

## 考虑设施深度的过道布置问题及改进烟花算法求解方法

刘思璐, 张则强, 管超, 龚举华

引用本文:

刘思璐, 张则强, 管超, 等. 考虑设施深度的过道布置问题及改进烟花算法求解方法[J]. *控制与决策*, 2020, 35(1): 45–54.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.0720>

## 您可能感兴趣的其他文章

Articles you may be interested in

### [基于精英混沌搜索策略的交替正弦余弦算法](#)

Alternating sine cosine algorithm based on elite chaotic search strategy

*控制与决策*. 2019, 34(8): 1654–1662 <https://doi.org/10.13195/j.kzyjc.2018.0006>

### [基于改进邻域搜索策略的人工蜂群算法](#)

Artificial bee colony algorithm based on improved neighborhood search strategy

*控制与决策*. 2019, 34(5): 965–972 <https://doi.org/10.13195/j.kzyjc.2017.1506>

### [基于排序选择和精英引导的改进人工蜂群算法](#)

An improved artificial bee colony algorithm based on the ranking selection and the elite guidance

*控制与决策*. 2019, 34(4): 781–786 <https://doi.org/10.13195/j.kzyjc.2017.1334>

### [改进遗传模拟退火算法在TSP优化中的应用](#)

Application of improved genetic simulated annealing algorithm in TSP optimization

*控制与决策*. 2018, 33(2): 219–225 <https://doi.org/10.13195/j.kzyjc.2016.1666>

### [考虑安全约束的连退生产过程多目标操作优化](#)

Multi-objective operation optimization of continuous production process with safety constraints

*控制与决策*. 2018, 33(10): 1740–1746 <https://doi.org/10.13195/j.kzyjc.2017.1232>

### [求解高维优化问题的混合灰狼优化算法](#)

Hybrid grey wolf optimization algorithm for high-dimensional optimization

*控制与决策*. 2016, 31(11): 1991–1997 <https://doi.org/10.13195/j.kzyjc.2015.1183>

### [改进差分进化算法在可靠性冗余分配问题中的应用](#)

Improved differential evolution algorithm for solving reliability redundancy allocation problem

*控制与决策*. 2015, 30(5): 917–922 <https://doi.org/10.13195/j.kzyjc.2014.0647>

### [一种深度扩展记忆的仿人粒子群算法仿真分析](#)

Simulation analysis of human simulated PSO based on deep extended memory

*控制与决策*. 2015(4): 630–634 <https://doi.org/10.13195/j.kzyjc.2013.1627>

# 考虑设施深度的过道布置问题及改进烟花算法求解方法

刘思璐, 张则强<sup>†</sup>, 管 超, 龚举华

(1. 西南交通大学 机械工程学院, 成都 610031; 2. 轨道交通运维技术与装备四川省重点实验室, 成都 610031)

**摘 要:** 针对过道布置问题中忽略设施深度及物流交互点置于过道边线的问题, 结合实际布局活动对过道布置问题进行拓展, 构建考虑设施深度的过道布置问题的混合整数规划模型, 并提出一种改进烟花算法. 该算法采用 2-opt 邻域构造方法对爆炸操作进行离散化处理; 设置搜索深度并执行两点变异操作, 实现变邻域搜索; 通过贪婪选择方法更新种群以记忆精英解. 为改进算法, 引入混沌映射初始化烟花种群和设置阈值以加速算法寻优. 通过对比数学规划方法与所提出算法的求解结果, 验证了模型的有效性和所提出算法的求解优势. 最后, 通过不同算法的对比实验表明, 改进烟花算法在寻优精度、收敛效率和稳定性方面均有明显优势.

**关键词:** 过道布置问题; 烟花算法; MIP 数学模型; 设施布局

中图分类号: TH165; TP301.6

文献标志码: A

## Improved fireworks algorithm for the corridor allocation problem with facility depth

LIU Si-lu, ZHANG Ze-qiang<sup>†</sup>, GUAN Chao, GONG Ju-hua

(1. School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China; 2. Technology and Equipment of Rail Transit Operation and Maintenance Key Laboratory of Sichuan Province, Chengdu 610031, China)

**Abstract:** In view of the shortcomings in the existing research on the corridor allocation problem (CAP), in which the facility depth is always ignored and the logistics interaction points are always placed on the side line of the corridor, the CAP is extended according to the actual layout activities. A mixed integer programming model is constructed and an improved fireworks algorithm is proposed. In the improved algorithm, the explosion operation is discretized through a neighborhood construction method. The search depth and the two point mutation operation are set up to realize the variable neighborhood search. The greedy selection method is applied to update the population to memorize the elite solution. Furthermore, chaos mapping is introduced to initialize the fireworks population, and the threshold is set under the premise of high quality solution to accelerate optimization. Through comparing the results of mathematical programming with the proposed algorithm, the validity of the proposed model and the advantages of the presented algorithm are verified. Finally, the comparison experiments of different algorithms show that the improved fireworks algorithm has obvious advantages in optimization accuracy, convergence efficiency and stability.

**Keywords:** corridor allocation problem; fireworks algorithm; mixed integer programming model; facility layout

## 0 引 言

生产制造企业的设施布局问题 (facility layout problem, FLP) 是设施规划领域的一个重要问题<sup>[1]</sup>, 合理的设施布局方案能加快物料处理速度, 减少中间产品和在制品的停留时间, 降低企业运营成本<sup>[2-3]</sup>. FLP 研究按照确定的布局形式, 将生产系统中的设备、设施的位置以及与之相关的物料流、人员流、资金流和信息流进行合理组织与布置, 以达到减少企业布局成本的目的. 因此, 研究该问题的重要意义和实际应用

价值日益凸显, 受到了工业界和学术界的高度关注和广泛研究<sup>[1-4]</sup>.

过道布置问题 (corridor allocation problem, CAP)<sup>[5]</sup> 的提出进一步丰富和扩展了设施布局领域的相关研究. 该问题由 Amaral<sup>[5]</sup> 于 2012 年首次提出, 以最小化总物流成本为目标, 在两过道边线上平行无间隙地排列给定数目的设施. CAP 具有较好的物料搬运结构和效率, 不仅广泛应用于医院<sup>[6]</sup>、学校教学楼、行政大楼<sup>[7]</sup> 和购物中心等服务部门, 而且在工业

收稿日期: 2018-05-27; 修回日期: 2018-07-23.

基金项目: 国家自然科学基金项目 (51205328, 51675450); 教育部人文社会科学研究青年基金项目 (18YJC630255); 四川省科技计划项目 (2019YFG0285).

责任编辑: 陈家伟.

<sup>†</sup>通讯作者. E-mail: zzq\_22@163.com.

制造领域具有重要作用,如低碳车间布局<sup>[8]</sup>、半导体生产线布局<sup>[9]</sup>、微芯片集成电路板布局、自动导引车(automated guided vehicle, AGV)行径下车间的布局<sup>[10]</sup>等.因此,该问题自提出便受到广泛关注,已成为设施布局问题领域的新热点<sup>[5,11-16]</sup>.

CAP属于典型的组合优化问题,具有NP-hard属性,其精确求解难度随问题规模和约束条件的增加呈指数级增长.由于CAP重要的学术研究价值和实际应用前景,人们对其展开了广泛的研究.文献[6]首次提出CAP问题并建立了该问题的混合整数规划(mixed integer programming, MIP)模型,随后应用3种启发式方法测试了该问题,验证了模型的正确性和算法的有效性. Ghosh等<sup>[11]</sup>采用改进遗传算法和分散搜索算法对CAP不同规模的算例进行了测试,试验结果表明两种算法均可获得高质量近优解,其中分散搜索算法在运行效率上较遗传算法更具优势. Ahonen等<sup>[12]</sup>应用改进模拟退火算法和禁忌搜索算法对该问题进行测试,结果显示改进模拟退火算法在求解质量和求解效率方面性能更优. Kalita等<sup>[13]</sup>进一步考虑了通道长度目标,将过道布置问题扩展为双目标优化问题,并采用遗传算法测试了一系列基准算例.毛丽丽等<sup>[14]</sup>构建了考虑通道宽度和总流量入口的混合整数规划模型,并提出了一种改进分散搜索算法. Kalita等<sup>[15]</sup>采用一种结合局部搜索技术的改进遗传算法对双目标过道布置问题进行求解.为了进一步解决生产中多层空间的实际情况,管超等<sup>[16]</sup>提出了双层过道布置问题,建立了该问题的数学模型,并采用一种基于C2Opt邻域搜索的启发式算法进行求解.上述文献对CAP问题进行了更为深入的探讨,所采用的求解方法也都正确有效,但在寻优精度和求解速度上有待进一步提高.

设施深度是布局设计中的重要影响因素,并且考虑设施深度更加符合实际生产活动.参考现有文献,尚未发现考虑设施深度的过道布置问题研究.在实际生产制造领域或服务部门中,设施布置往往如图1所示,在优化设施间沿过道边线方向物流路径的同时,需要考虑设施间物流交互点对目标函数的影响.因此,本文提出一种扩展过道布置问题(extend corridor allocation problem, ECAP).

烟花算法(fireworks algorithm, FWA)<sup>[17]</sup>是一种具有爆炸搜索机制的全局优化算法.该算法具有结构简单新颖、爆发性强、多样性好<sup>[18]</sup>等优点.目前,烟花算法已广泛地应用于求解装配顺序规划<sup>[19]</sup>、车间调度、多维背包<sup>[20]</sup>等优化问题.尽管烟花算法在离散

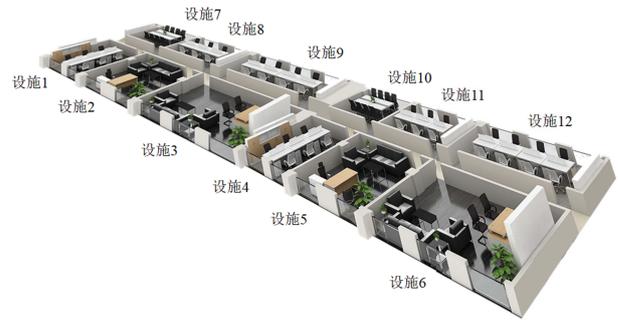


图1 过道布置问题

化问题上取得了良好的求解结果,但尚未发现有关应用烟花算法求解设施布局问题的公开报道.因此,本文采用烟花算法来求解过道布置问题.

鉴于实际生产中考虑设施深度和物流交互点位置的变化更加符合实际情况,本文考虑设施深度以及物流交互点的变化对过道布置的影响,提出ECAP,并构建该问题的MIP数学模型.为验证模型的正确性并给所提出算法提供参考依据,先用LINGO软件进行精确值求解;鉴于精确方法仅能求解小规模问题,难以应对中大规模问题,本文提出一种改进烟花算法(improved fireworks algorithm, IFWA)求解该问题.本文对烟花算法进行离散化设计,采用基于设施编码序列的整数编码方式,并重新定义了烟花爆炸过程.在局部搜索阶段,采用2-opt进行邻域搜索,设置搜索深度,不断搜寻最优解并更新记忆;在全局寻优阶段,采用两点变异跳出局部最优.为了对烟花算法进行改进,一方面,本文引入混沌变量来增强种群分散性以避免算法陷入局部最优,增加初始烟花群的多样性;另一方面,设置阈值以减少多余迭代过程.为了评估IFWA在ECAP新模型中的性能,考虑48个规模的标准算例,并将所获得的结果与LINGO精确方法求解值进行对比.最后,为进一步说明所提出算法的先进性,应用IFWA求解原CAP问题,并与其他相关文献所得结果进行比较,结果表明在求解复杂优化问题时,IFWA在寻优精度和求解效率方面更胜一筹.本文所得数值结果可为评估CAP的未来研究提供参考.

## 1 考虑设施深度的CAP数学模型

### 1.1 问题描述

在实际工业制造和服务部门中,设施深度是设施布局中需要考虑的重要影响因素.针对现有CAP均未考虑设施深度对物流成本的影响,本文提出一种ECAP问题. ECAP与单行布局问题(single row facility layout problem, SRFLP)<sup>[21]</sup>和并行行排序问题(parallel row ordering problem, PROP)<sup>[22]</sup>相比,在数学建模、问题求解等方面更加困难.其原因是SRFLP

不考虑异行间的物流交互活动;PROP 预先固定布置在不同行中的设施集合,问题难度大大简化. 另外,ECAP与双行布局问题(double row facility layout problem, DRFLP)<sup>[23]</sup>类似,主要区别在于ECAP中不同行设施的布置起点相同且相邻设施间无间隙.

### 1.2 SRFLP模型

在SRFLP中,通常不同设施的尺寸不同且设施均排列在同一直线上.图2所示为单行布局问题示意.图2中AGV小车在同行设施B、D间移动,设施间物料搬运长度记为 $d_{ij}$ ,设施 $i$ 的宽度记为 $l_i$ .

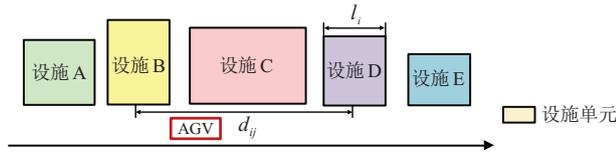


图2 单行布局问题示意

SRFLP数学模型的假设如下:设施形状均为矩形;设施的摆放方位、各设施的深度以及单位距离物流交互成本均一致;布局不受场地大小及其他条件限制.已知该数学模型如下:

$$\min \sum_{i,j \in N; i < j} c_{ij} d_{ij}. \quad (1)$$

s.t.

$$d_{ij} \geq \sum_{k < i} l_k \alpha_{ki} + \sum_{k > i} l_k (1 - \alpha_{ik}) - \sum_{k < j} l_k \alpha_{kj} - \sum_{k > j} l_k (1 - \alpha_{jk}) + (l_i - l_j)/2, \quad 1 \leq i \leq j \leq n; \quad (2)$$

$$d_{ij} \geq -\sum_{k < i} l_k \alpha_{ki} - \sum_{k > i} l_k (1 - \alpha_{ik}) + \sum_{k < j} l_k \alpha_{kj} + \sum_{k > j} l_k (1 - \alpha_{jk}) + (l_j - l_i)/2, \quad 1 \leq i \leq j \leq n; \quad (3)$$

$$\alpha \in H_n, \alpha \in R_+^{n(n-1)/2}; \quad (4)$$

$$H_n = \begin{cases} \alpha_{ij} + \alpha_{jk} - \alpha_{ik} \leq 1, 1 \leq i \leq j \leq n; \\ -\alpha_{ij} - \alpha_{jk} + \alpha_{ik} \leq 0, 1 \leq i \leq j \leq n; \end{cases} \quad (5)$$

$$d \in D_n, d \in R_+^{n(n-1)/2}; \quad (6)$$

$$D_n = \{d_{ij} \geq (l_i + l_j)/2, 1 \leq i \leq j \leq n\}; \quad (7)$$

$$\alpha_{ij} \in \{0, 1\}, 1 \leq i \leq j \leq n. \quad (8)$$

其中: $N$ 为问题规模; $c_{ij}$ 为设施 $i$ 、 $j$ 间单位距离物流交互成本; $\alpha_{ij}$ 为二进制变量,若设施 $i$ 位于设施 $j$ 的左侧,则 $\alpha_{ij} = 1$ ,否则 $\alpha_{ij} = 0$ ; $d_{ij}$ 为设施 $i$ 与设施 $j$ 中心

之间的间距.目标(1)表示最小化总物料搬运用,式(2)和(3)确定设施 $i$ 、 $j$ 之间的距离,式(4)和(5)约束两个设施之间的最小距离等于它们的长度之和的一半,式(6)~(8)保证 $\alpha$ 是线性排序的关联向量.

SRFLP中连续变量 $d_{ij}$ 的数量为 $n(n-1)/2$ ,其问题的组合情况为 $n!$ .虽然该问题数学模型不复杂,但是具有重大的研究价值和实际意义<sup>[21,24-25]</sup>.CAP与SRFLP相比,约束条件更多,问题难度更大,模型更加复杂.

### 1.3 ECAP模型

ECAP示意如图3所示.假设设施形状为规则矩形,上下两行设施均从过道最左边同一点开始布置,且不受场地大小及其他条件限制;已知各设施靠过道边线的宽度、深度及设施间的单位距离物流交互成本;各设施的物流交互点均位于设施几何中心,并且同行相邻的两设施间无间隙;忽略过道宽度.图3中AGV小车在异行设施A与设施B间移动,设施间物料搬运长度采用曼哈顿距离计量,其交互距离用虚线表示,记为

$$(x_i - x_j) + \frac{1}{2}(h_i + h_j).$$

其中: $x_i$ 、 $x_j$ 分别表示位于第1行的设施 $i$ 和第2行的设施 $j$ 到左端点的距离, $h_i$ 、 $h_j$ 分别表示设施 $i$ 、 $j$ 的深度.

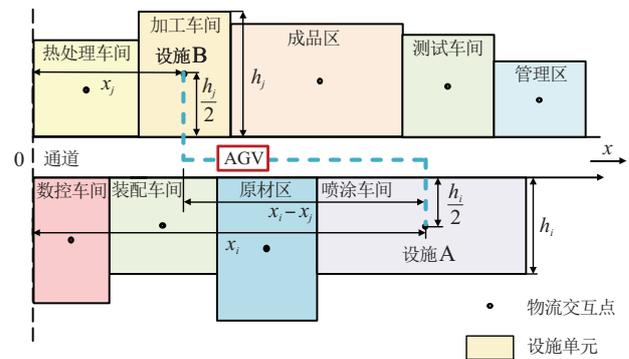


图3 异行设施间的物流交互距离

### 1.4 ECAP数学模型

定义如下参数和变量来描述该数学模型.

$n$ :问题规模;

$I$ :设施集合, $I = \{1, 2, \dots, n\}$ ;

$i, j$ :设施编号, $i, j \in I$ ;

$c_{ij}$ :设施 $i$ 、 $j$ 之间的单位距离物流交互成本,且

$$c_{ij} = c_{ji};$$

$l_i$ :设施 $i$ 的宽度;

$h_i$ :设施 $i$ 的深度;

$d_{ij}$ : 设施*i*和设施*j*的物流交互中心点在*X*轴方向的距离;

$\alpha_{ij}$ : 二进制变量,若设施*i*、*j*分配在同一行,且设施*i*布置在设施*j*的左边,则 $\alpha_{ij} = 1$ ,否则 $\alpha_{ij} = 0$ .

数学模型为

$$\min \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} \cdot \left( d_{ij} + \frac{h_i + h_j}{2} \right). \quad (9)$$

s.t.

$$d_{ij} \geq \frac{(l_i - l_j)}{2} + \sum_{k=1; k \neq i}^n l_k \alpha_{ki} - \sum_{k=1; k \neq j}^n l_k \alpha_{kj}, \quad (10)$$

$$1 \leq i < j \leq n;$$

$$d_{ij} \geq \frac{(l_j - l_i)}{2} - \sum_{k=1; k \neq i}^n l_k \alpha_{ki} + \sum_{k=1; k \neq j}^n l_k \alpha_{kj}, \quad (11)$$

$$1 \leq i < j \leq n;$$

$$d_{ij} - \frac{(l_i + l_j)}{2} \cdot \alpha_{ij} - \frac{(l_i + l_j)}{2} \cdot \alpha_{ji} \geq 0, \quad (12)$$

$$1 \leq i < j \leq n;$$

$$\alpha_{ij} + \alpha_{ik} + \alpha_{jk} + \alpha_{ji} + \alpha_{ki} + \alpha_{kj} \geq 1, \quad (13)$$

$$1 \leq i < j < k \leq n;$$

$$-\alpha_{ij} + \alpha_{ik} + \alpha_{jk} - \alpha_{ji} + \alpha_{ki} + \alpha_{kj} \leq 1, \quad (14)$$

$$i, j, k \in \mathbf{I}, i < j, k \neq i, k \neq j;$$

$$-\alpha_{ij} + \alpha_{ik} - \alpha_{jk} + \alpha_{ji} - \alpha_{ki} + \alpha_{kj} \leq 1, \quad (15)$$

$$i, j, k \in \mathbf{I}, i < j, k < j, i \neq k;$$

$$\alpha_{ij} \in \{0, 1\}, 1 \leq i, j \leq n, i \neq j. \quad (16)$$

其中:目标(9)为最小化总物流成本;式(10)和(11)用于确定各设施的绝对距离,即物流交互中心点在*X*轴方向的长度;式(12)用于防止同行设施间出现相互重叠的现象;式(13)~(15)根据狄利克雷抽屉原理确定决策变量 $\alpha_{ij}$ ;式(16)限定决策变量 $\alpha_{ij}$ 的取值范围.

关于式(13)~(15)的证明:给定任意设施*i*、*j*、*k* ∈ **I**,且假定条件*i* < *j* < *k*,由抽屉原理1可知,3个设施中至少有两个设施会布置在相同行,即 $(\alpha_{ij} + \alpha_{ji})$ 、 $(\alpha_{ik} + \alpha_{ki})$ 、 $(\alpha_{jk} + \alpha_{kj})$ 中至少一个取值为1,故式(13)成立;当设施*i*、*j*、*k*布置在同一行时, $(\alpha_{ij} + \alpha_{ji})$ 、 $(\alpha_{ik} + \alpha_{ki})$ 、 $(\alpha_{jk} + \alpha_{kj})$ 取值均为1,故式(14)成立;对于任意设施 $1 \leq i < j \leq n$ ,且 $(\alpha_{ij} - \alpha_{ji}) \in \{-1, 0, 1\}$ ,当 $(\alpha_{ik} - \alpha_{ki})$ 、 $(\alpha_{jk} - \alpha_{kj})$ 取值均为1时,设施*i*、*j*、*k*必布置在相同行,此时 $(\alpha_{ij} - \alpha_{ji})$ 取值必为1,故式(15)成立.

## 2 求解ECAP问题的IFWA算法

烟花算法作为非生物群体智能优化方法的一种,具有爆发性、多样性、简单性和分布并行性.鉴于烟花算法在解决连续性问题中具有收敛速度快、精度高等优点,本文将应用于离散优化问题,主要难点在于编码过程的离散化处理和相关操作的离散构造.ECAP作为典型的NP-hard问题,具有组合优化特性,随着问题规模的增加,可行解组合情况呈指数级增长.特别是求解大规模问题时,精确方法遍历的解空间范围太大,无法在合理时间内求得问题结果.因此,本文提出一种IFWA,以期快速求得高质量近优解.

本文对烟花算法进行离散化处理时,保留了爆炸算子、变异算子和选择策略.为了提高算法的搜索速度,增强搜索能力,引入混沌映射方法初始化烟花种群.由于局部搜索是解决设施布局问题最有效的方法之一<sup>[26]</sup>,结合ECAP特点,本文对爆炸算子采用2-opt邻域构造方法进行离散;对于变异算子,采用两点变异方法以实现变邻域搜索;针对选择策略,采用贪婪选择方法更新种群,选择并记忆适应度最优的目标值和解序列.此外,为提高算法求解速度,在保证高质量解的前提下,设置阈值以减少不必要的迭代过程.

### 2.1 初始烟花群产生方法

针对ECAP问题的组合优化特性,本文采用一种基于设施编码序列的整数编码方式.定义第*i*个烟花个体为 $X_i = [x_1, x_2, \dots, x_n]$ .其中: $x_j$ 表示布置在第*j*个位置的设施编号;用 $\mathbf{P} = [X_1, X_2, \dots, X_{\text{noP}}]$ 表示烟花种群,noP代表种群中的个体数目.对于随机排列集合 $\{x_1, x_2, \dots, x_n\}$ 中的元素,每种排列顺序代表一种布局方案.以设施数目*n* = 9的ECAP为例,假设该问题的一个可行解序列为 $X_1 = [5, 3, 1, 9, 8, 2, 6, 4, 7]$ ,若设布置在第1行的设施数目为 $n_u = 5$ ,则可行解序列 $X_1$ 表示布置在第1行和第2行中的设施序列分别为 $[5, 3, 1, 9, 8]$ 和 $[2, 6, 4, 7]$ ,该序列的解码与编码过程如图4所示.

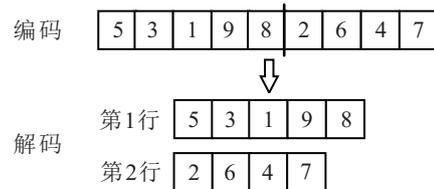


图4 解的编码和解码

为了增加烟花群的多样性以提高算法的寻优精度和搜索范围,结合混沌映射具有遍历性、规律性的

特点, 本文以混沌映射方法作为初始解产生策略, 采用通过严格数学推理的 Tent 混沌特性公式 (9)<sup>[27]</sup>, 使有限烟花个体遍历整个搜索空间. 所述混沌映射产生烟花初始群的具体流程如图 5 所示. 图 5 中  $R$  是随机产生的  $0 \sim 1$  之间的  $n$  维向量, 利用式 (9) 对  $R$  进行更新, 并采用从小到大的排列顺序确定每一维元素的位置向量  $\text{index}$ . 有

$$r_{i+1} = \begin{cases} 2r_i, & 0 \leq r_i \leq \frac{1}{2}; \\ 2(1 - r_i), & \frac{1}{2} \leq r_i \leq 1. \end{cases} \quad (17)$$

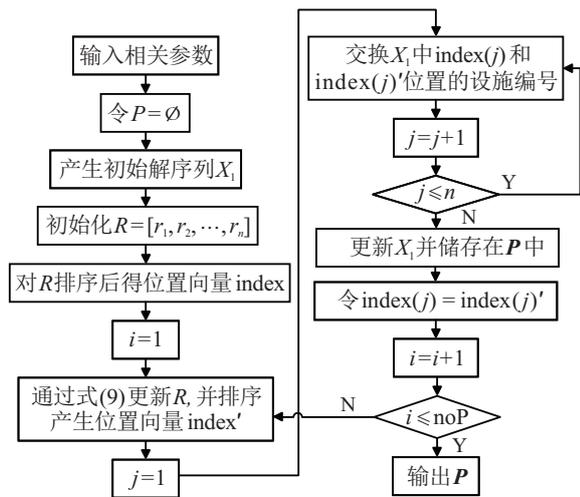


图 5 混沌方法初始烟花群流程

### 2.2 烟花算法操作的离散设计

传统的烟花算法主要包括爆炸算子、变异算子、映射规则和选择策略 4 个部分. 本文对烟花算法进行离散化处理时, 针对 ECAP 可行解的离散性、搜索空间复杂性, 重新定义烟花算法的全局搜索和局部搜索过程, 并对相应操作进行离散构造.

#### 2.2.1 爆炸算子

局部搜索算法是基于贪婪思想利用邻域函数进行搜索的. 本文中为实现算法的局部搜索, 采用 2-opt 邻域构造方法产生爆炸火花. 2-opt 是一种基于“交换”的启发式思想, 利用互换操作 (SWAP), 通过随机交换设施编码序列中两个不同设施编号的位置来实现局部搜索. 搜索过程中, 通过比较新解和最优解的目标函数值进行更新记忆. 本文设置搜索深度  $\text{dep}$ , 定义  $\text{dep} = \alpha \times n$ ,  $\alpha$  为搜索深度系数. 通过对每个烟花的邻域空间进行  $\text{dep}$  次随机抽样, 可以实现烟花个体的更新并记忆全局最优解.

#### 2.2.2 变异算子和终止准则

全局寻优阶段, 当搜索深度为零时, 对烟花个体采用两点变异操作, 通过随机搜索全局空间, 寻找更优解并不断更新全局最优解.

为提高算法寻优效率, 在保证求解质量前提下, 本文设置阈值  $\text{max\_step}$  以减少求解过程中的不必要迭代. 采用计数器  $p\_out$  记录爆炸过程中所得全局最优解  $\text{fit}_{\text{best}}$  保持不变的次数, 若  $p\_out > \text{max\_step}$ , 则算法提前终止并返回最优解  $\text{fit}_{\text{best}}$ .

### 2.3 算法流程

IFWA 的具体步骤如下.

Step 1: 算法初始化, 输入烟花数  $\text{noP}$ , 设施规模  $n$ , 最大循环次数  $\text{max\_iter}$ , 终止阈值  $\text{max\_step}$ , 搜索深度系数  $\alpha$ .

Step 2: 通过混沌映射策略初始化烟花群, 并挑选其中最优结果作为当前全局最优适应度值  $\text{fit}_{\text{best}}$  和全局最优布置方案  $X_{\text{best}}$ , 其中  $\text{fit}$  为适应度值.

Step 3: 令  $p\_out = 0$ .

Step 4: 令  $i = 1$ .

Step 5: 令  $j = 1$ .

Step 6: 令  $k = \text{dep}$ .

Step 7: 对初始烟花群中第  $j$  个烟花执行 2-opt 爆炸操作, 产生新的火花, 其对应的新解为  $\text{fit}'(j)$ , 新序列为  $X'(j)$ .

Step 8: 若  $\text{fit}'(j) < \text{fit}(j)$ , 则  $\text{fit}(j) = \text{fit}'(j)$ ,  $X(j) = X'(j)$ , 并进一步更新全局最优; 若  $\text{fit}(j) = \text{fit}_{\text{best}}$ , 则  $\text{fit}_{\text{best}} = \text{fit}(j)$ ,  $X_{\text{best}} = X(j)$ ; 否则保持  $\text{fit}(j)$ 、 $X(j)$  不变.

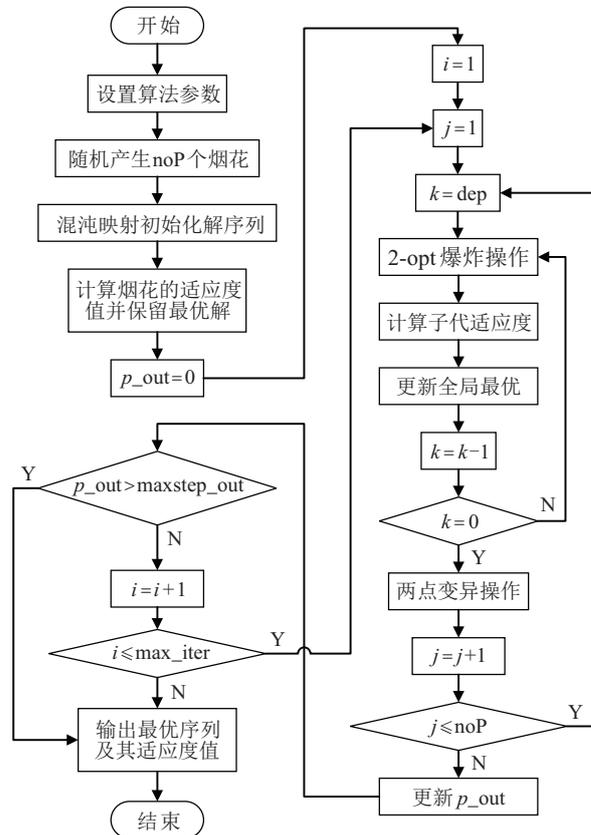


图 6 IFWA 算法流程

Step 9: 令  $k = k - 1$ , 当  $k = 0$  时执行变异操作, 保存变异火花; 否则执行 Step 7.

Step 10: 令  $j = j + 1$ , 若  $j \leq \text{noP}$ , 则执行 Step 6 ~ Step 9; 否则执行 Step 11.

Step 11: 若  $p\_out(i) > \text{max\_step}$ , 则算法终止, 并返回最优解.

Step 12: 令  $i = i + 1$ , 若  $i \leq \text{max\_iter}$ , 则执行 Step 5; 否则输出全局最优适应度值  $\text{fit}_{\text{best}}$  和全局最优布置方案  $X_{\text{best}}$ .

IFWA 流程如图 6 所示.

### 3 实验结果与分析

计算实验采用的计算机硬件配置为 Intel(R) Core(TM) i5-3230 M@2.6 GHz, 4 GB 内存, 在 Windows 10 操作系统下运行相应程序. 鉴于 ECAP 问题尚无测试运算结果, 本文采用 LINGO 11.0 优化器测试相关算例, 为所提出的算法提供依据. 随后, 运用 IFWA 求解原 CAP 问题并与文献 [5] 的精确方法和启发式方法 (CAP-Heuristic)、文献 [11] 中的遗传算法 (CAP-GA) 和分散搜索算法 (CAP-SS)、文献 [12] 中改进的模拟退火算法 (CAP\_SA) 和禁忌搜索算法

表 1 IFWA 算法和 LINGO 算法求解 CAP 的计算结果

序号	算例	规模	LINGO		IFWA				CPU time/s
			目标值	CPU time/s	目标值				
					min	max	avg	SD	
1	S9	9	2228.5	115	2228.5	2228.5	2228.5	0	0.21
2	S9H	9	3809	1057	3809	3812	3809.3	0.9	0.23
3	S10	10	2552	597	2552	2556	2553.4	2.2	0.29
4	S11	11	6088	4504	6088	6189	6139.7	34	0.36
5	Am12a	12	2710	14708	2710	2725	2717.5	5.7	0.37
6	Am12b	12	3030.5	12646	3030.5	3093.5	3072	16	0.39
7	Am13a	13	4198	1800	4175	4178	4176.5	1.5	0.41
8	Am13b	13	4952	1800	4921	4946	4936.8	9.7	0.39
9	Am15	15	5221	1800	5146	5303	5184.8	46	0.47
10	Am17	17	8338.5	1800	7186.5	7233.5	7207.2	14	0.63
11	Am18	18	9963	1800	8130	8187	8175	33	0.67
12	H20	20	13634.5	1800	11271.5	11438.5	11368.9	73	0.77
13	N25_01	25	4151	1800	3053	3062	3957.5	4.5	106.23
14	N25_02	25	35510	1800	25466	25472	25467.2	2.4	163.87
15	N25_03	25	21417	1800	16194	16228	16211.6	8.628	153.42
16	N25_04	25	48206	1800	33247	33276	33263.9	8.749	180.71
17	N25_05	25	13759.5	1800	10616.5	10626.5	10618.5	4	174.46
18	N30_01	30	12924	1800	5224	5224	5224	0	197.75
19	N30_02	30	26629	1800	13868	13869	13868.1	0.3	236.14
20	N30_03	30	36675.5	1800	29611.5	29615.5	29612.2	1.418	217.06
21	N30_04	30	51237.5	1800	37542.5	37570.5	37551.9	8.98	236.53
22	N30_05	30	324356	1800	74517.5	74555.5	74543.2	14.32	250.76
23	P3	33	—	1800	45277	45299	45292.1	10.67	276.14
24	P6	35	—	1800	43160.5	43180.5	43169.4	7.031	337.04
25	ste_36_01	36	—	1800	7591	7591	7591	0	299.59
26	ste_36_02	36	—	1800	144157	2144782	144582.5	269.9	494.45
27	ste_36_03	36	—	1800	74467.5	74847.5	74586.7	107.5	468.43
28	ste_36_04	36	—	1800	76658	76848	76796.4	58.27	466.97
29	ste_36_05	36	—	1800	67027.5	67336.5	67089.6	89.72	495.43
30	N40_01	40	—	1800	65498	65521	65514.1	7.622	534.08
31	N40_02	40	—	1800	60377	60382	60378.5	1.803	492.21
32	N40_03	40	—	1800	49030	49039	49032.9	4.253	797.51
33	N40_04	40	—	1800	47916	47916	47922.2	6.896	504.94
34	N40_05	40	—	1800	63530.5	63547.5	63540.5	5.385	526.42
35	sko_42_01	42	—	1800	15046	15053	15050.5	3.429	611.82
36	sko_42_02	42	—	1800	130800.5	130832.5	130814.5	9.592	868.99
37	sko_42_03	42	—	1800	104959	104989	104973.1	9.181	11131.12
38	sko_42_04	42	—	1800	82716	82852	82756.2	43.268	809.05
39	sko_42_05	42	—	1800	149751	149790	149771.8	17.308	877.44
40	sko_49_01	49	—	1800	23591	23666	23614	18.937	914.15
41	sko_49_02	49	—	1800	244464	244540	244500.3	22.1	1796.796
42	sko_49_03	49	—	1800	189764	244501	244500.3	32.666	1520.695
43	sko_49_04	49	—	1800	138145.5	138189.5	138264.6	15.984	1550.69
44	sko_49_05	49	—	1800	388070.5	388139.5	388106.8	19.329	1417.57
45	sko_56_01	56	—	1800	36237	36251	36238.6	4.176	1395.48
46	AKV_60_01	60	—	1800	742765.65	742967.7	742900.7	72.444	3190.31
47	AKV_70_01	70	—	1800	871819.5	871819.5	871628.9	89.87	4828.55
48	AKV_80_01	80	—	1800	1163441	1163646	1163512.3	65.988	6626.29

(CAP\_TS)的计算结果进行对比,以验证本文所提出算法求解小规模问题的有效性. 针对中大规模问题,对比文献[5]和文献[11]的计算结果,验证原CAP求解中大规模问题时的性能.

### 3.1 ECAP问题验证

为验证1.3节ECAP模型的正确性并给本文所提出算法运算结果提供对比依据,针对所选测试算例,运用LINGO软件对48个设施规模为9~80的基准算例进行精确求解. 测试算例的部分数据选自文献[14].

经测试,对于设施规模小于12的测试算例,LINGO优化器可在合理时间内求得精确解. 通过验证所得最优解序列,表明了本文模型的正确性. 测试问题S9、S9H、S10、S11、Am12a、Am12b的运算时间分别为115s、1057s、597s、4504s、14708s、12646s. 随着问题规模的增加,精确方法求解问题耗时大幅度增加. 为给所提出IFWA的求解结果提供参考,针对规模大于12的测试算例,设置其在LINGO的运算时间为1800s,将所得结果列于表1的第4、第5列. 针对问题规模大于30的测试问题,LINGO软件在运算1800s后仍无法求得计算结果,故用“—”表示.

兼顾求解效率,本文采用IFWA求解ECAP以验证离散化算法的求解性能. 上述算法的实验程序在Matlab R2012a软件开发环境下运行. 应用设施规模为9~80的48个不同规模的算例进行实验,兼顾求解效率和求解速度. 经大量实验测试,  $n_u$  取值下限  $T_1 = \text{floor}(n/2) - 3$ ,  $n_u$  取值上限  $T_2 = \text{floor}(n/2)$ ,  $n_u = [T_1, T_2]$ , 其余算法参数设置如表2所示.

表2 算法参数设置

规模	IFWA			
	noP	max_iter	max_step	$\alpha$
$n \leq 20$	4	16	9	5
$20 < n \leq 40$	8	300	100	40
$40 < n \leq 80$	10	400	100	50

针对每个测试算例,算法测试求解10次,共计480个计算结果. 经过整理和统计,将IFWA的目标函数值的最小值、最大值、平均值、标准差SD和CPU运算时间分别列于表1的第6~第10列.

通过对比表1的数据可知: 针对所选48个CAP测试问题,IFWA在求解效率上明显优于LINGO求解器. 对于所选测试问题Am12b,IFWA求得了与LINGO软件相同的最优解3030.5,但IFWA平均运算时间仅为0.39s,是LINGO运算时间的  $3.08 \times 10^{-5}$

倍. 对于所选问题规模为9~30的22个基准算例,所提IFWA仅在250.76s的运算时间内就求得比LINGO运算1800s更好的近似解,由此说明,所提IFWA的求解效率优异.

为验证算法稳定性,根据IFWA求得的220个运算结果(9~30规模),计算其与LINGO或目前求得的最优解值的求解偏差gap,其中

$$\text{gap}(\%) = \left| \frac{f - f_{\min}}{f_{\min}} \right| \times 100\%$$

并绘制如图7所示的求解偏差箱形图以显示求得数据的离散情况. 图7中:上下框线、⊙、点划线分别对应该组数据的上下四分位、中位值、平均值;上边缘(下边缘)分别表示最大值(最小值);“+”表示不在正常值分布区间的异常值. 通过分析箱形图可以看出: 针对12个小规模测试问题,箱盒长度均较小,偏差的最大值仅为0.03%; 针对10个中规模测试问题,箱盒长度几乎为零,其上下四分位、中位值、平均值等几乎都重合于0值; 而异异常值非常少. 结果显示所得数据的离散度较小,从而表明所提算法具有优良的寻优稳定性.

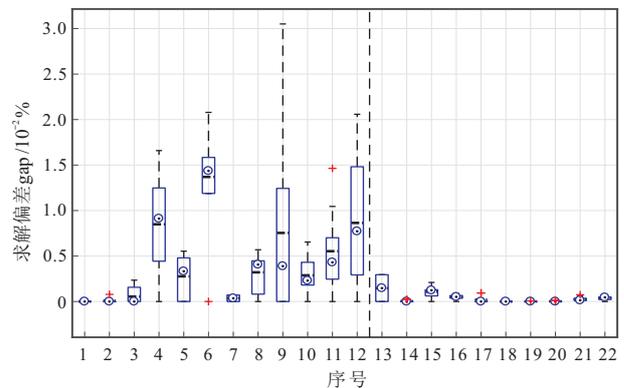


图7 求解偏差箱形图

### 3.2 小规模算例

为说明IFWA在求解小规模问题的有效性,对问题规模小于等于15的算例进行测试. 采用IFWA求解原CAP问题,每个测试问题均求解10次,将所得结果与CAP-Heuristic、CAP-GA、CAP-SS、CAP\_SA和CAP\_TS作比较. 小规模CAP问题各算法的计算结果如表3所示. 其中:加粗数据表示所得结果的最优目标函数值,下划线数据表示所得最短运算时间,“\*”表示文献[6]中所求精确算法的最优目标函数值.

通过对表3分析可知,对于小规模算例,6种算法均能求得与精确方法相同的最优解,但IFWA的运算时间更短. 这表明,在小规模问题中,6种算法的寻优质量都很优异,但IFWA的求解效率更加优异.

表3 小规模CAP问题各算法的计算结果

序号	算例	n	CAP-Heuristic		CAP-GA		CAP-SS		CAP-SA		CAP-TS		IFWA	
			f	CPU time/s	f	CPU time/s	f	CPU time/s	f	CPU time/s	f	CPU time/s	f	CPU time/s
1	S9	9	<b>1181.5<sup>+</sup></b>	18.42	<b>1181.5</b>	17.61	<b>1181.5</b>	<u>0.19</u>	<b>1181.5</b>	<b>4.05</b>	<b>1181.5</b>	<b>1.14</b>	<b>1181.5</b>	<u>0.19</u>
2	S9H	9	<b>2294.5<sup>+</sup></b>	1145.94	<b>2294.5</b>	17.54	<b>2294.5</b>	0.24	<b>2294.5</b>	<b>3.46</b>	<b>2294.5</b>	<b>1.15</b>	<b>2294.5</b>	<u>0.21</u>
3	S10	10	<b>1374.5<sup>+</sup></b>	62.75	<b>1374.5</b>	21.96	<b>1374.5</b>	0.27	<b>1374.5</b>	<b>4.88</b>	<b>1374.5</b>	<b>1.58</b>	<b>1374.5</b>	<u>0.22</u>
4	S11	11	<b>1374.5<sup>+</sup></b>	496.22	<b>3439.5</b>	29.21	<b>3439.5</b>	0.53	<b>3439.5</b>	<b>5.97</b>	<b>3439.5</b>	<b>2.02</b>	<b>3439.5</b>	<u>0.29</u>
5	Am12a	12	<b>1529.0<sup>+</sup></b>	1869.31	<b>1529.0</b>	40.2	<b>1529.0</b>	0.9	<b>1529.0</b>	<b>7.36</b>	<b>1529.0</b>	<b>2.43</b>	<b>1529.0</b>	<u>0.3</u>
6	Am12b	12	<b>1609.5<sup>+</sup></b>	1412.06	<b>1609.5</b>	38.31	<b>1609.5</b>	0.8	<b>1609.5</b>	<b>7.28</b>	<b>1609.5</b>	<b>2.43</b>	<b>1609.5</b>	<u>0.33</u>
7	Am13a	13	<b>2467.5<sup>+</sup></b>	10298.01	<b>2467.5</b>	50.3	<b>2467.5</b>	1.5	<b>2467.5</b>	<b>8.55</b>	<b>2467.5</b>	<b>2.99</b>	<b>2467.5</b>	<u>0.32</u>
8	Am13b	13	<b>2870.0<sup>+</sup></b>	5144.13	<b>2870.0</b>	53.76	<b>2870.0</b>	1.35	<b>2870.0</b>	<b>8.84</b>	<b>2870.0</b>	<b>3.01</b>	<b>2870.0</b>	<u>0.35</u>

3.3 中大规模算例

为进一步验证本文算法的先进性,应用IFWA的中大规模问题进行求解.算法运行10次,算法参数设置见表2.将所得结果与CAP-Heuristic、CAP-GA、CAP-SS的运算结果进行对比,IFWA算法与上述算

法求解原CAP的计算结果如表4所示.表4中:“+”和“++”分别表示两种启发式算法的近优解,加粗数据表示所得结果的最优目标函数值,下划线数据表示所得的最短运算时间.

表4 IFWA算法和其他算法求解CAP的计算结果

序号	算例	规模	目标值				CPU time/s			
			CAP-Heuristic	CAP-GA	CAP-SS	IFWA	CAP-Heuristic	CAP-GA	CAP-SS	IFWA
1	N30_01	30	<b>3195.0<sup>++</sup></b>	<b>4115</b>	<b>4115</b>	<b>4115</b>	6070.5	1506.69	<u>118.68</u>	128.19
2	N30_02	30	<b>10779.5<sup>+</sup></b>	<b>10779.5</b>	<b>10779.5</b>	<b>10779.5</b>	7412.6	1565.07	<u>150.06</u>	185.43
3	N30_03	30	<b>22702.0<sup>+</sup></b>	<b>22702.0</b>	<b>22702.0</b>	<b>22702.0</b>	7902.4	1611.19	163.1	<u>161.5</u>
4	N30_04	30	<b>28401.5<sup>+</sup></b>	<b>28401.5</b>	<b>28401.5</b>	<b>28401.5</b>	8270.7	1628	189.2	<u>186.07</u>
5	N30_05	30	<b>57400.0<sup>+</sup></b>	<b>57400.0</b>	<b>57400.0</b>	<b>57400.0</b>	8562.3	1672.68	188.97	<u>165.15</u>
6	ste_36_01	36	—	<b>4966</b>	<b>4966</b>	<b>4966</b>	—	3475.61	423.74	<u>194.5</u>
7	ste_36_02	36	—	88262	87489	<b>87430</b>	—	3237.71	721.35	<u>390.96</u>
8	ste_36_03	36	—	50696.5	50127.5	<b>50101</b>	—	3279.4	674.71	<u>331.7</u>
9	ste_36_04	36	—	46890.5	46824.5	<b>46737.5</b>	—	3446.11	504.18	<u>385.84</u>
10	ste_36_05	36	—	44783.5	<b>44488.5</b>	<b>44488.5</b>	—	3407.52	593.57	<u>259.2</u>
11	sko_42_01	42	—	12731	<b>12731</b>	<b>12731</b>	—	7174.64	1249.04	<u>321.29</u>
12	sko_42_02	42	—	108049.5	<b>108020.5</b>	<b>108020.5</b>	—	6815.77	1633.29	<u>604.61</u>
13	sko_42_03	42	—	86667.5	86667.5	<b>86655.5</b>	—	6854.25	956.28	<u>621.45</u>
14	sko_42_04	42	—	68769	68733	<b>68722</b>	—	6509.35	1141.34	<u>867.15</u>
15	sko_42_05	42	—	124099.5	124058.5	<b>124017.5</b>	—	7052.59	1236.47	<u>554.07</u>
16	sko_49_01	49	—	20472	20479	<b>20470</b>	—	14934.38	2683.93	<u>542.13</u>
17	sko_49_02	49	—	208270	208081	<b>208078</b>	—	14338.86	2733.87	<u>1009.45</u>
18	sko_49_03	49	—	162267	162196	<b>162187</b>	—	14351.91	2877.19	<u>1173.59</u>
19	sko_49_04	49	—	118311.5	118264.5	<b>118260.5</b>	—	14290.03	3716.16	<u>1193.88</u>
20	sko_49_05	49	—	332990	332855	<b>332811</b>	—	14913.18	2596.39	<u>989.91</u>

由表4的数据对比可知:针对所选的20个测试问题,在问题规模不断扩大时,IFWA的执行时间没有跳跃式增长,在寻优质量和求解效率方面均具有明显优势;目标函数值方面,随着问题规模的增长,IFWA不仅能求得与其他算法相同的解,而且甚至更优;CPU运行时间方面,除N30\_01和N30\_02两个算例中IFWA与CAP\_SS运行时间基本相当外,在其余算例中,IFWA均优于其他4种方法。由表4分析还可知,CAP\_GA、CAP\_SS和IFWA这3种算法的运算时间均随问题规模的增大而不断增加,但是IFWA运算时间的增幅平缓,明显小于其他两种算法。

## 4 结 论

针对现有过道布置问题中忽略设施深度及物流交互点置于过道边线的问题,考虑实际生产制造领域和服务部门中,设施间物流交互点通常在设施几何中心,它会对物流成本产生重大影响,本文构建了以最小化总物流成本为目标的ECAP数学模型,提出了一种改进的烟花算法进行求解,并与LINGO软件精确求解方法及其他算法进行对比,得到以下结论:

1) 建立了ECAP新模型,并验证该模型的正确性。在优化设施间总物流量的同时,考虑设施深度及物流交互点对目标函数的影响。

2) 由于目前尚未发现FWA在设施布局方面的研究,在传统烟花算法的基础上提出了IFWA。本文在采用基于设施编码序列的整数编码方式的同时,应用混沌映射方法产生初始烟花群,随后,采用2-opt邻域构造作为爆炸操作以产生新火花,并设置搜索深度dep,对火花进行两点变异操作,每次寻优结束时对个体解序列和全局最优解进行更新以保留精英解。为改善寻优效率,设置阈值以减少寻优过程的不必要迭代。

3) 应用IFWA对48个不同规模(9~80)的基准问题进行测试,通过对比LINGO优化器所得的精确解,结果表明:对于规模小于13的问题,IFWA能在远小于精确算法所需的时间内求得与其相同的最优解;对于规模大于等于13的问题,本文算法在求解质量和寻优效率方面均明显优于在给定时间内LINGO的精确求解结果。

4) 为验证本文算法的先进性,应用该算法求解原CAP问题并与其他文献中CAP\_SA、CAP\_TS、CAP\_GA、CAP\_SS、CAP-Heuristic等算法的求解结果进行比较,结果表明:IFWA具有较好的收敛性,能快速跳出局部最优,寻求最优目标函数值,在寻优精度、求解效率和平稳性方面均有明显优势。

## 参考文献(References)

- [1] Anjos M F, Vieira M V C. Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions[J]. *European Journal of Operational Research*, 2017, 261(1): 1-16.
- [2] Pei X, He Q. Research on floor improvement based on equipment layout optimization[C]. *International Conference on Machinery, Materials and Computing Technology*. Hangzhou, 2016, 60: 890-893.
- [3] 丁祥海, 姚文鹏. 基于粒子群算法的多目标可重构设施布局方法[J]. *中国机械工程*, 2017, 28(7): 852-861. (Ding X H, Yao W P. Multi-objective reconfigurable facility layout method based on particle swarm optimization[J]. *China Mechanical Engineering*, 2017, 28(7): 852-861.)
- [4] Kang S, Kim M, Chae J. A closed loop based facility layout design using a cuckoo search algorithm[J]. *Expert Systems with Applications*, 2018, 93: 322-335.
- [5] Amaral A R S. The corridor allocation problem[J]. *Computers & Operations Research*, 2012, 39(12): 3325-3330.
- [6] Lin Q, Liu H, Wang D, et al. Integrating systematic layout planning with fuzzy constraint theory to design and optimize the facility layout for operating theatre in hospitals[J]. *Journal of Intelligent Manufacturing*, 2015, 26(1): 87-95.
- [7] Izadina N, Eshghi K. A robust mathematical model and ACO solution for multi-floor discrete layout problem with uncertain locations and demands[J]. *Computers & Industrial Engineering*, 2016, 96: 237-248.
- [8] 刘琼, 赵海飞. 基于多目标果蝇算法面向低碳的车间布局与调度集成优化[J]. *机械工程学报*, 2017, 53(11): 122-133. (Liu Q, Zhao H F. Integrated optimization of workshop layout and scheduling to reduce carbon emissions based on a multi-objective fruit fly optimization algorithm[J]. *Journal of Mechanical Engineering*, 2017, 53(11): 122-133.)
- [9] Hong D, Seo Y, Xiao Y. A concurrent approach for facility layout and AMHS design in semiconductor manufacturing[J]. *International Journal of Industrial Engineering Theory Applications & Practice*, 2014, 21(4): 231-242.
- [10] Ndiaye M A, Dauzère-pères S, Yugma C, et al. Automated transportation of auxiliary resources in a semiconductor manufacturing facility[C]. *Winter Simulation Conference*. Las Vegas: IEEE Press, 2016: 2587-2597.

- [11] Ghosh D, Kothari R. Population heuristics for the corridor allocation problem[J]. *Iima Working Papers*, 2012, 98(2): 33-40.
- [12] Ahonen H, Alvarenga A G D, Amaral A R S. Simulated annealing and tabu search approaches for the corridor allocation problem[J]. *European Journal of Operational Research*, 2014, 232(1): 221-233.
- [13] Kalita Z, Datta D. Solving the bi-objective corridor allocation problem using a permutation-based genetic algorithm[J]. *Computer & Operations Research*, 2014, 52(5): 123-134.
- [14] 毛丽丽, 张则强, 汪开普, 等. 求解过道布置问题的一种改进分散搜索算法[J]. *计算机集成制造系统*, 2017, 23(8): 1641-1651.  
(Mao L L, Zhang Z Q, Wang K P, et al. Improved scatter search algorithm for corridor allocation problem[J]. *Computer Integrated Manufacturing System*, 2017, 23(8): 1641-1651.)
- [15] Kalita Z, Datta D, Palubeckis G. Bi-objective corridor allocation problem using a permutation-based genetic algorithm hybridized with a local search technique[J]. *Soft Computing*, 2017, 13(3): 961-986.
- [16] 管超, 张则强, 毛丽丽, 等. 双层过道布置问题的混合整数规划模型及启发式求解方法[J]. *计算机集成制造系统*, 2018, 24(8): 1972-1982.  
(Guan C, Zhang Z Q, Mao L L, et al. A mixed integer programming model and heuristic method for the double-layer corridor allocation problem[J]. *Computer Integrated Manufacturing Systems*, 2018, 24(8): 1972-1982.)
- [17] Tan Y, Zhu Y. Fireworks algorithm for optimization[C]. *Proceedings of the 1st International Conference on Advances in Swarm Intelligence*. Beijing: Springer-Verlag, 2010: 355-364.
- [18] 朱启兵, 王震宇, 黄敏. 带有引力搜索算子的烟花算法[J]. *控制与决策*, 2016, 31(10): 1853-1859.  
(Zhu Q B, Wang Z Y, Huang M. Fireworks algorithm with gravitational search operator[J]. *Control and Decision*. 2016, 31 (10): 1853-1859.)
- [19] Lu C, Li J Y. Assembly sequence planning considering the effect of assembly resources with a discrete fireworks algorithm[J]. *International Journal of Advanced Manufacturing Technology*, 2017, 93(6): 1-18.
- [20] 薛俊杰, 王瑛, 孟祥飞, 等. 二进制反向学习烟花算法求解多维背包问题[J]. *系统工程与电子技术*, 2017, 39(2): 451-458.  
(Xue J J, Wang Y, Meng X F, et al. Binary opposite backward learning fireworks algorithm for multidimensional knapsack problem[J]. *Systems Engineering and Electronics*, 2017, 39(2): 451-458.)
- [21] Tubaileh A, Siam J. Single and multi-row layout design for flexible manufacturing systems[J]. *International Journal of Computer Integrated Manufacturing*, 2017, 30(1): 1-15.
- [22] Amaral A R S. A parallel ordering problem in facilities layout[J]. *Computers & Operations Research*, 2013, 40(12): 2930-2939.
- [23] Zuo X Q, Murray C C, Smith A E. Sharing clearances to improve machine layout[J]. *International Journal of Production Research*, 2016, 54(14): 1-14.
- [24] Nagarajan L. Solution for bi-objective single row facility layout problem using artificial bee colony algorithm[J]. *European Journal of Industrial Engineering*, 2018, 12(2): 252-275.
- [25] Palubeckis G. Single row facility layout using multi-start simulated annealing[J]. *Computers & Industrial Engineering*, 2017, 103: 1-16.
- [26] Yu M, Zuo X, Murray C C. A tabu search heuristic for the single row layout problem with shared clearances[C]. *Evolutionary Computation*. Beijing: IEEE, 2014: 819-825.
- [27] 单梁, 强浩, 李军, 等. 基于Tent映射的混沌优化算法[J]. *控制与决策*, 2005, 20(2): 179-182.  
(Shan L, Qiang H, Li J, et al. Chaotic optimization algorithm based on Tent map[J]. *Control and Decision*, 2005, 20(2): 179-182.)

### 作者简介

刘思璐(1993—), 女, 博士生, 从事设施布局优化的研究, E-mail: liusilu\_66@163.com;

张则强(1978—), 男, 教授, 博士生导师, 从事制造系统与智能优化等研究, E-mail: zzq\_22@163.com;

管超(1994—), 男, 博士生, 从事设施布局优化的研究, E-mail: 17175371524@163.com;

龚举华(1990—), 男, 博士生, 从事设施布局优化的研究, E-mail: gongjuhua@my.swjtu.edu.cn.

(责任编辑: 李君玲)