

# 基于改进飞蛾扑火算法的集成调度仿真

张天瑞, 牛慧媛\*, 谢薇

(沈阳大学 机械工程学院, 辽宁 沈阳 110044)

**摘要:** 针对制造型企业成本上升等问题, 建立生产与运输集成调度的数学规划模型, 提出一种双自适应权重飞蛾扑火算法(double adaptive weights for moth flame optimizer, DAWMFO)进行求解。提出了一种双自适应权重机制; 使用螺旋函数更新种群, 提高了算法的收敛速度和精度。采用改进算法对基准函数进行了测试, 结果表明: DAWMFO 不仅能够快速收敛且不易陷入局部最优, 寻优能力较其他算法也有所提升。通过对生产与运输集成调度的数据集做仿真实验, 验证了改进算法在求解此类问题时可行高效。

**关键词:** 作业车间调度; 螺旋函数; 双自适应权重飞蛾扑火算法; 生产和运输; 集成调度

中图分类号: TP391.9

文献标志码: A

文章编号: 1004-731X(2023)10-2170-12

DOI: 10.16182/j.issn1004731x.joss.23-FZ0806

**引用格式:** 张天瑞, 牛慧媛, 谢薇. 基于改进飞蛾扑火算法的集成调度仿真[J]. 系统仿真学报, 2023, 35(10): 2170-2181.

**Reference format:** Zhang Tianrui, Niu Huiyuan, Xie Wei. Integrated Scheduling Simulation Based on Improved Moth Flame Optimizer[J]. Journal of System Simulation, 2023, 35(10): 2170-2181.

## Integrated Scheduling Simulation Based on Improved Moth Flame Optimizer

Zhang Tianrui, Niu Huiyuan\*, Xie Wei

(School of Mechanical Engineering, Shenyang University, Shenyang 110044, China)

**Abstract:** Aiming at the rising cost of manufacturing enterprises, a mathematical programming model of integrated scheduling of production and transportation is established, and a double adaptive weights for moth flame optimizer(DAWMFO) is proposed. A double adaptive weight mechanism is proposed. The spiral function is used to update the population, which improves the convergence speed and accuracy of the algorithm. The benchmark function is tested by the improved algorithm. The results show that the improved algorithm can converge quickly and not easily fall into local optimum. Compared with other algorithms, the optimization ability is also improved. Through the simulation experiment on the data set of integrated production and transportation scheduling, it is verified that the improved algorithm is feasible and efficient in solving such problems.

**Keywords:** job shop scheduling; spiral function; double adaptive weights for moth flame optimizer; production and transportation; integrated scheduling

## 0 引言

随着《中国制造 2025》的公布, 我国制造业由“制”到“智”进行转型。生产调度是调度的

一个重要方面, 已成为制造业快速响应定制化需求的有效途径。企业也开始向接单生产模式(make-to-order mode, MTO)转变。其本质是将生产与分销协同进行, 实现协同最优化, 使其更科学

收稿日期: 2023-07-02

修回日期: 2023-08-24

基金项目: 国家自然科学基金面上项目(52075088); 辽宁省研究生教育教学改革研究资助项目(LNYJG2022490)

第一作者: 张天瑞(1985-), 男, 副教授, 博士, 研究方向为车间调度和智能算法。E-mail: trzhang@syu.edu.cn

通讯作者: 牛慧媛(1999-), 女, 硕士生, 研究方向为车间调度和智能算法。E-mail: lareinaup@163.com

高效地提升订单生产的效益, 减少企业综合成本。因此, 研究生产与运输集成调度更具有实际意义。

越来越多的学者关注作业车间调度问题, 研究内容主要为动态作业车间、柔性车间调度。在生产调度方面, Caldeira 等<sup>[1]</sup>提出一种多目标的生产调度优化方法, 使其解决降低能耗、最小化最大完工时间的多目标问题。Firat 等<sup>[2]</sup>主要关注工厂型制造业, 研究了一种混合整数线性计划, 该模型可以通过订单接受决策来寻找工作负荷相关的规划水平。在运输调度方面, 陈魁等<sup>[3]</sup>针对考虑运输时间的柔性作业车间调度问题, 提出一种小生境粒子群优化算法求解该模型。Hidri 等<sup>[4]</sup>针对具有运输时间的混合流水车间调度问题, 提出两阶段的启发式方法解决此类问题。以上研究可以看出, 大多数学者是分开研究生产调度和运输调度的优化, 没有把生产和运输阶段联系起来。

随着研究的深入, 一些学者开始研究考虑生产与运输结合的集成调度。Geismar 等<sup>[5]</sup>将短期库存的产品特性纳入到生产与运输协调调度中, 使得制造和交货的周期最短。Mohammadi 等<sup>[6]</sup>针对解决具有时间窗的集成生产调度和车辆路线问题, 建立了双目标混合整数模型, 以达到产品与运输计划的费用和交货时间最小的目的。Gharaei 等<sup>[7]</sup>研究双代理在集成生产调度和分配问题中的应用, 并针对问题提出混合整数线性规划公式求解。Moazami 等<sup>[8]</sup>针对生产和分销计划的集成问题, 提出混合整数线性规划公式, 以最大限度减少延误和运输成本为目标来解决集成问题。梁春华等<sup>[9]</sup>以多个企业为例, 对多个企业和客户进行生产与分配优化问题进行研究, 将独立最优和协同最优的方法进行对比, 得到更优的结果。刘玲等<sup>[10]</sup>构建了以最短完成时间为优化目标的制造—车辆运输协调规划的数学模型。霍思云<sup>[11]</sup>对不同权重的产品在不同容量平行机上的排序问题进行分析, 将生产与运输的集成规划划分为两个子问题进行求解。吴倩云等<sup>[12]</sup>以最优的运输效率为目标, 建立

了一个考虑时间窗和最佳载荷限制的汽车路线分配模式。

飞蛾扑火算法(moth flame optimization, MFO)是 Mirjalili<sup>[13]</sup>提出的一种新型智能优化算法, 算法可行性强, 目前在解决函数优化<sup>[14]</sup>、图像分割<sup>[15]</sup>、预测模型<sup>[16]</sup>等问题上取得了较好的效果。文献[17]采用 MFO 针对预测问题求得高质量解。

综上所述, 目前国内外针对作业车间和流水车间为基础的调度问题研究已较为成熟, 但对于复杂车间环境的研究还需进一步深入。另一方面, 在生产和配送环节会根据目标建立相应模型, 在分销过程中, 由于目标的差异, 会形成相应的模式。在制造环节, 资源包括原材料、加工设备。而在运输环节, 资源包括运输车辆、成品产出等。各单位在制订规划时, 往往会顾及本部门的利益, 而忽视其他部门或总体的利益。因此, 本文着眼于生产与运输集成调度问题, 同时考虑最大完工时间和成本两个优化目标, 设计了一种双自适应权重飞蛾扑火算法(double adaptive weights for MFO, DAWMFO)。

## 1 问题描述与建模

### 1.1 问题描述

生产与运输集成调度模型描述: 含有  $g$  台机器和  $n$  个客户, 制造商接收  $n$  个顾客的订单  $D_n$ , 顾客分散在各地, 厂家接到订单后, 向供应商采购原材料, 供应商提供多种原材料, 一种原材料可以生产出多种产品。如果交货时间超过时间窗限度, 则产生罚款费用。

本文以运输成本和违反时间窗成本组成的总分配成本最小化为目标, 通过合理安排机器上的物品处理顺序, 交货路线和车辆的出发时间, 以实现成本和最大完工时间的性能达到整体最优的目的。本文针对所建模型, 做出以下假设:

- (1) 原材料充足且满足生产需求;
- (2) 厂商租赁车辆的种类和数量已确定;

(3) 忽略各工序在机器上加工准备时间和加工装卸时间;

(4) 假定运送材料的时间记为 0;

(5) 不考虑交通状况和运输中的设施等;

(6) 按批量生产方式生产同类产品。

## 1.2 数学模型

为建立总成本最小的生产与运输集成调度模型, 给出以下符号说明。

$p_i$ : 客户  $i$  的加工工序;

$s_i$ : 客户  $i$  项目的大小;

$r_{ij}$ : 在客户  $i$  和客户  $j$  的目的地之间的运输时间;

$q_k$ : 车辆  $k$  的最大承载量;

$[l_i, \mu_i]$ : 项目交货时间窗;

$C_k$ : 单位时间车辆  $k$  的运输成本;

$c_p$ : 工件  $p$  的生产成本;

$c_{pt}$ : 单位时间工件  $p$  的加工成本;

$t_{pi}$ : 加工工件  $p_i$  的时间;

$c_e$ : 单位时间内提前罚款成本;

$c_i$ : 单位时间内延误惩罚成本;

$t_{kai}^E$ : 第  $a$  次运输成品  $i$  的  $k$  号车到达最后一位客户的时刻;

$C_i$ : 客户  $i$  成品生成的产品完工时间;

$F_k$ : 车辆  $k$  交付产品的最大完成时间;

$D_i$ : 客户  $i$  收到产品时的交货时间;

$E_i$ : 客户  $i$  提前交货时间;

$T_i$ : 客户  $i$  延迟交货时间;

$o_{im}$ : 如果订单  $i$  被分配给机器  $m$  则为 1, 否则为 0;

$y_{ik}$ : 如果车辆  $k$  交付订单  $i$  则为 1, 否则为 0;

$x_{ijm}$ : 如果订单  $i$  在订单  $j$  之前被机器  $m$  加工为 1, 否则为 0;

$z_{ijk}$ : 如果订单  $i$  先于订单  $j$  被车辆  $k$  配送交货为 1, 否则为 0。

本文以生产成本、运输成本和违反时间窗的惩罚成本组成的, 以总分配成本最小化和最小化

最大完工时间为优化目标建立模型, 其目标函数为

$$\min C = c_p + c_k \sum_{i \in N} \sum_{j \in N} \sum_{k \in V} r_{ij} z_{ijk} + c_e \sum_{i \in N \setminus \{0\}} E_i + c_t \sum_{i \in N \setminus \{0\}} T_i \quad (1)$$

$$\min(\max(f_k + C_i)) \quad (2)$$

约束条件

(1) 订单分配阶段

每个订单都被分配给其中一台机器:

$$\sum_{m \in K} o_{im} = 1, \forall i \in N \setminus \{0\} \quad (3)$$

每个订单要么直接先于另一个订单, 要么直接接受另一个订单:

$$o_{im} = \sum_{j \in N} x_{ijm}, \forall i \in N \setminus \{0\}, i \neq j, m \in K \quad (4)$$

$$\sum_{j \in N} x_{ijm} = \sum_{j \in N} x_{ijm}, \forall i \in N \setminus \{0\}, i \neq j, m \in K \quad (5)$$

最多可将一个订单作为每台机器的第一个作业进行处理:

$$\sum_{j \in N \setminus \{0\}} x_{0jm} \leq 1, \forall m \in K \quad (6)$$

项目  $i$  的完成时间不小于其前身的完成时间加上其处理时间  $p_i$ :

$$c_i \geq c_j + p_i - M(1 - x_{ijm}), \forall i \in N \setminus \{0\}, j \in N, i \neq j, \forall m \in K \quad (7)$$

(2) 物品交付阶段

每个物品只能由一辆车交付:

$$\sum_{k \in V} y_{ik} = 1, \forall i \in N \setminus \{0\} \quad (8)$$

在每个客户位置只有一辆车到达和离开:

$$y_{0k} \geq y_{ik}, \forall i \in N \setminus \{0\}, k \in V \quad (9)$$

$$\sum_{j \in N} z_{ijk} = \sum_{j \in N} z_{ijk}, \forall i \in N \setminus \{0\}, i \neq j, k \in V \quad (10)$$

确保在每辆车的运输负荷不超过车辆的装载能力:

$$\sum_{i \in N \setminus \{0\}} s_i y_{ik} \leq q_k, \forall k \in V \quad (11)$$

(3) 物品运输阶段

保证每辆车交付的物品最长时间不小于该车辆运输的物品的最大完成时间:

$$f_k \geq c_i - M(1 - y_{ik}), \forall k \in V, i \in N \setminus \{0\} \quad (12)$$

确保每辆车的出发时间不小于其车辆运输物品的最长时间:

$$d_k \geq f_k, \forall k \in V \quad (13)$$

车辆  $k$  每个项目交付时间不小于车辆出发时间  $d_k$  与上订单  $i$  目的地和工厂之间的行驶时间  $r_{oi}$  的和:

$$D_i \geq d_k + r_{oi} - M(1 - z_{oik}), \forall i \in N \setminus \{0\}, \forall k \in V \quad (14)$$

计算了提前时间和延迟时间:

$$E_i \geq l_i - D_i, \forall i \in N \quad (15)$$

$$C_i, E_i, T_i \geq 0, \forall i \in N \quad (16)$$

决策变量的结构域:

$$y_{ik}, o_{im}, x_{ijm}, z_{ijk} \in \{0, 1\}, \forall k \in V, \quad (17)$$

$$m \in K, \forall i, j \in N, i \neq j \quad (18)$$

$$d_k, f_k \geq 0, \forall k \in V \quad (18)$$

## 2 双自适应权重飞蛾扑火算法

MFO 是一种元启发式算法, 在搜索空间中寻找最优的个体, 同时保留最优的位置, 分配到火焰上, 使飞蛾朝最佳的火焰靠近, 从而达到全局最优。搜索空间中火焰为

$$F = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,d} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,d} \\ \vdots & \vdots & & \vdots \\ f_{n,1} & f_{n,2} & \cdots & f_{n,d} \end{bmatrix} \quad (19)$$

飞蛾运动轨迹的螺旋函数:

$$S(M_i, F_j) = |F_j - M_i| \cdot e^{bt} \cdot \cos(2\pi t) + F_j \quad (20)$$

式中:  $M_i, i=1, 2, \dots, n$  为第  $i$  个飞蛾的位置, 其中,  $M_i$  代表机器加工的顺序,  $n$  代表种群规模;  $F_j$  为第  $j$  个火焰的位置, 它的值即适应度函数值, 也就是本文所求的工序在机器上的完工时间;  $|F_j - M_i|$  为从飞蛾到火焰的距离, 代表机器加工的最优方式;  $b$  为对数螺旋常数;  $t \in [-1, 1]$  为距离参数。

MFO 在不同时期没有特定的搜索策略, 使飞蛾在寻找最优火焰位置时易陷入局部最优, 因此, 需要提高全局搜索能力。本文提出一种双自适应权重机制, 可以避免不必要的局部搜索和全局搜

索。当使用螺旋函数更新飞蛾种群时, 根据不同的搜索周期, 给予更临界位置更高的权重, 从而提高改进算法的收敛速度和精度。

### 2.1 双自适应权重机制

原 MFO 算法中重要的一部分是飞蛾的螺旋函数  $S$ , 由飞蛾在飞行中建造, 它反映了飞蛾靠近火的过程。改进算法针对不同阶段的收敛性, 在原算法中增加了  $w_1$  和  $w_2$  权重, 使其在迭代过程中, 按照该方法找到火焰的最优位置, 即加工产品最大完工时间的最小值。

$$w_1 = (1 - K/T)^{1 - \tan(\pi \cdot (rand - 0.5)) \cdot \frac{S}{T}} \quad (21)$$

$$w_2 = (2 - 2K/T)^{1 - \tan(\pi \cdot (rand - 0.5)) \cdot \frac{S}{T}} \quad (22)$$

式中:  $K$  为当前迭代次数;  $T$  为最大迭代次数;  $rand$  生成  $[0, 1]$  中均匀随机数。在复杂问题中, 算法必须显示更多的渗透行为, 随机元素通常会带来更多的随机搜索模式, 以便找到工序完工时间的最优解。

### 2.2 双自适应权重飞蛾扑火算法的模型

当 MFO 算法迭代更新时, 判断当前飞蛾位置  $i$  与火焰数  $n_{\text{flame}}$  之间的关系。当小于  $n_{\text{flame}}$  时,  $F_j$  式(20)将对第一个飞蛾  $i$  的适应度值排序进行迭代; 当  $i$  数值大于  $n_{\text{flame}}$  时, 式(20)将在适应度值排序中选取第  $B$  个飞蛾进行迭代。用于迭代的螺旋函数是不变的。双自适应权重将在这 2 种情况下给出两部分不同的权重, 使迭代搜索更加准确。

当  $i < n_{\text{flame}}$  时, 螺旋函数中双权重  $w_1$  和  $w_2$  的位置为

$$S(M_i, F_j) = |F_j - M_i| e^{bt} \cos(2\pi t) + w_2 F_j \quad (23)$$

$$S(M_i, F_j) = w_1 |F_j - M_i| e^{bt} \cos(2\pi t) + F_j \quad (24)$$

当  $i \geq n_{\text{flame}}$  时, 螺旋函数中双权重  $w_1$  和  $w_2$  的位置为

$$S(M_i, F_j) = w_2 |F_j - M_i| e^{bt} \cos(2\pi t) + F_j \quad (25)$$

$$S(M_i, F_j) = |F_j - M_i| e^{bt} \cdot \cos(2\pi t) + w_1 F_j \quad (26)$$



## 2.3 双自适应权重飞蛾扑火算法流程

本文对原算法做了相应的策略改进, 引入了双自适应权重  $w_1$ 、 $w_2$ , 分情况地讨论了如何更新飞蛾  $S$  的螺旋函数位置更新, 根据不同的情况向目标函数中加入 2 个权重的位置, 以此解决前期容易造成局部最优和后期不必要的全局最优问题, 增加寻找最优解可能性, 其具体改进算法流程如图 1 所示, 步骤如下。

初始化种群  $M_i, i=1, 2, \dots, n$ , 用飞蛾位置  $M_i$  表示机器加工顺序, 最大迭代次数  $T$  参数, 种群规模  $n$  (机器数量), 维数  $d$  (产品任务的机器分配);

While  $l \leq T$

根据  $n_{\text{flame}}$  公式更新火焰位置;

计算每个飞蛾的适应度值  $OM$ ;

利用式(19)计算火焰的数量(产品完工时间集合);

若当前迭代次数  $k=1$ , 则根据  $OF = \text{sort}(OM)$ ,  $F = \text{sort}(M)$  更新火焰种群;

否则根据  $OF = \text{sort}(M_{k-1}, OM_k)$ ,  $F = \text{sort}(M_{k-1}, M_k)$  更新火焰种群;

记录第一个火焰最优位置(产品最大完工时间最小值);

for  $i = 1 : n$

更新  $b$ 、 $t$ ;

若飞蛾  $i < n_{\text{flame}}$ , 则根据式(23)和(24)更新螺旋函数;

否则根据式(25), (26)更新飞蛾螺旋函数;

判断飞蛾位置是否超出了搜索空间的上下限(调度问题的可行解范围);

若飞蛾跳出边界线, 则重新返回初始化位置;

end for

end While

输出最优解, 即得到产品最大完工时间的最小值。

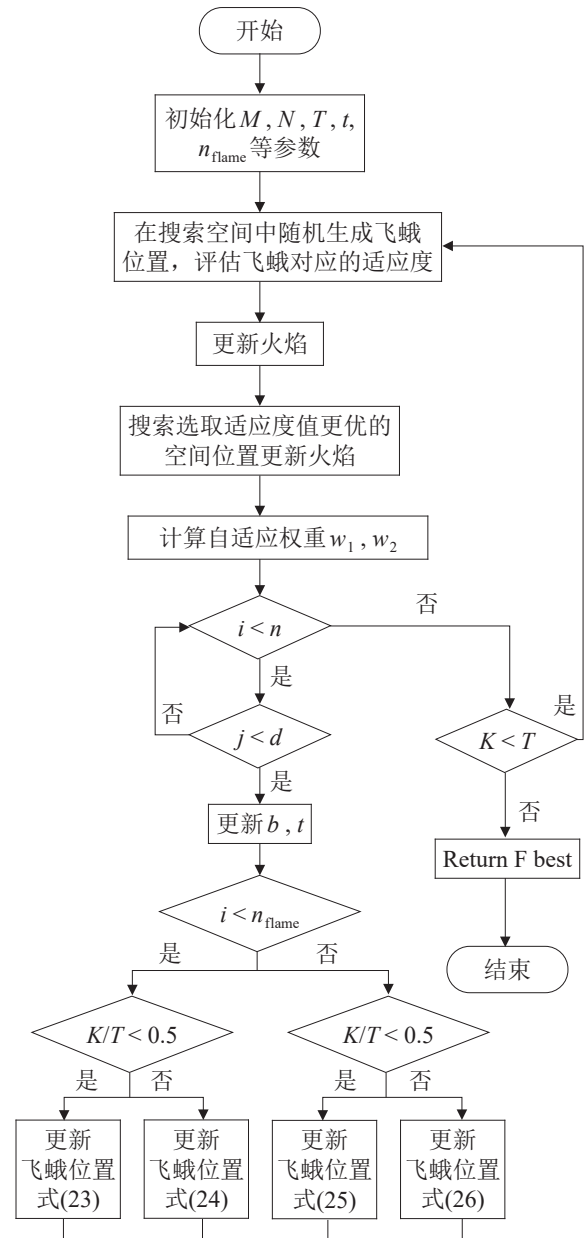


图 1 双自适应权重飞蛾扑火算法流程图

Fig. 1 Flowchart of moth flame optimization with double adaptive weights

## 3 仿真实验

本文采用经典测试函数和生产与运输集成调度实例进行仿真实验测试算法的有效性。所有算法均在 MATLAB2018a 中编码, 并在 Intel Core i7、1.80 GHz、8 GB、Windows10 操作系统的计算机上运行。

### 3.1 经典测试函数仿真

#### 3.1.1 基准函数测试

为对 DAWMFO 的性能进行检验和评估, 选取 21 个不同种类的测试函数。针对不同函数, 其取值范围均不同, 通过这些测试函数全面地检测算法的局部寻优和全局寻优能力。本文测试函数的维度测试均在 10 次以上。通过设置不同的位置进行测试, 进一步增加算法的寻优性能。

图 2 为单峰函数测试, 可以看出其具有单一峰值准则的寻优性能, 且不存在任何局部最优。通过图像可以看出, 图像几乎呈漏斗状, 且存在极小值点。

图 3 为多峰函数测试, 其局部最优值较多, 寻找问题的难度较大, 但有助于对该方法的整体优化和局部优化的综合性能进行分析, 与单一峰

值相比, 多个峰值的收敛性更好。

图 4 为固定维多峰函数的测试, 此类函数对算法的局部搜索能力有更明显的体现, 在算法运行约一半时, 进入局部搜索的状态。

通过部分函数的测试结果可以看出, 混合型函数的收敛性更佳, 优化出来的目标值更接近理论最优。从该模型的收敛曲线可以看出, 曲线初期起伏较大, 随后渐渐趋于平稳。结果表明: 该方法可以在较短时间内进行较大范围的搜寻, 且在接近的范围内, 也可以得到较好的结果。因此, 可以看出 DAWMFO 比原算法可以更快地找到问题的最优解, 说明对算法的改进是成功和有效的。

#### 3.1.2 不同智能优化算法对比实验

为了更全面地体现算法的优化性能, 先将 DAWMFO 与 MFO 和其他算法进行对比分析。

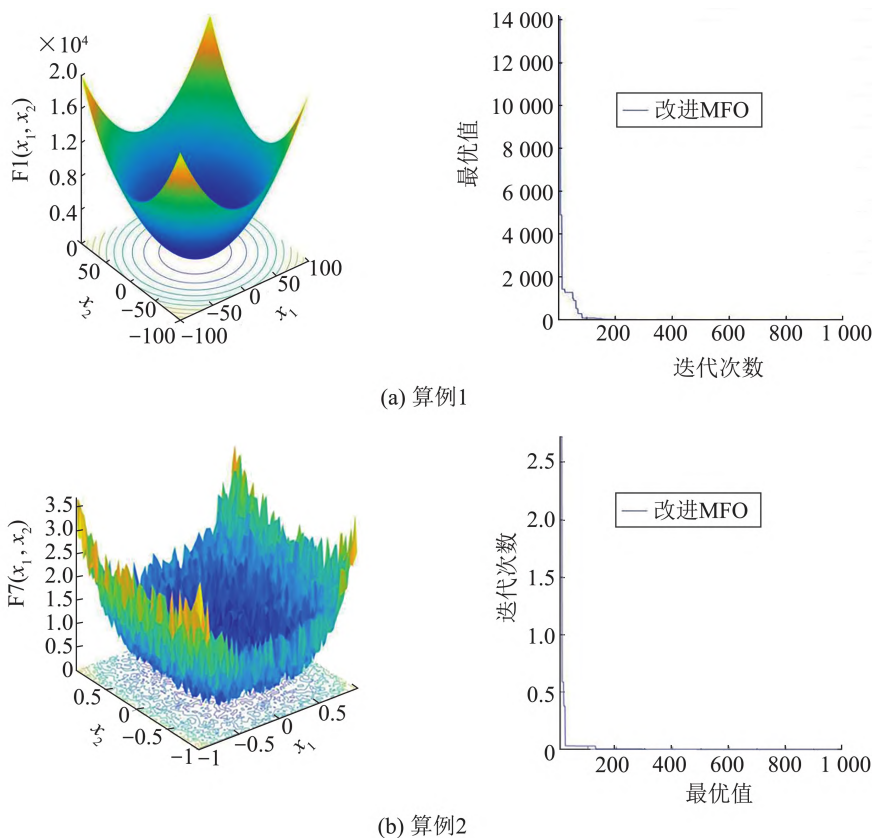


图 2 单峰函数测试图像和收敛曲线

Fig. 2 Unimodal function test image and convergence curve

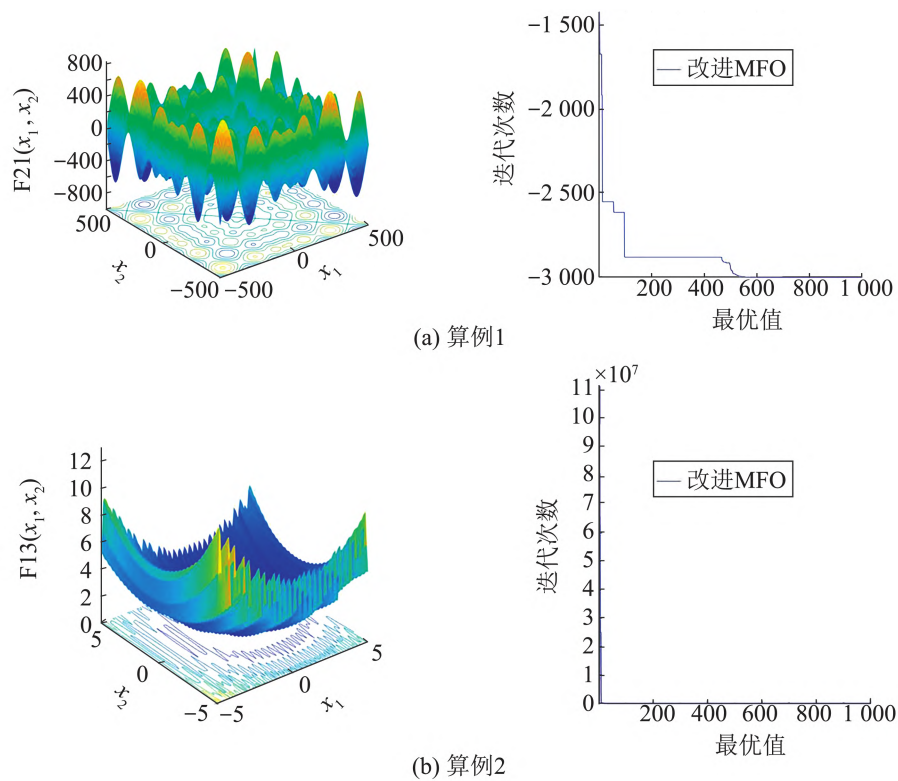


图 3 多峰测试函数图像和收敛曲线  
Fig. 3 Image and convergence curve of multi-peak test function

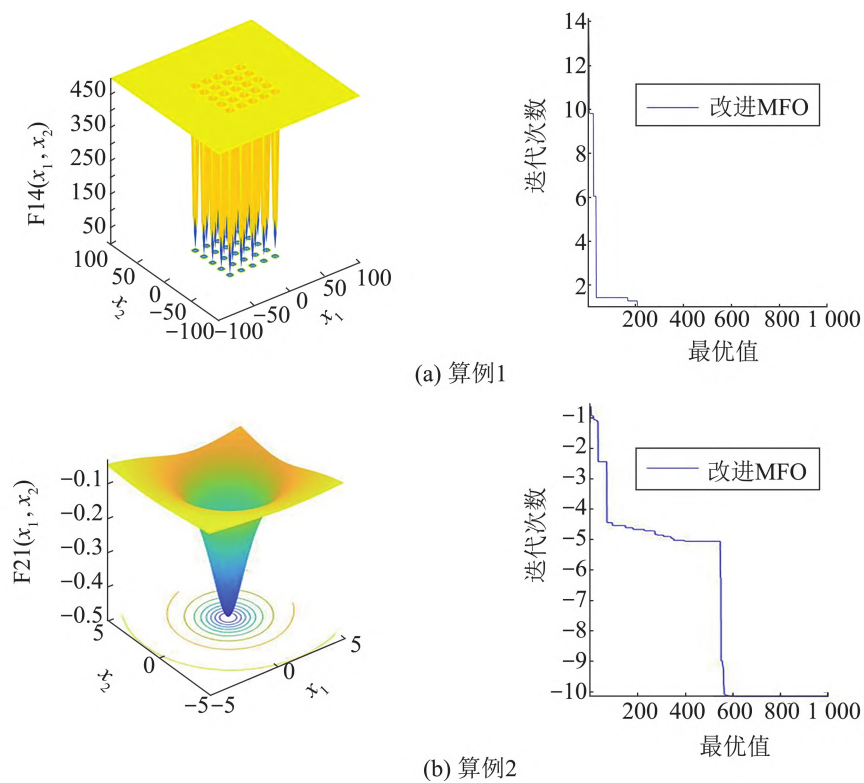


图 4 固定维多峰测试函数的搜索空间以及收敛曲线  
Fig. 4 Search space and convergence curve of multi-peak test function with fixed dimension

正弦余弦算法(sine cosine algorithm, SCA)<sup>[18]</sup>和萤火虫算法(firefly algorithm, FA)<sup>[19]</sup>在一次迭代过程中可以产生多个 Pareto 非支配解, 在求解集成调度问题方面更具优势, 因此, 将 DAWMFO 与 SCA、FA 和 MFO 进行仿真对比研究。

算法参数设置: SCA 控制参数  $a=2$ ; FA 步长因子  $\varepsilon=2$ , 吸引系数  $\delta=1$ ; MFO 和 DAWMFO 的对数螺旋常数  $b=1$ 。为避免偶然性, 将以上各算法在相同的基准测试函数上运行, 迭代次数  $T=1\ 000$ , 种群规模  $N=30$ , 运行结果如表 1 所示。

由表 1 可知, 由于均值表示测试结果, 均值越低, 算法的质量越高, 标准差越小, 算法的稳定性越好。因此, 可以看出在相同维度下测试, DAWMFO 算法性能均优于 MFO 算法。

此外, DAWMFO 与 SCA、FA 相比, 具有明显的优势, 其在混合基准函数方面的性能也更加优良。表明 DAWMFO 算法在跳出局部最优的问题上具有更好的性能。

## 3.2 实例仿真实验

由于目前没有本文所研究的集成调度实例, 因此参考文献[20]中生产与运输集成调度问题研究的算例。设置算法初始化参数: 100 个初始种群数, 经过 1 000 次反复计算后, 得到一个最优的生

产调度方案, 并在该优化的方案下, 列出这 6 种产品的各道工序在不同机器上的完工时间数据信息, 如表 2 所示。

在表 2 中, FT 表示完工时刻, 从表中可以看到各个生产过程在机器上完成的时限。

图 5 显示了相应的机器上的各种制品的处理时间, 最小化最大完工时刻是 409.14 min, 即第 2 个产品的第 6 个工序在 M3 机器上的处理。然后, 对企业的协作优化问题进行了研究。

### 3.2.1 产品完工时间分析

产品完工时间分析主要针对上述优化得出的结果, 对作业车间所生产的产品完工时间进行分析。根据每个订单的实际生产完成时间与客户需求的时间对比, 计算其差值, 来验证该方法的有效性。

根据以上的优化结果对产品生产的具体时间数据统计, 可以得出交付时间取决于生产加工的顺序、加工的完工时间。由此, 可以计算出 6 种订单产品在工厂的所有加工完成时间, 如图 6 所示。

由图 6 可知, 最小化最大完工时间为 409.14 min, 其中, 案例中 6 种产品都需要加工 6 道工序, 每个工序的具体完工时间如表 3 所示。

表 1 测试函数均值和标准差  
Table 1 Mean and standard deviation of test functions

函数	DAWMFO		MFO		SCA		FA	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F1	<b>1.2E+04</b>	<u>1.1E+04</u>	4.2E+10	2.4E+10	1.2E+10	2.0E+09	1.4E+10	1.5E+09
F2	1.8E+90	1.0E+91	1.8E+172	6.6E+04	1.9E+35	6.7E+34	<b>1.0E+34</b>	<u>1.6E+40</u>
F3	<b>1.5E+05</b>	<u>1.6E+04</u>	5.4E+05	1.9E+05	3.7E+04	6.3E+03	6.0E+04	1.1E+04
F4	6.7E+02	4.4E+01	2.5E+04	<u>1.2E+04</u>	<b>1.4E+03</b>	2.3E+02	1.4E+03	9.7E+01
F5	<b>1.6E+03</b>	2.4E+02	1.8E+03	1.4E+02	7.8E+02	1.9E+01	7.6E+02	<u>1.2E+01</u>

表 2 优化后工序的完工时间  
Table 2 Completion time of optimized process

机器							机器						
	FT1	FT2	FT3	FT4	FT5	FT6		FT1	FT2	FT3	FT4	FT5	FT6
M1	140.97	200.47	247.31	276.37	316.67	367.45	M4	179.23	215.23	297.36	314.03	343.17	395.14
M2	74.00	125.00	164.00	245.27	317.56	362.63	M5	29.00	102.45	152.32	243.68	321.37	398.46
M3	43.00	79.00	140.38	226.23	268.23	409.14	M6	60.00	87.77	155.77	221.76	195.76	280.43



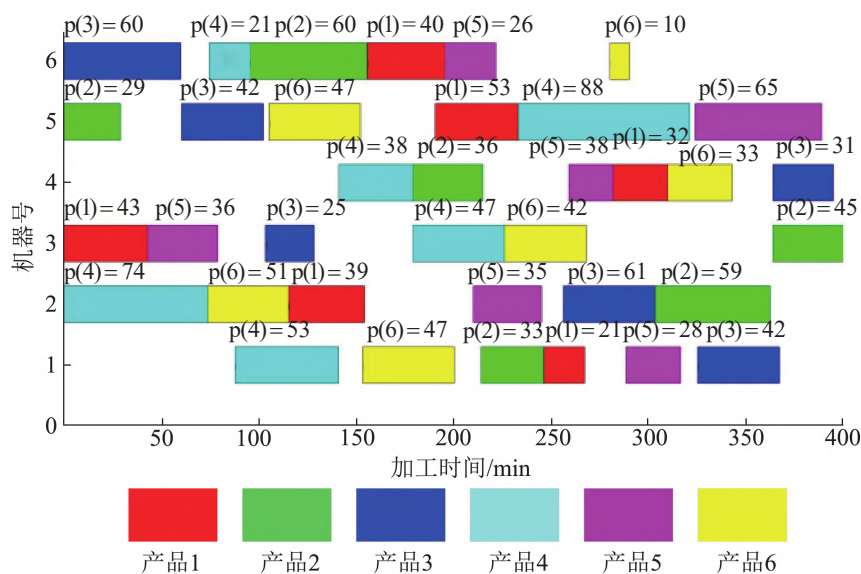


图 5 改进后生产调度甘特图  
Fig. 5 Gantt chart of improved production scheduling

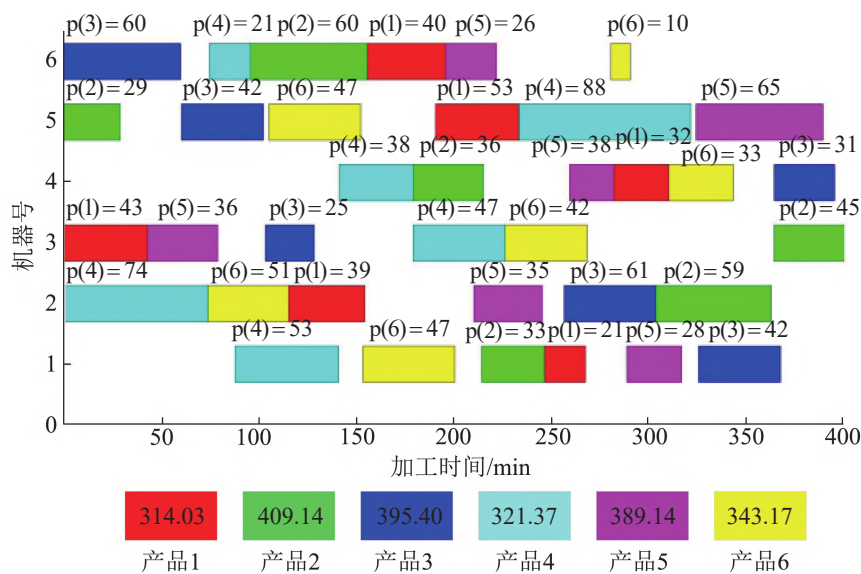


图 6 生产加工完成时间  
Fig. 6 Completion time of production and processing

表3 每个工序的具体完工时间												
Table 3 Specific completion time of each process												min
机器	工序	完工时间	工序	完工时间	工序	完工时间	工序	完工时间	工序	完工时间	工序	完工时间
M1	4-3	140.97	6-3	200.47	2-4	247.31	1-5	276.37	5-5	316.67	3-5	367.45
M2	4-1	74.00	6-1	125.00	1-2	164.00	5-3	245.27	3-4	317.56	2-6	362.63
M3	1-1	43.00	5-1	79.00	3-3	140.38	4-5	226.23	6-4	268.23	2-5	409.14
M4	4-4	179.23	2-3	215.23	5-4	297.36	1-6	314.03	6-6	343.17	3-6	395.14
M5	2-1	29.00	3-2	102.45	6-2	152.32	1-4	243.68	4-6	321.37	5-6	398.46
M6	3-1	60.00	4-2	87.77	2-2	155.77	1-3	221.76	5-2	195.76	6-5	280.43

在表 3 中, 工序 4-3 表示第 4 个产品的第 3 道工序, M1 代表机器 1, 完成时间表示 P 产品的完成时间。表 3 罗列出 6 种产品的完成周期, 以及各加工过程所需的相应机器, 利用本文方法和改进的计算方法, 得到了最终的产品完成时限与顾客产品完成时间的比较结果, 对比数据如表 4 所示。

在表 4 中, Dm 和 Dn 分别表示客户需求的最早交货时间和最晚交货时间, 与顾客需求的订单完工时间与距离客户和工厂的间距, 以及从工厂到客户

指定的位置的车辆行驶速度有关, 因此, 本文将汽车的运行速度设为 60 km/h。根据客户所能接受的时间范围, 规定了送达订单的最大提前时间和最大延后时间。由表 4 的数据可知, 本文应用的方案求解结果基本满足客户的需求, 第 2、4、5 和 6 个产品均能够满足客户的需要。第 1 个产品加工完, 实际运输时间比客户需求的时间提前了 0.91 min, 第 3 个产品加工完加上车辆运输时间的实际配送时间比客户需求的时间延后 34.66 min。

表 4 产品实际完工时间与配送时间需求时间对比  
Table 4 Comparison of actual product completion time and delivery time demand time

机器	工序	实际完工时间/min	距离/km	客户需求产品时间				偏差/min
				提前惩罚时间/min	延迟惩罚时间/min	Dm/min	Dn/min	
M1	1-6	314.03	30.41	345.35	375.35	350.35	370.35	0.91
M2	2-6	409.14	21.24	395.40	435.40	400.40	430.40	0
M3	3-6	395.14	44.72	365.20	405.20	370.20	400.20	-34.66
M4	4-6	321.37	8.21	305.00	335.00	310.00	330.00	0
M5	5-6	398.46	33.97	385.50	445.50	390.50	435.50	0
M6	6-6	343.17	36.72	325.57	385.57	330.57	380.57	0

3.2.2 结果对比分析

在顾客对运输时间有要求的情况下, 可以使产品的完成时限更好地符合顾客的要求; 如果没有满足顾客对交货时间的要求, 则产品的完成时间都有延迟的可能。本文将生产与运输集成调度的方案考虑客户需求运输时间与没有考虑客户的需求运输时间的情况进行对比, 如表 5 所示。

表 5 优化结果对比  
Table 5 Comparison of optimization results

成本项目	未考虑运输 时间成本/元	考虑运输 时间成本/元	优化率/%
运输	255.21	246.31	3.49
惩罚	343.13	153.00	55.41
生产	670.84	670.84	0
合计	1 269.18	1 070.15	15.68

由表 5 可见, 在考虑配送时间需求条件下, 本文构建的生产与运输集成调度优化模型可以满足客户对于配送的时间需求, 根据优化对比结果发现, 只有生产成本没有变化。与没有考虑顾客

时间需求情况相比, 利用该方法运输成本降低 3.49%, 惩罚成本降低 55.41%, 总成本减少的比率为 15.68%。其中, 惩罚成本的优化率最高, 表明受生产与运输时间的影响程度最大。因此, 说明本文设计的生产与运输集成调度优化模型及改进的双自适应权重飞蛾扑火算法在求解该类问题时具有一定的优越性。

4 结论

本文针对生产与运输集成调度问题, 从最大完成时间和成本两方面进行优化, 为提高传统飞蛾扑火算法的收敛性和稳定性, 提出双自适应权重飞蛾扑火算法。主要体现在以下 3 个方面:

- (1) 引入两种自适应权重, 使局部和整体的寻优性能得到较好的平衡, 解决了原飞蛾扑火算法在解决车间调度问题时容易陷入局部最优的问题。
- (2) 改进的算法相较于另外 3 种对比算法具有更好的收敛性和稳定性, 在节约时间和降低成本方面具有一定优势。

(3) 利用提出的双自适应权重飞蛾扑火算法求得的最大完工时间和成本大幅降低, 验证了改进算法的有效性。

由于本文没有考虑碳排放因素和机器的使用率问题, 所建模型具有一定的局限性, 后续的研究工作将在现有的基础上, 进一步完善调度模型。

## 参考文献:

- [1] Caldeira R H, Gnanavelbabu A, Vaidyanathan T. An Effective Backtracking Search Algorithm for Multi-objective Flexible Job Shop Scheduling Considering New Job Arrivals and Energy Consumption[J]. Computers & Industrial Engineering, 2020, 149: 106863.
- [2] Firat M, Julie De Meyere, Martagan T, et al. Optimizing the Workload of Production Units of a Make-to-order Manufacturing System[J]. Computers & Operations Research, 2022, 138: 105530.
- [3] 陈魁, 毕利. 考虑运输时间的多目标柔性作业车间调度研究[J]. 小型微型计算机系统, 2021, 42(5): 946-952.  
Chen Kui, Bi Li. Research on Multi-objective Flexible Job Shop Scheduling Considering Transport Time[J]. Journal of Chinese Computer Systems, 2021, 42(5): 946-952.
- [4] Hidri L, Elkosantini S, Mabkhot M M. Exact and Heuristic Procedures for the Two-center Hybrid Flow Shop Scheduling Problem With Transportation Times[J]. IEEE Access, 2018, 6: 21788-21801.
- [5] Geismar H N, Laporte G, Lei Lei, et al. The Integrated Production and Transportation Scheduling Problem for a Product With a Short Lifespan[J]. INFORMS Journal on Computing, 2008, 20(1): 21-33.
- [6] Mohammadi S, Al-e-Hashem S M J M, Rekik Y. An Integrated Production Scheduling and Delivery Route Planning With Multi-purpose Machines: a Case Study From a Furniture Manufacturing Company[J]. International Journal of Production Economics, 2020, 219: 347-359.
- [7] Gharaci A, Jolai F. A Combinatorial Benders Cut for the Integrated Production Scheduling and Distribution Problem[J]. Journal of Quality Engineering and Production Optimization, 2020, 5(1): 1-18.
- [8] Moazami Goodarzi R, Ahmadizar F, Farughi H. Integrated Hybrid Flow Shop Scheduling and Vehicle Routing Problem[J]. Journal of Industrial and Systems Engineering, 2021, 13(2): 223-244.
- [9] 梁春华, 周泓, 赵健. 一种求解供应链环境下生产-配送问题的集成优化方法[J]. 控制与决策, 2011, 26(1): 27-36.
- [10] 刘玲, 李昆鹏, 刘志学. 生产和运输协同调度问题的模型和算法[J]. 工业工程与管理, 2016, 21(2): 86-91.  
Liu Ling, Li Kunpeng, Liu Zhixue. Model and Algorithms for the Integrated Production and Transportation Scheduling[J]. Industrial Engineering and Management, 2016, 21(2): 86-91.
- [11] 霍思云. 差异容量平行机环境下生产与运输集成调度算法研究[D]. 合肥: 安徽大学, 2019.  
Huo Siyun. Research on Algorithm for Integrated Scheduling of Production and Transportation on Parallel Machines With Non-identical Capacities[D]. Hefei: Anhui University, 2019.
- [12] 吴倩云, 谢乃明, 邵雨婷. 考虑时间窗和装载约束的装配线集成物流调度[J]. 计算机集成制造系统, 2020, 26(3): 806-814.  
Wu Qianyun, Xie Naiming, Shao Yuting. Integrated Logistics Scheduling of Assembly Lines With Time Windows and Loading Constraints[J]. Computer Integrated Manufacturing Systems, 2020, 26(3): 806-814.
- [13] Mirjalili S. Moth-flame Optimization Algorithm: a Novel Nature-inspired Heuristic Paradigm[J]. Knowledge-Based Systems, 2015, 89: 228-249.
- [14] 李志明, 莫愿斌. 基于Lévy飞行的飞蛾扑火优化算法[J]. 计算机工程与设计, 2017, 38(3): 807-813.  
Li Zhiming, Mo Yuanbin. Moth-flame Optimization Algorithm Based on Lévy Flights[J]. Computer Engineering and Design, 2017, 38(3): 807-813.
- [15] 李唐兵, 胡锦涛, 周求宽. 基于Lévy飞行的改进飞蛾扑火算法优化红外图像分割[J]. 红外技术, 2020, 42(9): 846-854.  
Li Tangbing, Hu Jinhong, Zhou Qiukuan. Improved Moth-flame Optimization Algorithm Based on Lévy Flight to Optimize Infrared Image Segmentation[J]. Infrared Technology, 2020, 42(9): 846-854.
- [16] Zhang Zhendong, Qin Hui, Yao Liqiang, et al. Improved Multi-objective Moth-flame Optimization Algorithm Based on R-domination for Cascade Reservoirs Operation [J]. Journal of Hydrology, 2020, 581: 124431.
- [17] Zhao Huiru, Zhao Haoran, Guo Sen. Using GM (1, 1) Optimized by MFO With Rolling Mechanism to Forecast the Electricity Consumption of Inner Mongolia[J]. Applied Sciences, 2016, 6(1): 20.
- [18] Mirjalili S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems[J]. Knowledge-Based Systems,

- 2016, 96: 120-133.
- [19] Yang Xinshe. Firefly Algorithms for Multimodal Optimization[M]//Watanabe O, Zeugmann T. Stochastic Algorithms: Foundations and Applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 169-178.
- [20] 王彤. 低碳经济下考虑配送时间需求的生产调度优化研究[D]. 重庆: 重庆大学, 2019.
- Wang Tong. Study on Production Scheduling Optimization Considering Delivery Time Demand in Low Carbon Economy[D]. Chongqing: Chongqing University, 2019.