基因库初始化: 设优良基因库最大容量为 N_p , 初始化时, 从初始化种群引入适应度由低到高排序的前 $N_p/2$ 染色体到基因库中, 剩余的容量在种群进化时按更新策略填充引入; 同时对免疫因子初始化, 记录基因库中适应度最优的染色体;

4) 优良基因库更新策略: 在每代种群进化完成时, 对基因库更新, 更新策略分为两种情况: ①当优良基因库有容量时, 评价当代种群中每个染色

体，并与当前基因库中最差染色体比较，如果小于，则检查当前染色体是否有与基因库中染色体适应度值相同，如有相同，则计算两个染色体 H 距

离，若 $H=0$ ，则不引入，否则引入填充；②当优良基因库没有容量时，利用替换方法实现更新策略，并重新标记当前基因库中最差染色体及其适应度值。

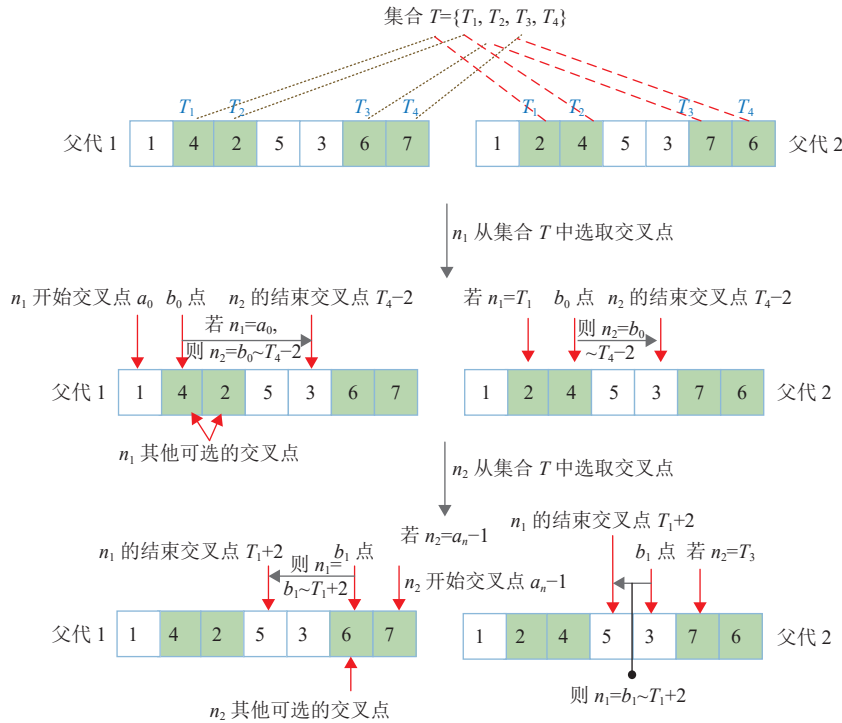


图 5 ITPX 算子的精确交叉取点方式

3.4.3 种群湮灭

为了更好地开发空间多样性和探索解空间，利用优良基因库中的免疫因子对种群湮灭是否触发进行判断，其判断过程为：当种群连续累计迭代一定次数后，免疫因子仍不更新，则触发；当免疫因子有更新时，则重新累计迭代次数。同时，设置湮灭因子 A_f ，种群每湮灭一次，湮灭因子则从初始值依次叠加。湮灭因子主要用来影响变异算子的局部搜索，旨在增强变异的局部搜索特性。

本文对种群 80% 个体实行湮灭，湮灭方式为将种群前 40% 的个体实行反向学习，并依次随机打乱生成新染色体，后 40% 的个体利用随机初始化法生成新染色体。基因库只保留由免疫因子映射影子标记的优质染色体，保留数目不得超过整个基因库最大容量的 20%，其余优质染色体实行删除操作，由种群湮灭后的染色体以适应度由低到高排序依次填充基因库，其中免疫因子等不参与湮灭。

3.5 变异操作

本文将具有丰富结构的多种扰动操作和局部搜索算法进行组合，构成组合变异算子，相比在选择-

交叉-变异等操作之后再引入局部搜索算法，组合变异算子能更好地维持种群的多样性。

变异过程如图 6 所示，主要步骤为：满足变异概率后，变异开始：生成随机概率 P_1 ，若 $P_1 < P_m$ ，则开始局部搜索，否则进行多样性变异；其中当父代染色体的适应度值等于免疫因子映射影子的值 ($f_c = f_{if}$)，则实施局部搜索 1，否则从局部搜索 2 开始，若搜索过程当中有新染色体优于免疫因子，则替换后，触发湮灭因子重置准则和直接跳转至多样性变异，即 $f_n < f_{if}$ ；因此图中的“是”表示有更加优质的新染色体，“否”表示无。具体实现如下：扰动操作主要包括①插入：随机选择染色体的两个不同基因位，将其中一个基因位插入到另一个基因位的前面；②反转逆序：随机选择染色体的两个不同基因位，对两个不同基因之间的所有基因反转逆序；③打乱互换：随机选择染色体中一些不同的基因位，然后随机打乱，重新排序；④两点互换：随机选择染色体的两个不同基因位，将这两个不同基因互换。

局部搜索主要分成 3 部分。

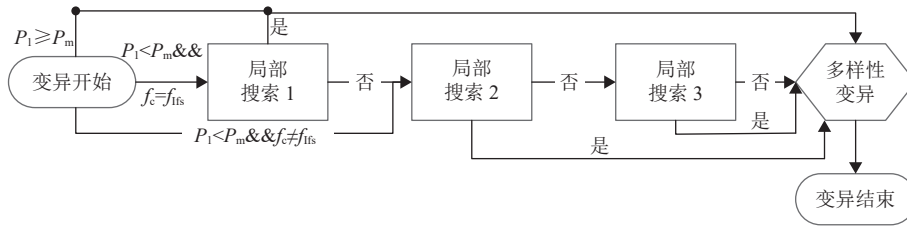


图6 变异过程

1) 首先利用扰动操作对此父代染色体局部变换, 共变换 $5A_f$ 次, 生成 $A_f \times 5$ 个子代染色体, 扰动选择过程为: 设置随机整数 r , 范围为 $0 \sim 5$, 当 $r=0$ 时, 采用①; 当 $r=1$ 时, 采用②; 当 $r=2$ 或 3 或 4 时, 采用③; 当 $r=5$ 时, 采用④。然后评价所有子代染色体, 判断 $f_n < f_{if}$; 否则从父代染色体开始以 $1-P_c$ 的概率进入循环搜索, 如图7所示。循环搜索分为大循环和小循环, 大循环次数为 A_f , 小循环次数为 $A_f \times 5$, 衰减系数 $Q=0.12/A_f \cdot (f_c - f_{if}) / (f_{if} \times 0.0624 \times A_f^2)$; 大循环均从零递增计数, 小循环从准许的最大次数递减计数, 满足循环次数或 $f_n < f_{if}$ 时结束, 同时记当前染色体适应度值为 f_1 和 f_2 , 染色体为 C_1 和 C_2 ; 记当前大循环的次数为 T , 则衰减值为 $1+TQ$, 记当前小循环的次数为 t 和

对染色体 C_1 的更新次数为 $Count$, 每次大循环时, 初始化 $Count=0$, 则小循环过程为: 先利用扰动①②④生成新的染色体, 根据文献[7]的分析, 设随机概率 $P < 0.15$ 时, 选择②, 随机概率 $P < 0.4$ 或 $P_1 < 0.4$ 时, 则选择①, 若上述概率都不满足, 则选择④, 新染色体判断准则和处理分类如下: **I.** 判断 $f_n < f_{if}$; **II.** 若 $f_n < f_1 \& f_n < f_2$, 则接受 f_n , 并更新 f_1 、 f_2 、 C_1 、 C_2 和 $Count$; **III.** 若 $f_n > f_1 \& f_n < f_2$, 也接受 f_n , 但只更新 f_2 和 C_2 ; **IV.** 若 $f_n > f_1 \& f_n > f_2$, 则计算接受概率 $P_a = (t \times \text{衰减值}) / (A_f \times 10)$, 若随机概率 $P \leq P_a$, 则接受 f_n , 并更新 f_2 和 C_2 , 否则不接受 f_n , 利用 C_2 对当前染色体重置, 若随机概率 $P > P_a$ 且 $Count=0$, 则利用 C_1 和 f_1 对当前染色体和 f_2 重置。

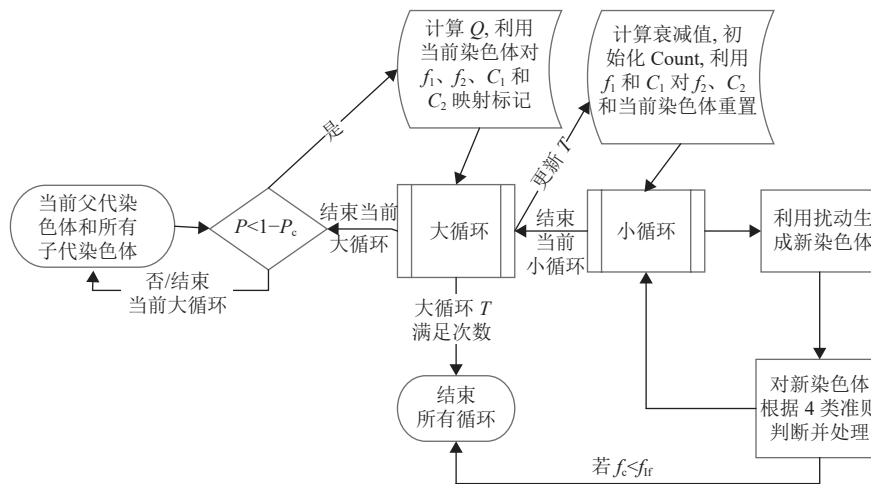


图7 循环搜索

每进入下一次小循环时, 则利用 C_1 和 f_1 对 C_2 、 f_2 和当前染色体重置, 直至当前大循环结束, 输出 C_1 和 f_1 , 同时若当前搜索的染色体为子代染色体且输出的 $f_1 \neq f_{if}$, 则利用 C_1 和 f_1 对当前子代染色体更新。若父代染色体和所有子代染色体循环搜索完毕, 且没有 $f_n < f_{if}$, 则随机概率 $P < 1 - P_c$, 将从子代染色体中随机选取一个替换当前染色体, 否则选择子代染色体中最优的个体替换, 并进入第二部

分局部搜索。

2) 利用基因库被标记的随机染色体对当前父代染色体进行比对, 保留相同或不同的基因, 对不同或相同的基因全排列搜索或扰动③搜索, 生成所有子代染色体, 然后评价所有子代染色体, 记录其中适应度最小的染色体 f_{min} 、适应度最大的染色体 f_{max} 和随机一条染色体 f_{rand} , 并判断 $f_n < f_{if}$; 否则以 P_c 概率接受 f_{min} 染色体, 如不满足上述概率, 则以

$1-P_c$ 概率接受 f_{rand} 染色体, 若不满足上述所有概率, 则接受 f_{max} 染色体, 且进行第三部分的局部搜索。比对选择方式: 若当前染色体经过第一部分局部搜索, 则随机概率 $P < P_0$ 时, 保留相同的基因, 对不同的基因搜索, 否则保留不同基因, 对相同的基因搜索; 若当前染色未经过第一部分局部搜索, 则随机概率 $P < P_0$ 时, 保留不同基因, 对相同的基因搜索, 否则保留相同的基因, 对不同的基因搜索; 如搜索的基因数超过 5 个, 则利用扰动③生成 120 个子代染色体, 否则全排列生成所有子代染色体。

3) 对当前染色体进行循环搜索, 大循环次数为随机整数 R , 范围 $1 \sim A_f$, 小循环次数为 30, 衰减系数 $Q = R_1 - (f_c - f_{\text{if}}) / (f_{\text{if}} \times 0.285 \times R)$, 其中 R_1 为 $0.004 \sim 0.013$ 的随机数。若记当前大循环的次数为 T , 则衰减值仍为 $1 + TQ$, 其循环过程与上述的循环过程一致。

多样性变异: 主要采用扰动①②③作为种群多样性保持的策略, 设置随机整数 r , 范围为 $0 \sim 2$, 当 $r=0$ 时, 采用扰动①; 当 $r=1$ 时, 采用扰动②; 当 $r=2$ 时, 采用扰动③; 不同的扰动操作可以保持丰富的多样性结构。

4 试验仿真与结果分析

为了验证本文所提算法的性能, 进行以下两个比较实验。

1) 改进激素浓度计算法 (IHCCM) 与普通激素浓度计算法的对比实验。通过 Rec^[23] 算例测试比较改进激素浓度计算法与普通激素浓度计算法的求解性能, 为客观评价改进激素浓度计算法对算法整体性能的提升提供依据。

2) IHCCM-IAGA 对比实验。通过国际标准算例集 Carlier^[24]、Rec^[23]、Taillard^[25] 测试比较 IHCCM-IAGA 与其他算法的求解性能, 为客观评价 IHCCM-IAGA 和 ITPX 的性能提供依据。同时以上所有比较实验最终输出解均为免疫因子信息。

改进的遗传算法采用 DEV C++ 编程, 程序在环境为 Intel(R) Core(TM), 主频 2.3 GHZ, 内存 8 GB 的个人计算机上运行。经过多次试验比较, 设定如表 1 所示的实验参数。同时为了节省优化时间和提高算法性能, ITPX 交叉点设定如下选取规则: 若集合 T 的个数为小于等于 4, 全部选取; 若集合 T 的个数大于 4, 但小于等于 10, 则记集合 T 的个数为 Num, 选取个数 SN 为 $5 \sim \text{Num}$, 选取点为从

集合 T 中随机选取 SN 个不同的点; 若集合的个数大于 10, 则选取个数 SN 为 $5 \sim 10$, 选取点为从集合 T 中随机选取 SN 个不同的点。其中 n_1 和 n_2 选取的交叉点一致, 针对每个算例分别连续运行 20 次, 同时使用最佳相对误差 (best relative error, BRE)、平均相对误差 (average relative error, ARE) 和最差相对误差 (worst relative error, WRE) 评价本文算法结果, 其计算公式分别如下:

$$\text{BRE} = (C_{\text{best}} - C^*) / C^* \times 100\% \quad (19)$$

$$\text{ARE} = (C_{\text{avg}} - C^*) / C^* \times 100\% \quad (20)$$

$$\text{WRE} = (C_{\text{worst}} - C^*) / C^* \times 100\% \quad (21)$$

式中, C^* 为目前该问题已知的最优值; C_{best} 为算法在规定次数内求解该算例的最优值; C_{avg} 为算法在规定次数内求解该算例的平均值; C_{worst} 为算法在规定次数内求该算例的最差值。

表 1 实验参数

参数名称	参数值		
自适应选择门槛	$P_o^{\text{init}} = 0.07$	$\gamma = 7.535$	$n_o = 3$
自适应交叉概率	$P_c^{\text{init}} = 0.1$	$\alpha = 7.535$	$n_c = 3$
自适应变异概率	$P_m^{\text{init}} = 0.01$	$\beta = 40.345$	$n_m = 3$
种群规模	$N = 60$		
优良基因库规模	$N_p = 30$		
迭代次数	若机器数与工件数的乘积 $m \times n \leq 3\ 000$, 则 $G = m \times n$; 否则 $G = 3\ 000$		
湮灭门槛	迭代最优解不变次数限制为 $m \times 4$ 次 (不超过)		
湮灭因子迭代准则	初始 $A_f = 1$, 每湮灭一次加 2; 若 $g < G/2$ 且 $A_f > 9$, 则 $A_f = 1$; 若 $g \geq G/2$ 且 $A_f > 5$, 则 $A_f = 3$		
湮灭因子重置准则	若 $g < G/2$, 则 $A_f = 1$; 若 $g \geq G/2$, 则 $A_f = 3$		

4.1 IHCCM 对比实验

激素浓度测试算法框架不包括种群湮灭和局部搜索变异, 改进激素浓度计算法与普通激素浓度计算法下自适应参数均采用如表 1 所示的实验参数, 为了节省运行时间, 迭代次数 G 改为 100。因变异除了可保持多样性还具有较强的局部搜索能力, 结合本文具有优良基因库等原因, 为了实验数据的有效性和严谨性, 将两者的 P_m 设为可收敛的最大近似值 0.41, 同时两者采用同一初始种群进化比较。

如图 8 所示, “I” 表示改进法的参数, 如 IP_{cav} 、 IP_o 等; “av” 表示连续运行 20 次的平均参数变化曲线, 如 P_{cav} 、 P_{oav} 等; “1” 表示其中的特例, 如 P_{c1} 、 P_{o1} 等。自适应参数主要表现在图 8 的上、中、下 3 个区间的变化曲线。

可以发现本文算法与 HSOS 一样, 在 BRE、ARE 和 WRE 均可取得最好的测试结果, 而其他算法却都会出现细微偏差, 说明本文算法不仅具有很好的稳定性, 而且具有非常好的局部搜索特性, 不易陷入局部最优, 这不仅得益于变异中的局部搜索, 更得益于 ITPX 算子和激素调节下的多样性变异, 同时算法的运行时间也在可接受范围内。

4.2.2 Rec 测试集

Rec 算例集为中、大规模的测试集, 为了检验

本文算法对中、大规模问题的性能, 与离散蝙蝠算法 (discrete bat algorithm, DBA)^[29]、混合差分估计分布算法 (differential evolution and estimation of distribution algorithm, DE-EDA)^[30]、混合共生生物搜索算法 (HSOS)^[13] 和变参数量子进化算法 (variable parameters quantum-inspired evolutionary algorithm, VP-QEA)^[31] 进行比较, 比较结果如表 4 所示, 其中 CPU/S 为连续运行 20 次的平均优化时间。

表 4 Rec 测试集的计算结果误差比较

实验组	问题规模	最优解	DBA			DE-EDA			HSOS			VP-QEA		IHCCM-IAGA			CPU/S
			BRE	ARE	WRE	BRE	ARE	WRE	BRE	ARE	WRE	BRE	ARE	BRE	ARE	WRE	
Rec01	20×5	1 247	0.000	0.080	0.160	0.000	0.020	0.160	0.000	0.000	0.000	0.000	1.050	0.000	0.064	0.160	3.3
Rec03	20×5	1 109	0.000	0.081	0.180	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.270	0.000	0.000	0.000	3.9
Rec05	20×5	1 242	0.242	0.242	0.242	0.000	0.230	0.240	0.000	0.000	0.000	0.240	0.990	0.000	0.217	0.242	3.8
Rec07	20×10	1 566	0.000	0.575	1.149	0.000	0.000	0.000	0.000	0.000	0.000	1.140	1.560	0.000	0.000	0.000	7.9
Rec09	20×10	1 537	0.000	0.638	2.407	0.000	0.010	0.260	0.000	0.000	0.000	0.970	2.080	0.000	0.000	0.000	7.3
Rec11	20×10	1 431	0.000	1.167	2.655	0.000	0.000	0.000	0.000	0.000	0.000	0.620	2.060	0.000	0.000	0.000	8.5
Rec13	20×15	1 930	0.415	1.461	3.782	0.000	0.580	1.040	0.000	0.273	0.777	1.340	2.480	0.000	0.228	0.415	12.3
Rec15	20×15	1 950	0.154	1.226	2.103	0.150	0.570	0.820	0.000	0.523	1.020	0.820	1.840	0.000	0.141	0.718	24.4
Rec17	20×15	1 902	0.368	1.227	2.154	0.370	1.080	2.100	0.000	1.388	2.154	2.780	3.940	0.000	0.223	1.157	19.8
Rec19	30×10	2 093	0.573	0.929	2.023	0.910	1.430	1.860	0.620	1.274	2.099	2.340	4.250	0.287	0.542	0.908	51.6
Rec21	30×10	2 017	1.438	1.671	2.231	1.140	1.550	1.636	1.437	1.537	2.033	1.630	3.520	1.190	1.445	1.636	46.7
Rec23	30×10	2 011	0.796	1.173	2.381	0.800	1.200	1.740	0.348	1.280	3.050	2.430	3.960	0.149	0.457	0.746	53.1
Rec25	30×15	2 513	1.632	2.921	3.940	1.230	2.250	3.220	0.835	2.067	2.850	2.620	4.610	0.279	0.854	2.069	87.4
Rec27	30×15	2 373	1.011	1.419	2.298	1.220	1.690	2.190	0.969	1.432	2.570	2.140	4.040	0.253	0.775	1.306	84.8
Rec29	30×15	2 287	1.049	2.580	3.935	1.620	2.320	2.890	0.831	2.488	2.970	4.320	5.200	0.000	0.901	1.530	90.2
Rec31	50×10	3 045	2.299	3.392	4.532	1.840	2.710	3.650	0.427	0.644	0.920	4.630	6.140	0.328	0.931	1.839	121.2
Rec33	50×10	3 114	0.610	0.728	1.734	0.350	0.770	0.870	0.000	0.565	0.835	1.280	2.890	0.000	0.621	0.835	126.7
Rec35	50×10	3 277	0.000	0.037	0.092	0.000	0.000	0.000	0.000	0.000	0.000	0.090	1.610	0.000	0.000	0.000	143.5
Rec37	75×20	4 951	3.373	4.872	5.979	3.980	5.090	6.140	2.565	3.001	3.555	6.840	7.810	2.222	2.849	3.555	484.6
Rec39	75×20	5 087	2.280	3.851	5.347	2.320	3.370	4.440	1.828	2.222	3.380	4.970	6.190	1.062	1.632	2.261	413.7
Rec41	75×20	4 960	3.810	5.095	6.532	3.750	5.190	6.290	2.388	3.350	3.770	6.990	7.800	2.097	3.000	3.770	418.3
AVG	—	—	0.955	1.684	2.660	0.937	1.431	1.883	0.583	1.050	1.523	2.295	3.538	0.375	0.709	1.102	105.4

从表中可发现, 本文算法在 21 个 Rec 算例取得的误差, 绝大多数都小于表中所比较的算法, 其中有 20 个算例的 BRE 优于或等于其他算法求解的 BRE; 有 17 个算例的 ARE 优于或等于其他算法的 ARE; 有 17 个算例的 WRE 优于或等于其他算法的 WRE。

图 10~12 所示为表 4 中各个算法在 BRE、ARE 和 WRE 的曲线图, 从图中可以发现不管是 BRE 还是 ARE、WRE, 本文算法的整体曲线均处在其他所有算法的最下面, 可见本文算法可以取得非常好的优化效果。本文算法在 BRE、ARE 和 WRE 的平均值上均为最小值, 排在所有比较算法中的第一位, 表现出本文算法具有很强的稳定性和收敛性。

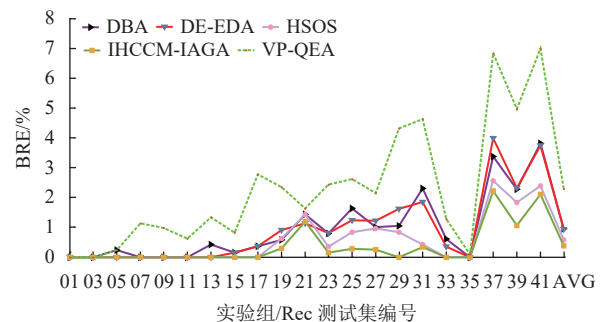


图 10 调度结果最佳相对误差比较

如图 13 所示为 Rec29 算例的进化曲线图, 可见算法在激素调节的作用下, 迭代 25 次左右就已经收敛于局部最优解, 此时自适应选择门槛、交叉

概率和变异概率均已达到理论峰值, 随后 ITPX 将结合变异算子中的局部搜索和湮灭因子等因素, 进行精细化搜索, 使得算法在中后期仍有很强的搜索能力和局部开发特性, 以至在迭代 250 次之后仍有下降趋势, 直至找到最优解 2 287。相比其他算法, 均无法找其最优解, 且存在一定误差, 体现了本文算法的有效性, 具有克服个体早熟、增强全局探索能力和局部开发特性。

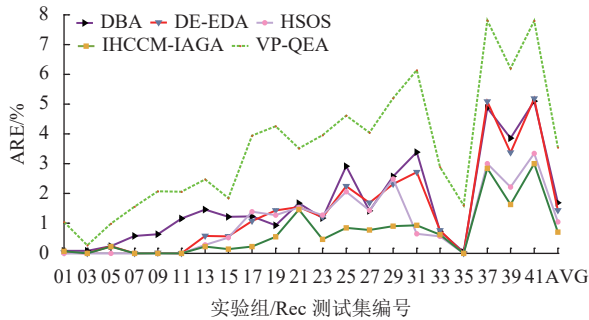


图 11 调度结果平均相对误差比较

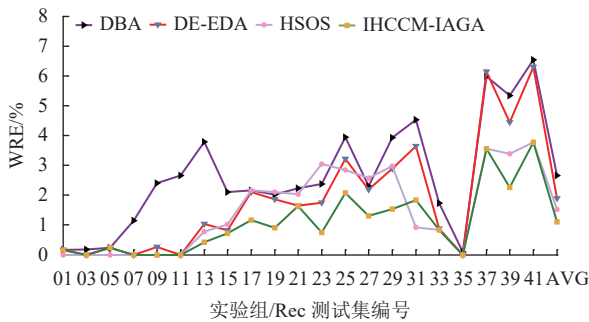


图 12 调度结果最差相对误差比较

因此, 在实际生产应用中, 若着重实时调度, 则 IHCCM-IAGA 在激素的调节下可快速收敛于最

优解或近似最优解, 优化时间较少, 如 Rec29 算例的近似最优解; 若着重解的质量, 则 IHCCM-IAGA 可在 ITPX 的支持下精细化搜索, 以达到最优解, 如 Rec29 算例的最优解。

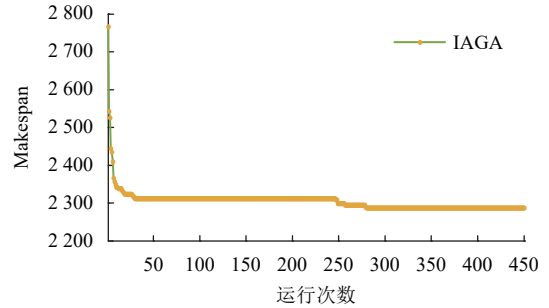


图 13 Rec29 算例的进化曲线

4.2.3 Taillard 测试集

Taillard 算例集由中到超大规模的测试集组成, 包括 20×5、20×10、20×20、50×5、50×10、50×20、100×5、100×10、100×20、200×10、200×20、500×20 共 12 个不同规模组成的算例集, 每个规模有 10 个算例, 取每种规模最后一个算例作为代表, 组成本文的测试集。

为了检验本文算法对中、大和超大规模问题的性能, 与混合遗传算法 (HGA)^[32]、混合共生生物搜索算法 (HSOS)^[13]、贪心禁忌搜索算法 (tabu-mechanism improved iterated greedy, TMIIG)^[33]、离散水波优化算法 (discrete water wave optimization algorithm, DWWO)^[34]、NEH 启发式算法^[35] 以及改进的贪心迭代算法 (improved iterated greedy algorithm, IIGA)^[36] 进行比较。比较结果如表 5 所示, 其中 C_{av} 表示该算法在一定运行次数内的平均值。

表 5 Taillard 测试集的计算结果比较

实验组	问题规模	最优解	C_{av}						
			IHCCM-IAGA	HGA	HSOS	TMIIG	DWWO	NEH	IIGA
Ta010	20×5	1 108	1 108.00	1 345.5	1 108.0	1 377.0	1 377.0	1 127	1 377.0
Ta020	20×10	1 591	1 601.30	2 019.3	1 601.0	2 051.0	2 051.0	1 656	2 051.0
Ta030	20×20	2 178	2 181.50	2 956.3	2 205.0	2 979.0	2 979.0	2 257	2 979.0
Ta040	50×5	2 782	2 782.00	3 341.5	2 782.0	3 327.2	3 336.5	2 822	3 377.2
Ta050	50×10	3 065	3 118.70	4 279.0	3 140.0	4 286.2	4 286.2	3 267	4 301.0
Ta060	50×20	3 756	3 850.40	5 963.5	3 887.0	5 959.0	5 958.8	4 036	5 982.0
Ta070	100×5	5 322	5 329.25	6 542.5	5 326.0	6 401.6	6 427.2	5 342	6 561.0
Ta080	100×10	5 845	5 894.10	8 255.3	5 898.0	8 148.8	8 141.0	8 186	8 263.6
Ta090	100×20	6 434	6 600.95	10 928.8	6 650.0	10 817.2	10 808.5	10 794	10 944.6
Ta100	200×10	10 675	10 730.00	15 869.1	10 798.0	15 410.8	15 412.2	15 803	15 754.4
Ta110	200×20	11 288	11 592.35	20 587.6	11 698.0	19 968.6	19 946.3	20 437	20 284.0
Ta120	500×20	26 457	26 850.10	49 545.6	26 780.8	47 402.4	47 183.2	49 092	48 483.4

本文使用平均错误率 (average error, AE) 评价算法对 Taillard 测试集的性能, 计算公式如下:

$$AE = \frac{1}{k} \sum_{i=1}^k \frac{C_{avi} - UB}{C_{avi}} \times 100\% \quad (22)$$

式中, C_{avi} 为算法求第 i 算例的平均完工时间; UB 为该算例目前已知解的上界值; k 为算例的数量。

从表 5 可以发现, IHCCM-IAGA 有 9 个算例的 C_{av} 均优于或等于其他算法, 有 3 个算例的 C_{av} 略高于 HSOS, 但均优于除 HSOS 外的其他算法, 可见 IHCCM-IAGA 在 Taillard 测试集上也具有较好的优越性。图 14 为基于 Taillard 测试集不同算法的平均错误率 (AE), 可以发现 IHCCM-IAGA 的 AE 为 1.11, 均小于其他算法, 其中与 HSOS 的 AE 相差值为 0.4, 与除 HSOS 外的其他算法 AE 相差值均超过 20, 表现出 IHCCM-IAGA 具有更好的稳定性和鲁棒性。

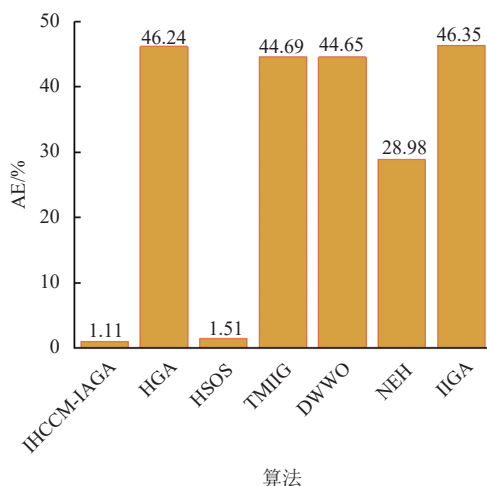


图 14 基于 Taillard 测试集不同算法的平均错误率

综述上面的实验结果与分析, 可以发现本文的 IHCCM-IAGA 具有更好的局部开发特性和更快的全局收敛速度, 使得解在质量和收敛速度上都有明显提升。

5 结束语

本文提出了一种基于改进激素浓度计算法的 IHCCM-IAGA 来求解 NP 问题的置换流水车间调度问题。针对传统遗传算法在求解 PFSP 中存在的不足和缺陷, 使用了一种基于改进激素浓度计算法的 IAGA, 通过激素的相互促进与抑制, 自适应地调整选择门槛、交叉概率和变异概率值, 使得种群可自适应各种进化环境, 同时引入优良基因库及免疫因子, 实现两种交叉方式并监控整个进化过程,

避免优质染色体的丢失。最后, 为了进一步提高算法在中后期的探索和开发能力, 提出了一种 ITPX 算子, 并结合组合变异算子和种群湮灭算子等, 确保 IHCCM-IAGA 在收敛速度和求解的质量上能够得到更好应用。通过 Rec41 算例比较实验, 验证了改进激素浓度计算法的优越性和有效性。通过国际标准算例比较实验, 也验证了 IHCCM-IAGA 和 ITPX 的有效性, 其在缩减最小完工时间的应用中可取得满意效果。

参考文献

- [1] PEZZELLA F, MORGANTI G, CIASCETTI G. A genetic algorithm for the flexible job-shop scheduling problem[J]. *Computers & Operations Research*, 2008, 35(10): 3202-3212.
- [2] ZIAEE M. A heuristic algorithm for solving flexible job shop scheduling problem[J]. *Journal of Supercomput*, 2014, 67(1): 69-83.
- [3] LIAO C J, TSENG C T, LUARN P. A discrete version of particle swarm optimization for flowshop scheduling problems[J]. *Computers & Operations Research*, 2007, 34(10): 3099-3111.
- [4] 张博凡, 黄宗南. 基于变形遗传算法交叉算子的 Flow-Shop 问题求解[J]. *制造业自动化*, 2011, 33(19): 27-29.
ZHANG B F, HUANG Z N. Solution of flow-shop problem based on deformed genetic algorithm crossover operator[J]. *Manufacturing Automation*, 2011, 33(19): 27-29.
- [5] 李小缤, 白焰, 耿林宵. 求解置换流水车间调度问题的改进遗传算法[J]. *计算机应用*, 2013, 33(12): 3576-3579.
LI X B, BAI Y, GENG L X. Improved genetic algorithm to solve the permutation flow shop scheduling problem[J]. *Computer Application*, 2013, 33(12): 3576-3579.
- [6] CHENG M Y, PRAYOGO D. Symbiotic organisms search: A new metaheuristic optimization algorithm[J]. *Computers and Structures*, 2014, 139(1): 98-112.
- [7] 刘志雄. 置换流水车间调度粒子群优化与局部搜索方法研究[J]. *机械设计与制造*, 2010(11): 167-169.
LIU Z X. Particle swarm optimization and local search method for permutation flow shop scheduling[J]. *Mechanical Design and Manufacturing*, 2010(11): 167-169.
- [8] SOROUGH S, SHAHRAM S, ABOLFAZL S. A heuristic scheduling method for the pipe-spool fabrication process[J]. *Journal of Ambient Intelligence and Humanized Computing*, 2018, 9(6): 1901-1918.
- [9] CALIS B, BULKAN S. A research survey: Review of AI solution strategies of job shop scheduling problem[J]. *Journal of Intelligent Manufacturing*, 2015, 26(5): 961-973.
- [10] ISHIBUCHI H, MURATA T. A multi-objective genetic local search algorithm and its application to flowshop scheduling[J]. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 1998, 28: 392-403.
- [11] ISHIBUCHI H. Balance between genetic search and local search in memetic algorithms for multi objective permutation flowshop scheduling[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 204-223.
- [12] 李林林, 刘东梅, 王显鹏. 基于改进 MOEA/D 算法的多目标置换流水车间调度问题[J]. *计算机集成制造系统*,

- 2021, 27(7): 1929-1940.
- LI L L, LIU D M, WANG X P. Multi objective permutation flow shop scheduling problem based on improved MOEA/D algorithm[J]. *Computer Integrated Manufacturing Systems*, 2021, 27(7): 1929-1940.
- [13] 秦旋, 房子涵, 张赵鑫. 混合共生生物搜索算法求解置换流水车间调度问题[J]. *浙江大学学报(工学版)*, 2020, 54(4): 712-721.
- QIN X, FANG Z H, ZHANG Z X. Hybrid symbiotic biological search algorithm for permutation flow shop scheduling problem[J]. *Journal of Zhejiang University (Engineering Edition)*, 2020, 54(4): 712-721.
- [14] GU W B, TANG D B, ZHENG K. Manufacturing resources coordination organisation and tasks allocation approach inspired by the endocrine regulation principle[J]. *IET Collaborative Intelligent Manufacturing*, 2020, DOI: [10.1049/iet-cim.2019.0031](https://doi.org/10.1049/iet-cim.2019.0031).
- [15] GU W B, LI Y X, ZHENG K, et al. A bio-inspired scheduling approach for machines and automated guided vehicles in flexible manufacturing system using hormone secretion principle[J]. *Advances in Mechanical Engineering*, 2020, 12(2): 1-17.
- [16] 王雷, 唐敦兵, 万敏, 等. 激素调节机制 IAGA 在作业车间调度中的应用研究[J]. *农业机械学报*, 2009, 40(10): 199-202.
- WANG L, TANG D B, WAN M, et al. Job-Shop scheduling problem based on improved adaptive genetic algorithm with hormone modulation mechanism[J]. *Transactions of the Chinese Society for Agricultural Machinery*, 2009, 40(10): 199-202.
- [17] 顾文斌, 唐敦兵, 郑堃, 等. 基于激素调节机制改进型自适应粒子群算法在置换流水车间调度中的应用研究[J]. *机械工程学报*, 2012, 48(14): 177-182.
- GU W B, TANG D B, ZHENG K, et al. Research on permutation flow-shop scheduling problem based on improved adaptive particle swarm optimization algorithm with hormone modulation mechanism[J]. *Journal of Mechanical Engineering*, 2012, 48(14): 177-182.
- [18] 顾文斌, 李育鑫, 钱煜晖, 等. 基于激素调节机制 IPSO 算法的相同并行机混合流水车间调度问题[J]. *计算机集成制造系统*, 2021, 27(10): 2858-2871.
- GU W B, LI Y X, QIAN Y H, et al. The same parallel machine hybrid flow shop scheduling problem based on ipso algorithm with hormone regulation mechanism[J]. *Computer Integrated Manufacturing Systems*, 2021, 27(10): 2858-2871.
- [19] 刘奕晨, 王毅, 牛奕龙, 等. 基于标准差的自适应激素调节遗传算法[J]. *数据采集与处理*, 2012, 27(3): 333-339.
- LIU Y C, WANG Y, NIU Y L, et al. An adaptive genetic algorithm based on hormone regulation[J]. *Journal of Data Acquisition and Processing*, 2012, 27(3): 333-339.
- [20] FARHY L S. Modeling of oscillations in endocrine with feedback[J]. *Methods in Enzymology*, 2004, 384: 54-81.
- [21] TIZHOOSH H R. Opposition-based Learning: A new scheme for machine intelligence[C]//*Proc of International Conference on Computational Intelligence for Modelling Control and Automation*. Piscataway: IEEE, 2005: 695-701.
- [22] GOLDBERG D E, DEB K. A comparative analysis of selection schemes used in genetic algorithms[C]//*Foundations of Genetic Algorithms*. San Francisco: Morgan Kaufmann, 1991: 69-93.
- [23] REEVES C R. A genetic algorithm for flowshop sequencing[J]. *Computers and Operations Research*, 1995, 22(1): 5-13.
- [24] CARLIER J. Ordonnancements à contraintes disjonctives[J]. *RAIRO-Operations Research*, 1978, 12(4): 333-350.
- [25] TAILLARD E. Benchmarks for basic scheduling problems[J]. *European Journal of Operational Research*, 1993, 64(2): 278-285.
- [26] TOSUN Ö, MARICHELVAM M K. Hybrid bat algorithm for flow shop scheduling problems[J]. *International Journal of Mathematics in Operational Research*, 2016, 9(1): 125-138.
- [27] TASGETIREN M F, LIANG Y C, SEVKLI M, et al. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem[J]. *European Journal of Operational Research*, 2007, 177(3): 1930-1947.
- [28] LIN Q, GAO L, LI X, et al. A hybrid backtracking search algorithm for permutation flow-shop scheduling problem[J]. *Computers and Industrial Engineering*, 2015, 85(C): 437-446.
- [29] LUO Q, ZHOU Y, XIE J, et al. Discrete bat algorithm for optimal problem of permutation flow shop scheduling[J]. *The Scientific World Journal*, 2014, DOI: [10.1155/2014/630280](https://doi.org/10.1155/2014/630280).
- [30] LI Z, GUO Q, TANG L. An effective DE-EDA for permutation flow-shop scheduling problem[J]. *2016 IEEE Congress on Evolutionary Computation*. [S.l.]: IEEE, 2016: 2927-2934.
- [31] 张先超, 周泓. 变参数量子进化算法及其在求解置换流水车间调度问题中的应用[J]. *计算机集成制造系统*, 2016, 22(3): 774-781.
- ZHANG X C, ZHOU H. Variable parameters quantum-inspired evolutionary algorithm and its application permutation flow-shop scheduling problem[J]. *Computer Integrated Manufacturing Systems*, 2016, 22(3): 774-781.
- [32] TSENG L Y, LIN Y T. A hybrid genetic algorithm for no-wait flowshop scheduling problem[J]. *International Journal of Production Economics*, 2010, 128(1): 144-152.
- [33] 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 flowshop scheduling problem[J]. *Applied Soft Computing*, 2015, 30(1): 604-613.
- [34] ZHAO F, LIU H, ZHANG Y, et al. A discrete water wave optimization algorithm for no-wait flow shop scheduling problem[J]. *Expert Systems with Applications*, 2017, 91(1): 347-363.
- [35] NAWAZ M, ENSCORE E, HAM I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem[J]. *The International Journal of Management Science*, 1983, 11(1): 91-95.
- [36] PAN Q K, WANG L, ZHAO B H. An improved iterated greedy algorithm for the no-wait flow shop scheduling problem with makespan criterion[J]. *The International Journal of Advanced Manufacturing Technology*, 2008, 38(7): 778-786.