

DOI: 10.13718/j.cnki.xdzk.2023.05.019

基于改进免疫遗传算法的汽车零件排样

闫嘉^{1,2}, 李林峰¹, 林毓培², 段书凯^{1,2}

1. 西南大学人工智能学院, 重庆 400715; 2. 智能传动和控制国家地方联合工程研究中心(重庆), 重庆 400715

摘要: 为提高汽车零件排样的板材利用率, 对汽车零件的排样技术进行研究. 汽车零件大多是不规则零件, 因此其排样属于二维不规则排样问题范畴. 在智能优化算法求解二维不规则排样领域, 遗传算法容易产生早熟现象, 并且局部寻优能力较差, 而免疫算法具有很强的局部寻优能力. 结合遗传算法与免疫算法的全局与局部寻优能力, 提出一种改进的免疫遗传算法. 该算法的遗传选择操作加入个体浓度计算, 操作步骤和变异概率可以自适应调整, 从而有效避免早熟现象. 自适应计算免疫选择个体数以及免疫克隆个体数以提高算法的收敛性能. 分别使用 ESICUP 网站提供的公开数据和实际汽车零件专用数据对提出的免疫遗传算法的性能进行测试, 实验结果表明, 和现有的几种算法相比, 本研究提出的免疫遗传算法能获得更高的材料利用率.

关键词: 二维不规则排样; 临界多边形; 汽车零件;

智能优化算法

中图分类号: TP391

文献标志码: A

开放科学(资源服务)标识码(OSID):



文章编号: 1673-9868(2023)05-0204-11

Automobile Parts Packing Based on Improved Immunogenetics Algorithm

YAN Jia^{1,2}, LI Linfeng¹, LIN Yupei², DUAN Shukai^{1,2}

1. College of Artificial Intelligence, Southwest University, Chongqing 400715, China;

2. National & Local Joint Engineering Research Center of Intelligent Transmission and Control Technology (Chongqing), Chongqing 400715, China

Abstract: To improve the plate utilization rate of automobile parts packing, the packing technology of automobile parts is studied. Most of the automobile parts are irregular, so this problem belongs to the category of two-dimensional irregular packing problem. In the field of intelligent optimization algorithms to solve the two-dimensional irregular packing, the genetic algorithm is liable to produce premature phenomenon and poor in local optimization ability, while immune algorithm has strong local optimization ability. In this paper, an improved immunogenetic algorithm is proposed by leveraging of combining the global and local optimization capabilities of the genetic algorithm and immune algorithm. The genetic selection opera-

收稿日期: 2022-05-23

基金项目: 国家重点研发计划项目(2018YFB1306600).

作者简介: 闫嘉, 博士, 副教授, 主要从事人工智能、智能信息处理、模式识别研究.

通信作者: 段书凯, 博士, 教授.

tion of the algorithm adds the individual concentration calculation, and the operation steps and mutation probability can be adaptively adjusted to avoid the premature phenomenon effectively. The number of immune-selected individuals and immune-cloned individuals are calculated adaptively to improve the convergence performance of the algorithm. The performance of the proposed immunogenetic algorithm was tested using the public data provided by ESICUP website and the actual automobile parts private data, respectively. The experimental results show that the proposed immunogenetic algorithm can achieve higher material utilization rate compared with several existing algorithms.

Key words: two-dimensional irregular packing; no fit polygon; automobile parts; intelligent optimization algorithms

随着计算机科学技术的发展,汽车制造业逐渐从传统制造向智能制造转型.汽车智能制造的关键环节包括汽车零件下料,也称为排样,是指将一系列不规则多边形零件不重叠地排放在给定的原板材上,目标是使板材利用率达到最高,当给定的原板材是定宽无限长板材时,该排样又称为二维不规则排样,此问题属于 NP-HARD 问题.原始材料的利用率对企业的市场竞争和经济效益有很大影响,特别是在全球自然资源有限的情况下,最大程度地节约资源,确保企业可持续发展显得至关重要.因此,该问题长期受到国内外研究人员的关注.

汽车零件排样属于二维不规则排样范畴,因此和传统二维不规则排样一样,明确待排零件的排样策略与排放次序是解决汽车零件排样问题的关键.排样策略有“底左”策略^[1]、“重心最低”策略^[2]、“混合定位”策略^[3]等.确定定位策略后,问题就转化为所有待排零件排放顺序优化问题.目前国内外大量文献都将排样算法与智能优化算法相结合解决排样次序优化问题.汤德估等人^[4]提出 GEFHNA 排样方法结合遗传算法和禁忌搜索算法优化排样顺序,选取 ESICUP 网站的通用数据集进行实验,仿真结果明显优于原始的 GEFHNA 排样方法;刘海明等人^[5]提出小生境遗传算法来求解二维不规则排样问题,在遗传算法中引入排挤原理的小生境策略,改善了遗传算法的早熟现象;杨卫波等人^[6]首次将实数编码的量子进化算法与不规则排样问题相结合,为不规则排样问题的求解提供了新思路;王静静等人^[7]提出并行交叉遗传算法求解二维不规则排样,该算法构造两个种群进行杂交进化,同样改善了遗传算法的早熟现象;Burke 等人^[8]提出了一种改进型“底左”填充算法来解决传统“底左”算法的缺陷,并将其与爬山算法以及禁忌搜索算法相结合,对 26 个通用数据集进行了测试,表现出更好的排样性能;Rao 等人^[9]提出了混合分支限界禁忌搜索方法求解二维不规则排样问题,该算法将分支限界搜索的全局寻优能力和禁忌搜索算法的局部寻优能力结合,弥补两种算法的缺陷;Shalaby 等人^[10]使用基于实数编码的粒子群优化算法优化多边形的排放次序,改善了以往十进制离散编码容易早熟的现象.然而,现有文献中用于求解二维不规则排样问题的优化算法大都有一定的缺陷,例如遗传算法容易早熟并且局部寻优能力差,模拟退火算法全局寻优能力较差等.虽然后续论文在原有算法基础上进行了一定的改进,但是效果并不明显,并且大多数文献的实验都是选用 ESICUP 网站提供的通用数据集进行算法验证,并没有针对具体应用进行实际效果测试,利用改进智能优化算法对实际汽车零件排样进行研究,国内还未见相关文献报道.

二维不规则排样过程涉及大量多边形重叠计算以及靠接位置计算,现有多边形重叠检测算法的时间复杂度为 $O((M+N)\log(M+N))$,其中 N 和 M 为两多边形的边数.为了提高算法的运行效率,有学者提出了临界多边形^[11]的概念,排样过程采用临界多边形模型可以省略重叠计算,还可以提供待排零件所有可能的靠接位置.基于上述分析,本文提出一种改进的免疫遗传算法,优化零件排放次序,使用临界多边形作为工具计算待排零件所有靠接位置进而基于“底左”策略选择排放点.算法验证方面首先使用 ESICUP 网站提供的通用数据集测试,然后使用汽车生产厂商提供的实际汽车零件进行排样测试.

1 二维不规则排样的数学模型

二维不规则排样要保证零件与零件不重叠、零件不超出原材料标示范围.本文研究的是二维矩形带排

样问题,即原材料定宽无限长.假设待排零件集合为 $P = \{p_1, p_2, \dots, p_n\}$, n 为零件数, p_i 表示第 i 个零件, O 为允许的旋转角度集合,结合二维不规则带排样的目标及约束给出二维不规则条带排样的数学模型如式(1)所示:

$$\begin{aligned} \min L &= \max\{x \mid (x, y) \in p_i(o_i)\} - \min\{x \mid (x, y) \in p_i(o_i)\}, 1 \leq i \leq n \\ \text{st: } &p_i, p_j \in P, p_i(o_i) \cap p_j(o_j) = \emptyset, 1 \leq i < j \leq n \\ &p_i(o_i) \subseteq C(W, L), 1 \leq i \leq n \\ &o_i \in O, 1 \leq i \leq n \end{aligned} \quad (1)$$

式(1)中优化目标 L 为原材料长度,即寻找一种排样方案使得 L 最小; (x, y) 表示零件二维坐标点; $p_i(o_i)$ 表示零件 p_i 按角度 o_i 进行顺时针旋转以后的坐标点集合; $C(W, L)$ 指原材料表示的坐标范围; W 为原材料的宽度;约束条件中 $p_i(o_i) \cap p_j(o_j) = \emptyset$ 保证零件之间不重叠, $p_i(o_i) \subseteq C(W, L)$ 保证零件被排放在原材料内.

2 基于重心临界多边形的 BL 排样策略

2.1 临界多边形

临界多边形(No Fit Polygon, NFP)^[11]:给定多边形 P_i, P_j , P_i 静止,选 P_j 边上或内部点作参考点,将 P_j 不旋转地绕 P_i 滑动一周,滑动过程 P_j 始终与 P_i 相切, P_j 上的参考点所形成的轨迹叫做 P_j 相对于 P_i 的临界多边形,记为 $NFP(P_i, P_j)$.如图 1, x_j 为 P_j 的参考点:

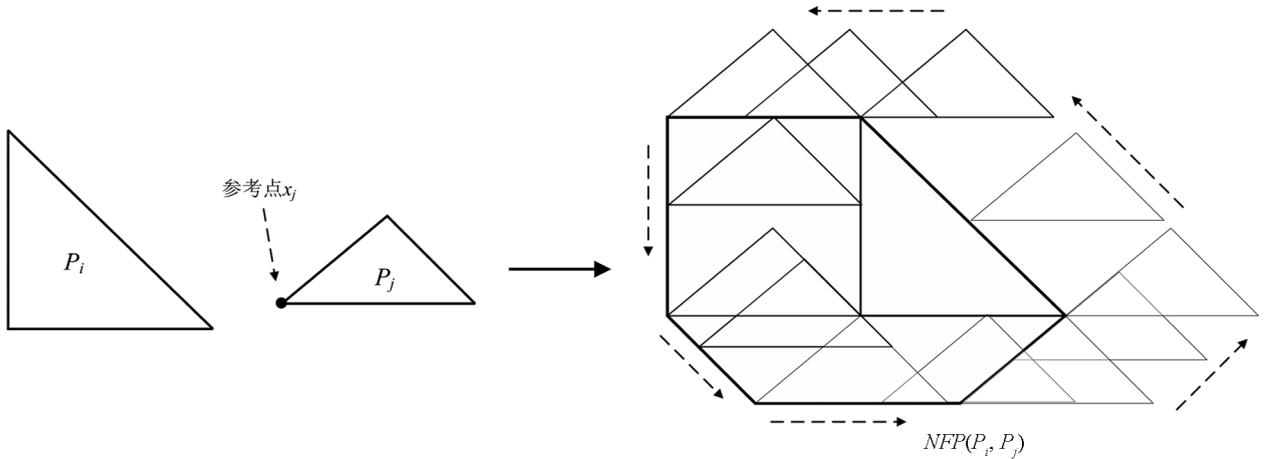


图 1 临界多边形示意图

NFP 有如下重要特性:选取 P_j 的参考点在临界多边形内部时多边形 P_i 和 P_j 重叠;选取 P_j 的参考点在 NFP 边上时多边形 P_i 和 P_j 相切但不重叠;选取 P_j 的参考点在 NFP 外部时多边形 P_i 和 P_j 处于相离状态.

内靠接矩形(Inner Fit Rectangle, IFR)^[12]:指多边形 P_i 在矩形 C 内部滑动过程参考点形成的轨迹,滑动过程中也要满足两多边形边不相交,如图 2 所示:

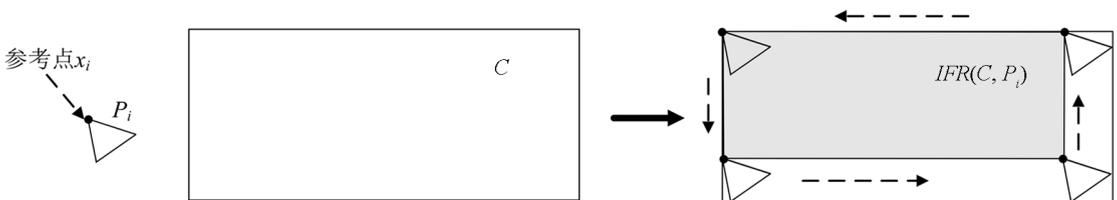


图 2 内靠接矩形 IFR 示意图

依靠 NFP 与 IFR 可以计算待排零件在板材内部的所有靠接位置,假设 P_1, P_2, \dots, P_{i-1} 为已排零件,当前待排零件为 P_i ,原材料为矩形 C ,具体算法步骤如下:

- 步骤1 计算 P_i 与 P_1, P_2, \dots, P_{i-1} 的临界多边形, 记为 $NFP_{1,i}, NFP_{2,i}, \dots, NFP_{i-1,i}$;
- 步骤2 计算 P_i 与 C 的内靠接矩形, 记为 IFR_i ;
- 步骤3 计算 $NFP_{1,i} \cup NFP_{2,i} \cup \dots \cup NFP_{i-1,i}$, 并集记为 $Union_{NFP}$;
- 步骤4 计算 $IFR_i - Union_{NFP}$, 差集多边形记为 Dif , Dif 边上点都是 P_i 的可排放位置。

计算示意图如图3所示, 假设待排零件为三角形, 选取三角形的重心作为参考点, 已排零件为两个正八边形, 灰色区域即为差集多边形, 边上所有点即为待排零件的可排放点。

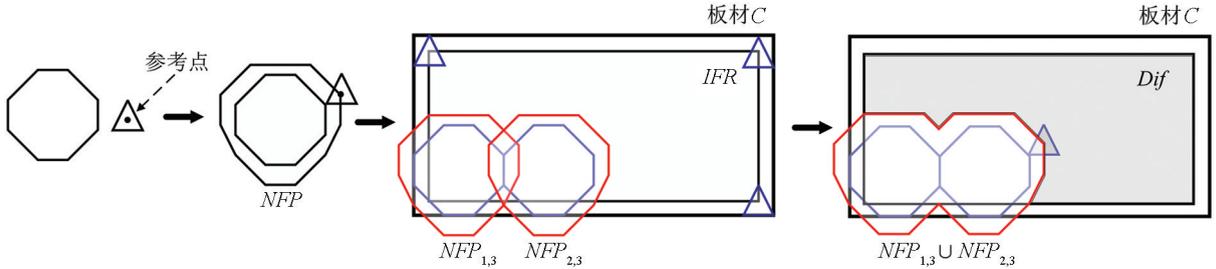


图3 待排零件靠接位置计算示意图

上述算法省去了零件之间的重叠计算, 只需将待排零件的参考点移动到差集多边形上某一点即可, 排放就简化为基于差集多边形的定位选优过程。基于该算法, 如果较大零件之间产生孔洞, 后续面积较小的零件还可以填充该孔洞。

2.2 通过 GNFP 和 GIFR 选取 BL 排样点

重心临界多边形(Grivaty-NFP, GNFP)指选取滑动多边形的重心作为参考点构造的临界多边形。同样, 重心内靠接矩形(Grivaty-IFR, GIFR)的构造也选取滑动多边形重心作为参考点。在多边形定位算法中, 应用最多、最广泛的是“底左(Bottom-Left, BL)”^[1]定位算法, 其基本思想是将待排零件尽量往板材的最左最下角排放。采用 GNFP 与 GIFR 模型计算排样点后, 将待排零件放到最左最下位置, 可以保证零件的重心位置最低, 零件的重心位置越低, 越有利于降低整体排样方案的重心位置, 从而有利于提高面积分布密度, 提高原板材的利用率。降低每个零件重心位置还可以使剩余板材的边界较为平缓, 有利于后续零件的放置。

3 改进免疫遗传算法设计

遗传算法(Genetic Algorithm, GA)^[13]是一种模拟自然界物种遗传进化机理的算法, 具有很强的全局寻优能力, 但是局部寻优能力较差。免疫算法(Immune Algorithm, IA)^[14]是受人体免疫学原理启发而提出的, 该算法通过免疫选择、克隆及变异操作实现对解空间的局部搜索, 因此具有很强的局部寻优能力。若将两种算法结合, 既可以继承遗传算法的全局寻优能力, 又可以兼顾免疫算法的局部寻优能力。

3.1 自适应遗传算法设计

遗传算法应用于二维不规则排样问题容易产生早熟现象, 算法运行初期种群个体容易变得极为相似, 这时交叉操作很难改变个体基因顺序, 而变异概率往往取值很小, 导致算法停滞。文献[15]通过自适应调整遗传算法步骤, 以及交叉、变异概率, 有效避免了遗传算法产生早熟现象。本文改进简化文献[15]的算法过程, 在每次迭代进行交叉变异前根据平均适应度 f_{avg} 与最好适应度 f_{best} 的比值与 δ 的关系(δ 一般取 $(0.9, 1)$)修订相关操作及参数, 如果比值小于 δ , 说明种群个体比较分散, 则按传统遗传算法步骤先交叉再变异, 如果比值大于等于 δ , 说明种群比较集中, 交叉操作不易产生较大改变, 这时增大变异概率, 让变异概率等于交叉概率, 先进行变异再进行交叉, 可以帮助种群跳出局部解。

3.2 改进选择操作

个体浓度表征种群多样性。个体浓度偏高意味种群中相似个体较多, 种群会集中在某一区间, 不利于全局搜索, 因此算法应对浓度过高的个体进行抑制, 保证种群多样性。个体浓度一般定义为:

$$D(p_i) = \frac{1}{N} \sum_{j=1}^N s(p_i, p_j) \quad (2)$$

$D(p_i)$ 表示第 i 个个体的浓度; N 为种群规模; $s(p_i, p_j)$ 表明个体 p_i 与 p_j 之间是否相似, 相似取值为 1, 否则取值为 0. 具体公式如下:

$$s(p_i, p_j) = \begin{cases} 1 & a(p_i, p_j) \leq \delta_s \\ 0 & a(p_i, p_j) > \delta_s \end{cases} \quad a(p_i, p_j) = \frac{\sum_{k=1}^D \partial_k}{d} \quad \partial_k = \begin{cases} 1 & p_{ik} = p_{jk} \\ 0 & p_{ik} \neq p_{jk} \end{cases} \quad (3)$$

$a(p_i, p_j)$ 表示个体 p_i 与个体 p_j 的相似度; δ_s 为相似度阈值; d 为解的维度; p_{ik} 为 p_i 的第 k 维度. 个体之间相似度计算根据个体编码方式不同采用不同的计算方式. 本文个体采用十进制顺序编码, 所以采用基于海明距离^[16]的计算方法. 综合考虑适应度和浓度, 计算种群所有个体激励度的方法如下:

$$S(p_i) = \alpha F(p_i) + \beta D(p_i) \quad (4)$$

$S(p_i)$ 为个体 p_i 的激励度; $F(p_i)$ 为 p_i 的适应度; α 和 β 为权重系数, 满足 $\alpha + \beta = 1$, 用于调节个体适应度和浓度对激励度的影响. 进行选择操作时, 根据种群个体激励度进行选择可以有效保证种群的多样性.

3.3 自适应免疫选择与克隆

人工免疫算法免疫选择操作需要选择种群中的个体进行克隆, 假设每次迭代选择的个体数 N 固定, 如果 N 取值较大会影响算法的运行效率, 取值较小会影响算法的收敛精度. 所以 N 的取值很关键, 本文通过自适应调整 N 的取值来确保算法的效率以及精度, 具体公式如下:

$$S_{avg}(G) = \frac{\sum_{i=1}^n S_G(p_i)}{n} \quad (5)$$

$$M(G) = \begin{cases} p_i & S_G(p_i) \geq S_{avg}(G) \\ \emptyset & S_G(p_i) < S_{avg}(G) \end{cases} \quad (6)$$

$S_{avg}(G)$ 为第 G 代种群平均激励度; $S_G(p_i)$ 为第 G 代个体 p_i 激励度; n 表示种群规模; $M(G)$ 表示克隆选择出的个体集合.

选择出的个体需要进行克隆操作, 假设每个个体复制出 N_c 份, 同样 N_c 取值太大影响算法运行效率, 取值太小影响收敛精度. 通过自适应调整 N_c 的值能够提高算法的运行效率, 具体公式如下:

$$C(p_j) = \text{int} \left(\frac{\omega \times S(p_j)}{\sum_{j=1}^{CL} S(p_j)} \right) \quad (7)$$

$C(p_j)$ 为个体 p_j 克隆出的个体数; ω 为克隆系数; CL 免疫选择的个体数; $\text{int}()$ 表示向上取整操作.

3.4 改进免疫遗传算法步骤

基于以上分析, 本文使用的改进免疫遗传算法(IGAA)具体步骤如下:

步骤 1 设置参数, 初始化种群, 计算初始种群的适应度;

步骤 2 判断是否满足终止条件, 若满足输出最优解并停止迭代, 否则执行步骤 3;

步骤 3 计算种群平均与最高适应度 f_{avg} , f_{best} , 如果 $f_{avg}/f_{best} \leq \delta$, 则执行步骤 4, 否则执行步骤 5;

步骤 4 种群进行交叉变异操作, 计算子代适应度、浓度, 根据式(4)选择产生新一代种群;

步骤 5 临时调整变异概率等于交叉概率, 种群进行变异交叉操作, 计算子代适应度、浓度, 根据式(4)选择产生新一代种群;

步骤 6 根据式(5)(6)计算免疫选择操作需要选择出的个体数量, 并进行免疫选择操作;

步骤 7 根据式(7)计算经免疫选择的个体需要复制出的数量, 并进行克隆操作;

步骤 8 将步骤 7 中复制出的个体进行变异操作;

步骤 9 克隆抑制操作, 计算步骤 8 中变异后每个个体的适应度, 保留适应度最高的克隆变异个体, 如果其适应度比原克隆体适应度高就进行替换. 否则不替换, 并转步骤 2.

4 优化二维不规则排样的免疫遗传算法设计

4.1 编码方法

优化算法首要解决的问题是个体的编码问题, 本文采用十进制顺序编码, 例如某个体编码: $[(1, 90),$

(2, 90), (3, 180), (4, 0), (5, 90)], 每一个基因由两位组成, 第一位表示零件编号, 第二位表示对应零件的旋转角度.

4.2 交叉算子

本文采用顺序循环交叉: 两个父辈个体 P_1, P_2 交叉产生子代个体 Son_1, Son_2 , 先生成随机数 pos_1, pos_2 , P_1, P_2 个体 $[pos_1, pos_2]$ 之间的基因不变. 以 Son_1 为例进行说明: 遍历 P_2 个体的基因, 并赋给 P_1 的 $[0, pos_1)$ 和 (pos_2, N) , 遍历过程中若 P_2 的基因已经在 P_1 的 $[pos_1, pos_2]$ 之间出现则直接跳转到 P_2 的下一位, Son_2 同理. 具体操作如下: 假设 $pos_1=2, pos_2=5$, 并且有两个父代个体 P_1, P_2 :

$$P_1 = [(1, 90), (2, 90), \underline{(3, 0)}, (4, 180), (5, 270), \underline{(6, 180)}, (7, 0), (8, 180), (9, 90)]$$

$$P_2 = [(5, 270), (3, 0), \underline{(2, 90)}, (4, 180), (1, 90), \underline{(6, 180)}, (9, 90), (8, 180), (7, 0)]$$

P_1, P_2 中加下划线的表示不变的基因位, 交叉后产生子代个体:

$$Son_1 = [(2, 90), (1, 90), \underline{(3, 0)}, (4, 180), (5, 270), \underline{(6, 180)}, (9, 90), (8, 180), (7, 0)]$$

$$Son_2 = [(3, 0), (5, 270), \underline{(2, 90)}, (4, 180), (1, 90), \underline{(6, 180)}, (7, 0), (8, 180), (9, 90)]$$

4.3 变异算子

本文基因的变异方法使用以下 3 种:

序号插入法: 随机选取染色体中的某一位基因插入到随机位置, 设染色体: $[(1, 90), \underline{(2, 90)}, (3, 0), (4, 180), (5, 270)]$, 假设将下标为 1 的基因插入到下标为 3 的位置, 插入后变成: $[(1, 90), (3, 0), (4, 180), \underline{(2, 90)}, (5, 270)]$.

两个零件交换位置: 随机选取两位基因交换位置, 设染色体: $[(1, 90), \underline{(2, 90)}, (3, 0), \underline{(4, 180)}, (5, 270)]$, 交换下标为 1 和下标为 3 的基因, 交换后变成 $[(1, 90), \underline{(4, 180)}, (3, 0), \underline{(2, 90)}, (5, 270)]$.

两组零件交换顺序: 随机选取两组基因交换位置, 先对染色体基因进行分组, 将染色体基因分成若干个一组, 然后随机交换两组基因的位置.

前两种变异方式适用于小规模零件的排样, 第三种方式适用于大规模零件的排样.

上述 3 种变异方法是对个体基因顺序变异, 如果不对角度变异就无法产生新的旋转角度. 角度变异通过随机选择基因位, 然后根据可旋转角度随机修改该基因位的角度值. 每次进行角度变异时, 随机选择 $\text{int}(\text{numPieces}/10)$ 位基因进行角度变异, 其中 numPieces 为待排零件的数量.

4.4 适应度计算

本文研究的是二维矩形带排样问题, 即原材料是定宽无限长, 排样的目标就是使排样方案所切割的板材长度尽可能短, 所以个体适应度表示如下:

$$F(p_i) = \frac{\sum_{j=1}^n A(P_j)}{L \times W} \quad (8)$$

上式 p_i 表示第 i 个个体; P_j 表示第 j 个零件; $A(P_j)$ 表示第 j 个零件的面积; L 为切割的板材长度; W 为板材宽度.

4.5 算法步骤

假设种群规模为 M , 交叉概率 P_c , 变异概率 P_m , $pbest$ 用于记录每次迭代种群中适应度最高个体. 迭代过程中, 若出现超过 K 代种群最优解未发生变化, 说明算法陷入极值点, 此时对种群中除最优个体外所有个体进行变异以便种群跳出局部极值.

步骤 1 构造初始种群并计算种群适应度, 记录适应度最高的个体到 $pbest$ 中;

步骤 2 判断是否满足终止条件, 若满足则输出最优解并停止迭代, 否则执行步骤 3;

步骤 3 计算当前种群平均适应度 f_{avg} 以及适应度最高个体 f_{best} , 如果 $f_{avg}/f_{best} \leq \delta$, 则执行步骤 4, 否则执行步骤 5;

步骤 4 将种群个体随机组合, 遍历每一对个体, 根据交叉变异概率进行交叉变异操作, 转向步骤 6;

步骤 5 临时让 $P_m = P_c$, 将种群中的个体随机组合, 遍历每对个体, 根据变异交叉概率进行变异交叉

操作, 转向步骤 6;

步骤 6 计算交叉变异后子代个体的适应度, 将子代与父代合并, 计算所有个体浓度以及激励度, 选择激励度最高的前 M 个个体形成新的种群;

步骤 7 计算免疫选择出的个体数量, 进行免疫选择以及克隆操作;

步骤 8 对所有克隆出的个体进行变异操作并计算变异后所有克隆体的适应度;

步骤 9 克隆抑制, 将变异个体中适应度最高个体与原个体适应度比较, 如果比原个体高就进行替换, 否则不替换;

步骤 10 若新种群中适应度最高的个体比 $pbest$ 的适应度更高, 更新 $pbest$, 否则判断是否超过 K 代 $pbest$ 未更新, 是则需要对种群中除适应度最高个体外的所有个体进行变异, 否则转步骤 2.

5 仿真实验

为了验证算法的可行性及性能, 本文下载 ESICUP 的 14 个基准问题来测试算法性能, 然后使用汽车生产厂商提供的汽车零部件图测试本文算法. 本文算法仿真使用的计算机为 PC 机, 处理器为 AMD R7 5800H, 基准频率为 3.2 GHz, 内存为 16 G, Win11 操作系统, 在 VisualStudio2017 平台进行, 使用 C++ 作为仿真语言. 本文将每个测试用例测试 30 次, 记录最优结果以及平均结果. 相关参数以及操作说明:

1) 初始化参数说明: IGAA 中的免疫克隆操作后会对所有克隆体进行适应度评估, 为了保证算法收敛速度, 种群规模不宜太大, 设置种群规模为 12. 交叉概率过小不利于种群特征遗传, 变异概率过大会使算法近似于随机搜索, 因此交叉概率设为 0.9, 变异概率设为 0.1; 阈值 δ 和 δ_s 一般取接近 1 的数, 本文 $\delta = \delta_s = 0.95$; 在能够找到最优解的情况下尽可能减少迭代次数以减少算法运行时间, 因此最大迭代次数设为 200 次; 克隆系数与种群规模有关, 为了保证每代都能产生克隆体, 本文 $\omega = 20$, 激励度计算的权重系数 $\alpha = 0.9$, $\beta = 0.1$; 在迭代过程中, 若出现超过 K 代种群最优解未发生变化, 说明算法陷入极值点, 此时对种群中除最优个体外所有个体进行变异操作以便种群跳出局部极值, K 取值太大起不到作用, 取值太小算法随机性太强, 本文 $K = 15$.

2) 本文取零件数为 30, 60 为分界值, 当小于 30 时零件数较少, 以相同概率采取 4.3 部分所述的前两种变异方式; 大于 30 小于 60 时以 0.4 的概率采取第三种变异方式; 大于 60 时, 以 0.9 的概率采取第三种变异方式.

3) 种群初始化: 种群初始化方法包括按零件面积递减、递增排序; 按矩形度递减、递增排序; 按零件长边递减、递增排序; 其他个体随机产生.

5.1 ESICUP 的通用数据集测试

目前, 用于求解二维不规则排样的优化算法大多都是遗传算法(GA)、模拟退火(SA)等, 因此本文首先选取 3 种传统优化算法 GA、SA、IA 以及近几年被提出的鲸鱼优化算法(WOA)^[17-18]、乌鸦搜索算法(CSA)^[19-20]等 5 种优化算法与 IGAA 进行比较, 其中 WOA 与 CSA 解的编码方式使用文献[10]中的方法, 对比结果见表 1. 其次, 本文还将 IGAA 与文献[9]的混合分支限界禁忌搜索算法(BSTS)以及文献[21]的混合遗传模拟退火算法(GSAA)进行对比, 结果见表 2. 表中 Best 表示运行的最优利用率; Avg 表示运行的平均利用率; Dev 表示算法最优结果与平均结果的差值; 最后一行的 Tot Avg 表示 14 个数据库对应指标的平均值. 从表 1 可以看出, 混合 IGAA 算法无论在最优利用率还是平均利用率方面都要优于单一智能优化算法.

分析表 2 中 14 个数据集的运行结果, IGAA 在 10 个数据集的最优利用率结果高于 BSTS 的最优利用率; IGAA 在 9 个数据集的平均利用率高于 BSTS 的平均利用率, 其中 Dighe1、Dighe2、Dagli 数据集的平均利用率比 BSTS 的最优利用率都要好; 与 GSAA 比较, IGAA 的所有数据集, 无论是最优利用率还是平均利用率, 都比 GSAA 的要高; 分析总体平均 Total Avg 值, 无论是最优利用率还是平均利用率, IGAA 均为最高; 算法稳定性方面分析每一行 Dev 的值, 由于 Dighe1、Dighe2 属于“拼图”类型的排样, 零件数较

少, 所以 IGAA 的平均利用率与最优利用率相差很大, 但是随着数据集规模的增大, IGAA 算法的稳定性要比 BSTS、GSAA 的稳定性要好. 图 4 展示了算法运行所有数据集的收敛曲线:

表 1 GA、IA、SA、WOA、CSA 算法与 IGAA 算法结果对比

DATA	GA		IA		SA		WOA		CSA		IGAA	
	Best/ %	Avg/ %										
Dighe1	76.16	75.56	78.89	77.58	78.82	75.58	81.23	78.94	77.45	75.18	99.99	85.00
Dighe2	78.59	77.74	84.36	80.16	81.90	77.90	88.63	79.28	79.94	78.88	99.99	93.37
Fu	86.36	83.84	86.36	83.24	87.25	84.01	83.88	82.47	84.53	82.81	89.31	85.98
Jakobs1	78.95	75.43	77.37	75.89	80.56	77.23	80.50	76.82	78.83	75.90	81.74	81.59
Jakobs2	72.18	70.64	72.23	70.68	71.58	70.20	72.23	70.58	71.97	70.23	77.22	74.00
Albano	82.98	81.50	82.83	82.16	83.54	81.89	83.83	81.91	82.67	81.06	85.67	84.35
Mao	78.84	76.78	78.90	76.24	78.12	76.53	78.84	76.93	79.25	77.54	81.97	79.00
Dagli	80.29	78.84	81.21	79.47	81.83	78.24	79.96	77.73	80.17	78.12	84.67	83.81
Marques	84.23	83.34	85.30	83.48	84.40	83.30	84.71	83.43	85.12	83.48	87.57	85.84
Shapes0	59.48	57.87	59.28	58.23	58.79	57.18	59.96	59.00	60.16	58.89	61.64	60.70
Shapes1	63.56	62.61	63.64	62.34	62.36	60.92	62.84	61.08	63.07	61.83	67.20	66.00
Blaz1	75.10	74.52	75.74	74.89	76.40	75.14	78.38	75.27	77.94	75.59	79.58	77.61
Trousers	87.88	86.24	86.94	86.23	87.53	86.11	88.01	86.24	87.44	86.20	88.84	88.07
Swim	68.84	67.90	68.47	67.56	68.23	67.30	67.90	67.03	68.94	67.21	71.82	70.10
Tot Avg	76.67	75.20	77.25	75.58	77.23	75.11	77.92	75.48	76.96	75.21	82.66	79.67

表 2 BSTS、GSAA 与 IGAA 算法结果对比

Data	BSTS			GSAA			IGAA		
	Best/%	Avg/%	Dev	Best/%	Avg/%	Dev	Best/%	Avg/%	Dev
Dighe1	82.21	79.48	2.73	78.24	76.16	2.08	99.99	85.00	14.99
Dighe2	85.12	82.56	2.56	88.16	78.24	9.92	99.99	93.37	6.62
Fu	88.79	86.38	2.41	88.31	84.68	3.63	89.31	85.98	3.33
Jakobs1	83.55	81.87	1.68	81.64	79.90	4.74	81.74	81.59	0.15
Jakobs2	75.72	74.66	1.06	74.24	72.23	2.01	77.22	74.00	3.22
Albano	84.46	83.37	1.09	84.24	82.78	1.46	85.67	84.35	1.32
Mao	81.00	80.12	0.88	79.84	77.78	2.06	81.97	79.00	2.97
Dagli	83.01	80.93	2.08	81.40	79.24	2.16	84.67	83.81	0.86
Marques	87.31	86.40	0.91	87.21	85.70	1.51	87.57	85.84	1.73
Shapes0	62.19	60.38	1.81	60.21	59.48	1.73	61.64	60.70	0.94
Shapes1	67.55	65.26	2.29	65.56	63.62	1.94	67.20	66.00	1.20
Blaz1	80.23	75.82	4.41	77.68	75.32	2.36	79.58	77.61	1.97
Trousers	88.84	86.41	2.43	88.21	87.34	0.87	88.84	88.07	0.77
Swim	71.74	69.46	2.28	69.74	68.59	1.15	71.82	70.10	1.22
Tot Avg	80.12	78.08	2.04	78.91	76.48	2.69	82.66	79.67	2.95

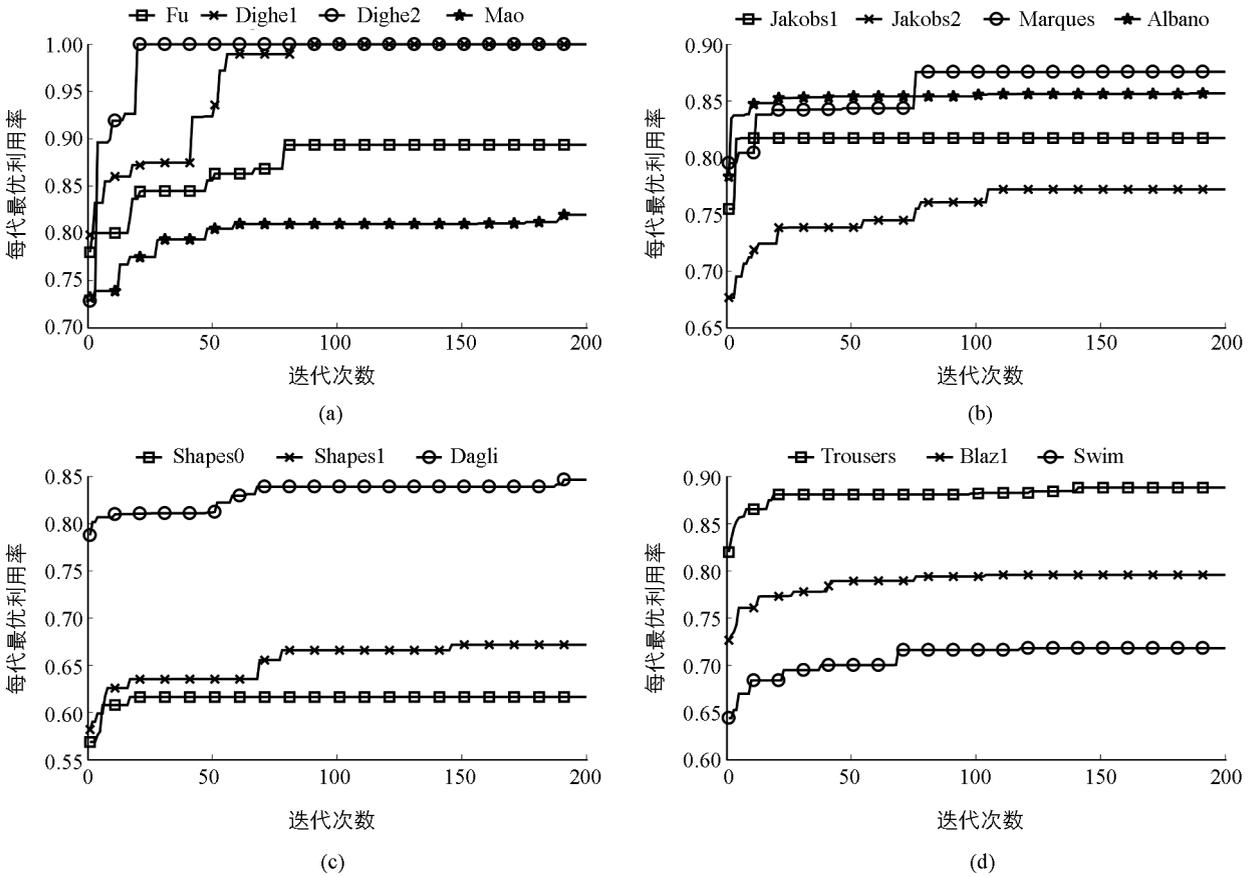


图 4 各个数据集曲线收敛情况

分析图 4, Fu、Dighe1、Dighe2、Jakobs1、Marques、Shapes0 曲线可以在 100 代内收敛; Albano、Jakobs2、Shapes1、Blaz1、Trousers、Swim 曲线可以在 150 代内收敛, Mao、Dagli 曲线在 150 代以后出现轻微变化. 总的来说 IGAA 可以在 150 代以内收敛到一个不错的解. 图 5 展示了几个基准数据集排样图.

5.2 实际汽车零件排样测试

为了验证本文算法的实用性, 本文选取重庆某汽车装配制造公司提供的汽车零件数据进行算法测试. 实际生产当中, 往往需要对单一零件进行排样, 因此本文测试单一零件的排样以及多种零件的混合排样, 并将本文算法的运行结果同开源软件 SvgNest 以及 5.1 节的 GA、IA、SA、WOA、CSA 的运行结果进行对比, 将每个汽车零件数据集运行 20 min 并记录最终结果, 对比结果见表 3.

表 3 实际汽车零件数据的对比结果

	GA/%	IA/%	SA/%	WOA/%	CSA/%	SvgNest/%	IGAA/%
零件 1	78.84	80.12	78.89	81.56	80.94	78.28	84.64
零件 2	90.23	88.24	89.37	90.23	89.88	90.23	90.23
混合零件	78.67	79.14	79.85	80.98	81.17	78.54	82.27

从表 3 可以看出, 无论是单一零件还是多种零件混合排样, IGAA 算法仿真结果都要优于其他算法的仿真结果. 表明本文的 IGAA 算法具有一定的实用性. 图 6 为 IGAA 算法的排样结果.

6 结论

本文针对二维不规则排样问题, 使用基于重心临界多边形选取“BL”排放点, 保证将待排多边形排放到重心最低的位置; 在确定定位规则后, 问题就转化为基于序列的优化问题, 对于此, 本文提出一种改进的免疫遗传算法, 兼顾了两者的全局与局部寻优特性, 并且自适应调整算法的运行步骤及相关参数, 从而有效避免算法的早熟现象. 对相关基准问题测试, 表明了该算法的有效性. 对于实际汽车零件的排样, 本文算法表现出的排样效果也优于开源软件 SvgNest 和现有 5 种智能优化算法的排样结果. 总的来说, 本文提

出的 IGAA 算法在解决汽车零件排样问题具有一定的应用价值。

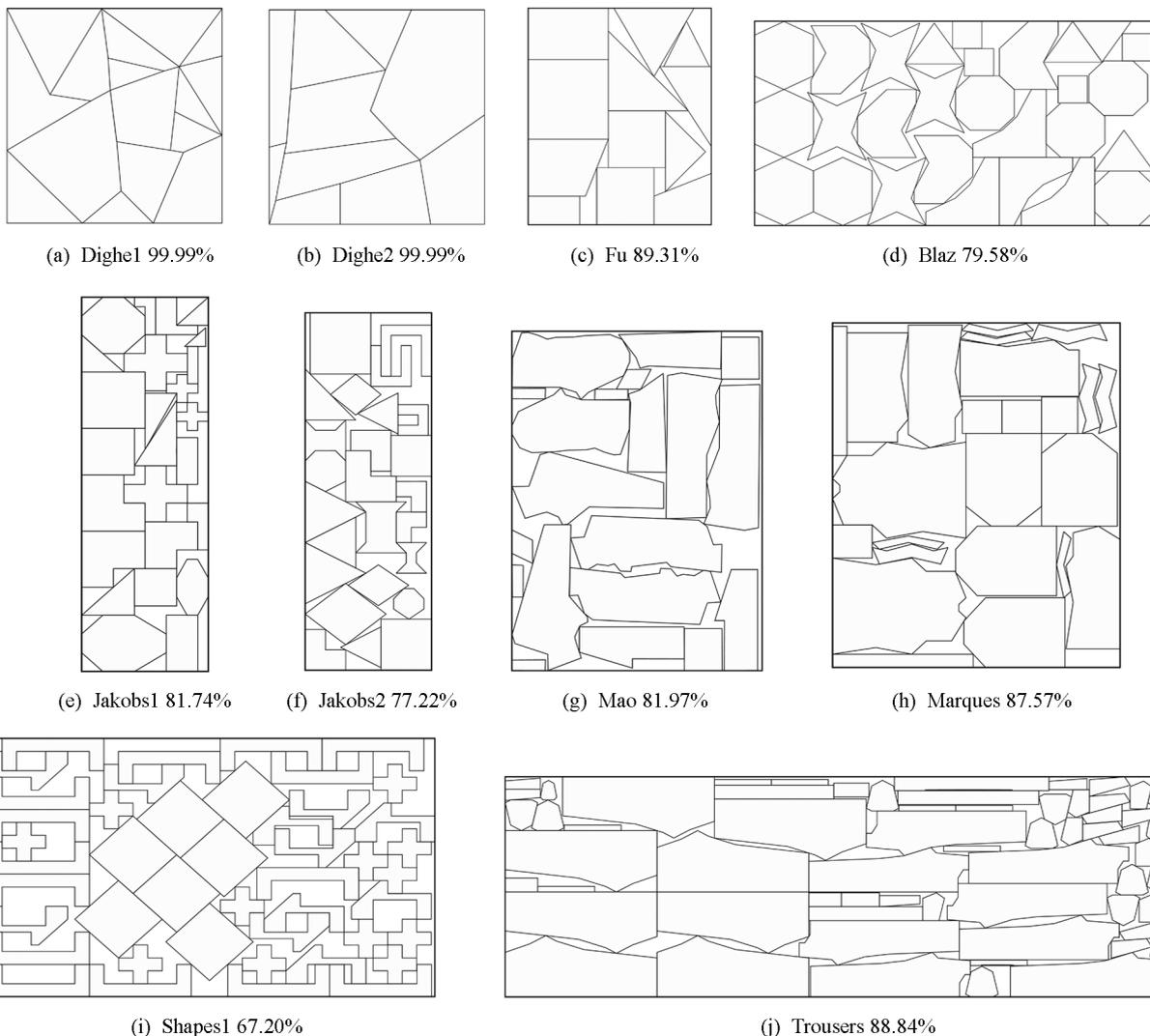


图 5 IGAA 算法排样图

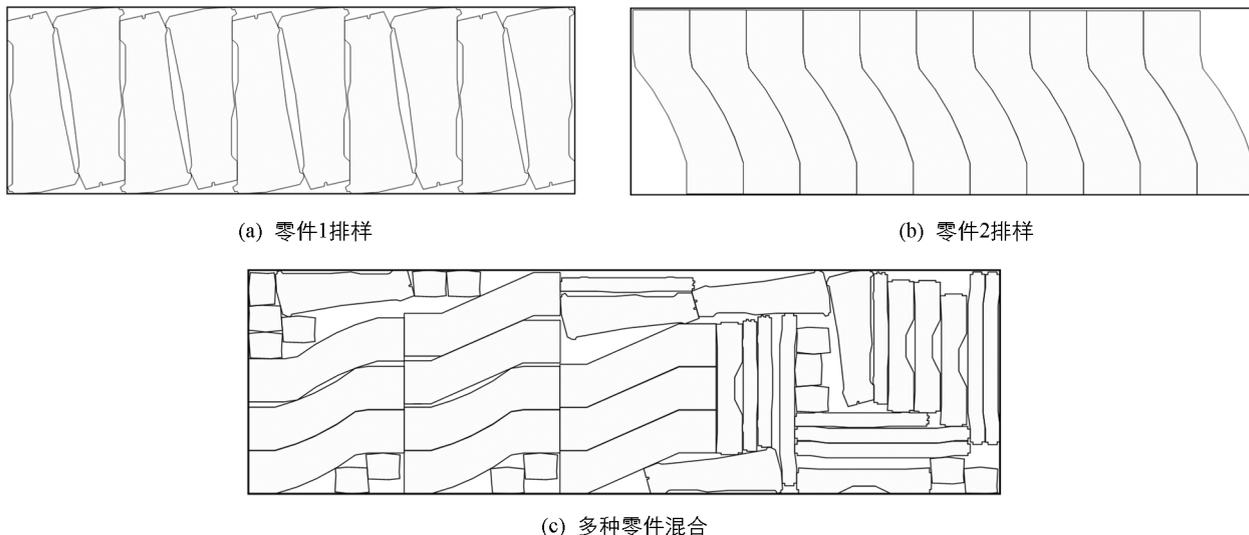


图 6 汽车零件排样

参考文献:

- [1] LIU D, TENG H. An Improved BL-Algorithm for Genetic Algorithm of the Orthogonal Packing of Rectangles [J]. European Journal of Operational Research, 1999, 112(2): 413-420.
- [2] LIU H, HE Y. Algorithm for 2D Irregular-Shaped Nesting Problem Based on the NFP Algorithm and Lowest-Gravity-Center Principle [J]. Journal of Zhejiang University-Science A, 2006, 7(4): 570-576.
- [3] OLIVEIRA J F, GOMES A M, FERREIRA J S. TOPOS - A New Constructive Algorithm for Nesting Problems [J]. OR-Spektrum, 2000, 22(2): 263-284.
- [4] 汤德佑, 周子琳. 基于临界多边形的不规则件启发式排样算法 [J]. 计算机应用, 2016, 36(9): 2540-2544.
- [5] 刘海明, 周炯, 吴忻生. 应用临界多边形方法与小生境遗传算法求解不规则排样问题 [J]. 小型微型计算机系统, 2016, 37(5): 1002-1007.
- [6] 杨卫波, 王铮, 王万良, 等. 基于实数编码量子进化算法的不规则多边形排样 [J]. 计算机集成制造系统, 2016, 22(5): 1235-1243.
- [7] 王静静, 瞿少成, 李科林. 一种基于并行交叉遗传算法的二维不规则排样问题求解 [J]. 计算机应用与软件, 2020, 37(7): 188-193.
- [8] BURKE E, HELLIER R, KENDALL G, et al. A New Bottom-Left-Fill Heuristic Algorithm for the Two-Dimensional Irregular Packing Problem [J]. Operations Research, 2006, 54(3): 587-601.
- [9] RAO Y Q, WANG P, LUO Q. Hybridizing Beam Search with Tabu Search for the Irregular Packing Problem [J]. Mathematical Problems in Engineering, 2021, 2021: 1-14.
- [10] SHALABY M A, KASHKOUSH M. A Particle Swarm Optimization Algorithm for a 2-D Irregular Strip Packing Problem [J]. American Journal of Operations Research, 2013, 3(2): 268-278.
- [11] BURKE E K, HELLIER R S R, KENDALL G, et al. Complete and Robust No-Fit Polygon Generation for the Irregular Stock Cutting Problem [J]. European Journal of Operational Research, 2007, 179(1): 27-49.
- [12] GOMES A M, OLIVEIRA J F. Solving Irregular Strip Packing Problems by Hybridising Simulated Annealing and Linear Programming [J]. European Journal of Operational Research, 2006, 171(3): 811-829.
- [13] 李建国, 张海飞, 周璐婕, 等. 基于改进遗传算法的立体车库布局对比及服务资源优化 [J]. 西南大学学报(自然科学版), 2019, 41(4): 139-148.
- [14] WANG L, PAN J, JIAO L. The Immune Algorithm [J]. Acta Electronica Sinica, 2000, 28(7): 96-101.
- [15] 杨从锐, 钱谦, 王锋, 等. 改进的自适应遗传算法在函数优化中的应用 [J]. 计算机应用研究, 2018, 35(4): 1042-1045.
- [16] BOOKSTEIN A, KULYUKIN V A, RAITA T. Generalized Hamming Distance [J]. Information Retrieval, 2002, 5(4): 353-375.
- [17] 孙二华, 胡云冰. 基于鲸鱼优化和深度学习的不平衡大数据分类算法 [J]. 西南师范大学学报(自然科学版), 2021, 46(5): 127-133.
- [18] MIRJALILI S, LEWIS A. The Whale Optimization Algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [19] ASKARZADEH A. A Novel Metaheuristic Method for Solving Constrained Engineering Optimization Problems: Crow Search Algorithm [J]. Computers & Structures, 2016, 169(6): 1-12.
- [20] MERAIHI Y, GABIS A B, RAMDANE-CHERIF A, et al. A Comprehensive Survey of Crow Search Algorithm and Its Applications [J]. Artificial Intelligence Review, 2021, 54(4): 2669-2716.
- [21] 李敬花, 樊付见, 王昊, 等. 遗传模拟退火融合算法求解工程二维排样问题 [J]. 计算机集成制造系统, 2011, 17(9): 1962-1967.