

基于鱼骨型仓库的拣选路径问题优化

张新艳, 周雨晴

(同济大学 机械与能源工程学院, 上海 201804)

摘要: 为提高鱼骨型仓库布局下的订单拣选效率, 基于拣货路径距离计算模型和以最小化拣货路径总距离为优化目标的拣选路径优化模型, 提出一种混沌模拟退火粒子群优化算法, 引入混沌理论使粒子更高效地遍历搜寻空间, 同时结合了模拟退火算法的概率突跳特点使算法在迭代后期仍具有较好的全局寻优能力。最后, 通过实例仿真验证了该算法在解决鱼骨型仓库布局拣选路径优化问题上的有效性, 并通过与其他算法比较, 证明了该算法的先进性, 为鱼骨型仓库布局下拣选路径规划问题提供了新的解决思路。

关键词: 鱼骨型仓库布局; 拣货路径优化; 混沌理论; 模拟退火粒子群优化算法

中图分类号: TH165⁺.1

文献标志码: A

Order Picking Routing Optimization on Fishbone Aisle Warehouse

ZHANG Xinyan, ZHOU Yuqing

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

Abstract: To improve the order picking efficiency in the fishbone aisle warehouse, a chaotic SAPSO (Simulated annealing particle swarm optimization algorithm) was proposed based on the picking path distance calculation model and the picking path optimization model with the minimum distance of the picking path as the optimization objective. The chaos theory was introduced to improve the global convergence property. The SA (simulated annealing algorithm) was adopted to make the algorithm able to jump out of the local optimization and achieve global optimization. Finally, the outperformance of chaotic SAPSO algorithm to solve order picking optimization on fishbone aisle warehouse was verified by the simulation results and the comparison with other algorithms, and it provides a new solution to order picking problem in fishbone aisle warehouses.

Key words: layout of fishbone warehouse; order picking

routing optimization; chaos theory; simulated annealing particle swarm optimization

随着国家经济快速发展, 物流产业也呈现快速发展的趋势, 而仓储配送是其核心环节。为提高仓储物流管理水平, 应对仓储配送环节进行合理优化。2008 年 Hackman^[1] 提出有效进行仓储管理主要解决两个问题: 一是快速地满足季节性和供应量等需求变化, 二是整合和消除不必要的拣货路径。从仓库布局层面来说, 设计仓库布局的最新趋势是改变拣选主通道的设计, 以便更高效地进行拣选作业。2009 年美国学者 Gue 等^[2] 提出了鱼骨型仓库布局, 并证明相比传统仓库布局, 鱼骨型仓库布局可减少约 20% 的作业路径。2011 年 Pohl 等^[3] 研究非传统仓库中的货位优化问题, 并针对鱼骨型仓库提出了最佳的仓库布局方法。2013 年蒋美仙等^[4] 提出改进的鱼骨型仓库布局方法, 并设计最佳仓库布局角度。2015 年刘艳秋等^[5] 基于鱼骨型仓库布局, 建立仓储货位分配优化的数学模型, 并设计遗传算法求解。2015 年 Cardona 等^[6] 提出生成鱼骨型仓库布局的三维详细设计方法, 并通过寻找 4 个主要特征值的优化模型来最小化仓库的总运营成本。2016 年刘权等^[7] 提出基于遗传算法的仓库布局优化模型, 证明改进后的鱼骨型布局在仓库布局设计中具有更高的可行性和实用性。2017 年刘少华等^[8] 基于鱼骨型仓库布局用遗传算法、蚁群算法和布谷鸟算法对拣货路径问题进行求解。从以上研究可以看出, 鱼骨型仓库布局相对于传统仓库布局的优势所在, 大部分关于鱼骨型仓库布局的研究都处于仓库设计和仓储货位分配阶段, 少有研究涉及到拣货作业。在鱼骨型仓库布局下, 由于拣货主通道与子通道之间并非简单的平行或垂直关系, 因此用于传统仓库布局下的拣货路径模型不可直接被用于这种新型的仓库布局下。本文

收稿日期: 2019-01-25

第一作者: 张新艳(1974—), 女, 讲师, 工学博士, 主要研究方向为物流设施规划与设计。E-mail: Alicezhang@tongji.edu.cn

通信作者: 周雨晴(1994—), 女, 工学硕士, 主要研究方向为物流设施规划与设计。E-mail: 1630921@tongji.edu.cn

将建立在鱼骨型仓库布局下适用的拣选路径优化模型,同时考虑到在拣选车辆以平均速度进行拣选作业时,总作业能耗和总作业时间与总作业距离呈正相关关系,因而在此模型中以总作业距离作为优化目标。

目前常用于拣货路径优化问题的算法有:粒子群算法、模拟退火算法、遗传算法。遗传算法的操作繁杂,需要不断交叉变异,收敛速度慢,易陷入局部最优解。粒子群算法使用简单,收敛速度快,但容易陷入局部最优^[9]。模拟退火算法全局搜索能力强,但搜索速度慢。为此,2013年刘爱军等^[10]提出混沌模拟退火粒子群算法,并将算法应用在车间调度问题中。本文提出采用混沌模拟退火粒子群算法求解鱼骨型仓库布局下路径规划问题,同时在算法的相关参数选择上采取自适应调整的策略,以提高算法的效率和求解精度。

1 问题描述

本文所研究的鱼骨型仓库布局示意图如图 1 所

示。该仓库应用人到货的拣货系统。拣选作业由拣货员操作叉车进行作业,以存取货品点为起点。拣货员从不同的存储位置上收集订单上的货品^[11]。假设仓库只有一个出入点,每次完成存取作业后都必须回到出入点等待下一次的仓储作业,因此仓库的出入点就是仓库的存取点,简称 P&D (picking and deposit) 点。该仓库采用的是左右对称的仓库布局,其中有 3 条拣货主通道,3 条主通道都通过 P&D 点。

1.1 问题参数与假设条件

参数设定如下: W 为仓库的宽度; W_r 为两侧拣货通道和后部拣货通道的宽度; W_d 为两条斜拣货通道的宽度; W_h 为存储货格的长度; L 为仓库长度的一半; L_1 为拣货通道的宽度; L_2 为双排货架的宽度; α 为斜拣货通道的角度。

仓库的拣选环境假设如下:

- (1) 鱼骨型布局仓库左、右两部分关于中心对称;
- (2) 鱼骨型布局仓库拣选通道的排列方式如图 1 所示,堆垛区 1、2、3、4 拣选通道数量相等,为 m ;

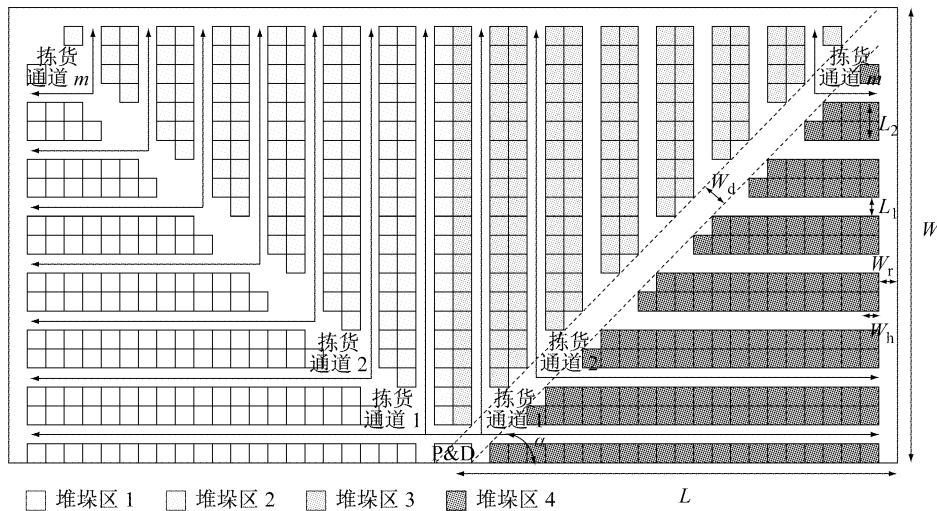


图 1 鱼骨型仓库布局

Fig.1 Layout of fishbone warehouse

- (3) 鱼骨型布局仓库斜拣货通道角度 $\alpha = 45^\circ$; 每个货格的长度、宽度、拣货通道宽度三者相等,即 $W_h = 0.5L_2 = L_1, W_r = L_1$;
- (4) 拣选车辆在拣选作业开始时从 P&D 点出发,结束后回到 P&D 点;
- (5) 拣选作业开始之前,满足所有订单的要求,且拣选作业过程中不会发生缺货现象;
- (6) 待拣选货物存储于货架之上,货架由货格组成,每个货格的长度和宽度相等,不考虑货架高度;

- (7) 拣选距离计算只考虑 P&D 与待拣货物所在位置的垂直距离,垂直方向上的拣货距离忽略不计;
- (8) 拣选距离计算不考虑拣选通道两侧货架的待拣货物所发生的左右移动距离;
- (9) 一次拣选订单总量小于拣货车辆最大承载力。

1.2 鱼骨型仓库布局拣货路径模型

基于以上鱼骨型仓库布局的参数设定以及拣选环境的假设条件,本文所建立的拣选路径优化模型如下:

目标函数:

$$D = \min(d_{01}x_{01} + \sum_{i=1}^n \sum_{j=1}^n d_{ij}x_{ij} + d_{n0}x_{n0})$$

即

$$D = \min(d_{01} + \sum_{i=1}^n \sum_{j=1}^n d_{ij}x_{ij} + d_{n0}) \quad (1)$$

约束条件:

$$\sum_{i=1}^n x_{ij} = 1, j = 1, 2, 3, \dots, n \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1, i = 1, 2, 3, \dots, n \quad (3)$$

$$\sum_{i,j \in K} x_{ij} \leq |K| - 1, |K| \subset V \quad (4)$$

$$x_{ij} \in \{0, 1\} \quad (5)$$

$$x_{01} = 1, x_{n0} = 1 \quad (6)$$

式(1)~(6)中: D 为拣选车辆完成一次拣选作业总行走距离; i, j 为任意两货位; n 为最后一个拣选货位; $d_{ij}(1 \leq i, j \leq n, i \neq j)$ 为货位*i*到货位*j*之间的最短距离; d_{01} 为P&D点到货位*i*的拣货距离,即从P&D点开始拣货作业; d_{n0} 表示货位*i*到P&D点的拣货距离,即完成拣货作业回到P&D点。

$x_{ij} =$

- { 1, 表示拣选车辆经过货位*i*到货位*j*的路径
 - { 0, 表示拣选车辆不经过货位*i*到货位*j*的路径
- $i, j = 1, 2, 3, \dots, n$

目标函数式(1)表示在一次拣选作业中最短的总行走距离;式(2)和式(3)表示所有待拣选点所在货位都被拣选且仅一次;式(4)表示除去未包含某个拣货点的路径,即不存在小回路;式(5)表示货位与货位之间是否存在拣货路径;式(6)表示拣选车辆在进行拣选作业时从P&D点出发,拣选作业结束后需回到P&D点。

1.3 鱼骨型仓库布局拣货距离计算模型

鱼骨型布局仓库中的拣选路径优化问题属于NP难问题,把目标函数设定为拣选行走路径的总距离最短.为计算上述拣货路径优化模型中的 d_{01} 、 d_{n0} 和 d_{ij} ,首先,假设任一待拣选点的序号为(s, x, y, z),其中, $s=1, 2, 3, 4$ 表示待拣选物品所在堆垛区的编号; $x=1, 2, \dots, X$,表示待拣选物品所在的拣货通道的编号, $x \leq 7$ (本文中鱼骨布局仓库的通道数,如图1所示,共7条); $y=0, 1$,表示待拣选物品所在通道位置的方位,如果物品在通道的下侧或右侧取 $y=0$,相反,如果物品在拣选通道的上侧或左侧取 $y=1$; $z=1, 2, 3, \dots, Z$,表示待拣选物品在第几个货格里,编号方法为离两侧和后部过道近的货格的编号

为1,并依次编号为1,2,3,⋯,Z.假如待拣选点的序号为(1,2,0,3),表示的是位于堆垛区1,拣货通道2,下侧的货架,从左侧过道向右数第3个货格.P&D点的位置设为(0,0,0,0),编码为0.现假设有任何两个拣选点*i, j*,序号分别为(s_i, x_i, y_i, z_i)和(s_j, x_j, y_j, z_j), d_{ij} 表示两点之间的距离。

首先,对 d_{01} 求解,数字1是指待拣选的第一个货位,可以是待拣选点中的任意一点,此处假设为1点,序号为 s_1, x_1, y_1, z_1 , d_{01} 可表示为

$$d_{01} = \left((3\sqrt{2} - 3)x_1 - z_1 + 25 - \frac{3}{2}\sqrt{2} \right) L_1 \quad (7)$$

设 d_{n0} 表示从最后一个拣选点回到P&D所行走的距离,此处假设点*n*为最后一个拣选点,点*n*的序号为 s_n, x_n, y_n, z_n ,此时, d_{n0} 可表示为

$$d_{n0} = \left((3\sqrt{2} - 3)x_n - z_n + 25 - \frac{3}{2}\sqrt{2} \right) L_1 \quad (8)$$

d_{ij} 表示任意两个拣选点*i, j*之间的距离,可表示为

(1) 当*i, j*两拣选点位于同一拣选通道时,即 $x_i = x_j$ 时

$$d_{ij} = |z_i - z_j| \quad (9)$$

(2) 当*i, j*两拣选点位于不同拣选通道时

$$d_{ij} = \min \begin{cases} (3\sqrt{2}|x_i - x_j| - 3(x_i + x_j) - (z_i + z_j) + 50)L_1 \\ (3|x_i - x_j| + (z_i + z_j)L_1 \end{cases} \quad (10)$$

$x_i, x_j \in X; z_i, z_j \in Z$

(3) 当*i, j*两拣选点位于不同堆垛区内时,分以下4种情况讨论:

① 当*i, j*两拣选点分布在堆垛区1和堆垛区4两个不同区域内时

$$d_{ij} = \min \begin{cases} (53 - 6\sqrt{2} + (3\sqrt{2} - 3)(x_i + x_j) - z_i - z_j)L_1 \\ (97 - 3x_i - 3x_j - z_i - z_j)L_1 \end{cases} \quad (11)$$

$x_i, x_j \in X; z_i, z_j \in Z$

② 当*i, j*两拣选点分布在堆垛区1和堆垛区2或堆垛区3和堆垛区4两个不同区域内时

$$d_{ij} = \min \begin{cases} (50 - 3(x_i + x_j) - z_i - z_j + 3\sqrt{2}|x_i - x_j|)L_1 \\ (50 - 6x_i - z_i + z_j + 3|x_i - x_j|)L_1 \\ (50 - 6x_j - z_j + z_i + 3|x_i - x_j|)L_1 \end{cases} \quad (12)$$

$x_i, x_j \in X; z_i, z_j \in Z$

③ 当*i, j*两拣选点分布在堆垛区1和堆垛区3或堆垛区2和堆垛区4两个不同区域内时

$$d_{ij} = \min \begin{cases} (53 + (3\sqrt{2} - 3)(x_i + x_j) - z_i - z_j)L_1 \\ (47 - 3(x_i - x_j) - (z_i - z_j)L_1 \end{cases}$$

$$x_i, x_j \in X; z_i, z_j \in Z \quad (13)$$

④ 当 i, j 两拣选点分布在堆垛区 2 和堆垛区 3 两个不同区域内时

$$d_{ij} = \min \begin{cases} (3(x_i + x_j) + z_i + z_j - 3)L_1 \\ (53 + (3\sqrt{2} - 3)(x_i + x_j) - z_i - z_j)L_1 \end{cases} \quad (14)$$

式(7)到式(14)包含了图 1 所示鱼骨型仓库布局下任意两拣选点之间距离的求解方法。

2 模型求解

拣货路径规划问题是一个 NP 难问题,因此采用 PSO (particle swarm optimization) 对该问题进行求解。在粒子群算法中,许多粒子被放在某个问题的搜索空间中,以一定的速度在搜索空间探索^[12]。在一个 D 维的目标搜索空间中,由 N 个粒子组成的粒子群落,其中第 k 个粒子的位置表示为 $\mathbf{x}_k = (\mathbf{x}_{k1}, \mathbf{x}_{k2}, \dots, \mathbf{x}_{kD})$, $k=1, 2, \dots, N$; 速度表示为 $\mathbf{v}_k = (\mathbf{v}_{k1}, \mathbf{v}_{k2}, \dots, \mathbf{v}_{kD})$, $k=1, 2, \dots, N$; 个体极值是第 k 个粒子迄今为止搜索到的最优位置,可表示为 $\mathbf{p}_k = (\mathbf{p}_{k1}, \mathbf{p}_{k2}, \dots, \mathbf{p}_{kD})$, $k=1, 2, \dots, N$, 适应度记为 p_{best} ; 全局极值是整个粒子群落迄今为止搜索到的最优位置,可表示为 $\mathbf{p}_g = (\mathbf{p}_{g1}, \mathbf{p}_{g2}, \dots, \mathbf{p}_{gD})$, $g=1, 2, \dots, N$, 适应度记为 g_{best} 。在每次迭代中,粒子的速度和位置按式(15)和式(16)更新,直到满足最大迭代次数后停止。

$$\mathbf{v}_k \leftarrow \omega \cdot \mathbf{v}_k + \varphi_1 \cdot r_1 (\mathbf{p}_k - \mathbf{x}_k) + \varphi_2 \cdot r_2 (\mathbf{p}_g - \mathbf{x}_k) \quad (15)$$

$$\mathbf{x}_k \leftarrow \mathbf{x}_k + \mathbf{v}_k \quad (16)$$

式中: ω 表示惯性权重,使其有拓展搜索空间的能力; φ_1 和 φ_2 表示学习因子,即每个粒子推向 \mathbf{p}_k 和 \mathbf{p}_g 位置的统计加速项的权重大小; r_1 和 r_2 是在 $[0, 1]$ 范围内均匀分布的随机数。

为了解决粒子群算法计算过程中振荡与过早收敛的问题,提出 SAPSO (simulated annealing particle swarm optimization), 即在粒子群算法中引入模拟退火算法的概率突跳特性,使粒子群算法不但可以接受好的解,也能以一定概率接受不好的解,以提高算法的全局搜索性能^[10]。为了提高算法的收敛速度,利用非线性自适应惯性权重 $w(t)$ 代替式(15)中的惯性权重值,其表达式为式(17),使算法在前期跳出当前极值,在后期较快收敛。为优化粒子种群运动,提高搜索空间的多样性,本文采用混沌理论对 r_1, r_2 进行动态调整。用 Logistic 模型产生混沌序

列^[10]公式(18)。

$$w(t) = \left(\frac{w_{\max} - w_{\min}}{2} \right) \cos\left(\pi \frac{t}{t_{\max}}\right) + \left(\frac{w_{\max} + w_{\min}}{2} \right) \quad (17)$$

式中: $w(t)$ 表示第 t 次迭代时的惯性权重取值; t_{\max} 表示最大迭代次数; t 表示当前迭代次数。姜建国等^[13]提出 $w_{\max}=0.95, w_{\min}=0.4$ 时算法性能最优,本文采用此取值。

$$r_q^{t+1} = 4r_q^t(1 - s_q^t), r_q \in (0, 1), q = 1, 2 \quad (18)$$

式中: r_q^t 表示 r_q 在第 t 次迭代时的值,由混沌理论可知,当 r_q^1 不等于 0.25、0.50、0.75 时,序列能呈现完全混沌的特性,变量 r_q 可以不重复地遍历整个搜索空间,以提高变量 r_q 的随机性。

混沌模拟退火粒子群算法的流程如图 2 所示。

混沌模拟退火粒子群算法的步骤如下:

步骤 1 初始化参数设定。最大迭代次数 t_{\max} , 粒子的速度 \mathbf{v}_k , 位置 \mathbf{x}_k , 学习因子 φ_1, φ_2 , 惯性权重 ω , 模拟退火因子 φ_{sa} 。

步骤 2 初始化拣货路径。随机生成一系列初始路径集合,用随机模拟方法判断初始路径群体是否满足约束条件式(2)~(6),以确保种群达到粒子群算法所需规模。

步骤 3 粒子群算子操作。按 $\varphi_1 = \varphi_1(1 - t/t_{\max}), \varphi_2 = \varphi_2(1 - t/t_{\max}), \varphi_{\text{sa}} = \varphi_{\text{sa}} \times 0.97$ 更新学习因子和模拟退火因子,同时按式(15)和(16)更新各粒子的位置和速度。

步骤 4 交换序操作。确定个体到个体最优解和全局最优解的交换顺序,混沌产生 r_1, r_2 序并和 φ_1, φ_2 比较选择执行的交换序,进行交换。

步骤 5 计算粒子适应度。针对更新后的新种群计算其适应度函数,即一次拣选作业完成行走总距离,如公式(1)所示,并更新个体的最优位置和最优适应值。

步骤 6 模拟退火操作。

① 生成模拟退火初始解 S_1 ;

② 插入扰动因子,生成模拟退火的新解 S_2 ;

③ 计算新解 S_2 的适应度函数;

④ 判断新解 S_2 的适应度函数值是否小于初始解 S_1 的适应度函数值,若不大于,则用 S_2 代替 S_1 ,并转至步骤 6;若大于,则以 φ_{sa} 的概率大小接受新解,并转至步骤 5;

⑤ 产生伪随机数 r ,并判断 φ_{sa} 是否大于 r 。若大于,则用新解 S_2 代替模拟退火初始解 S_1 ,并转至⑥;否则,则直接转至⑥;

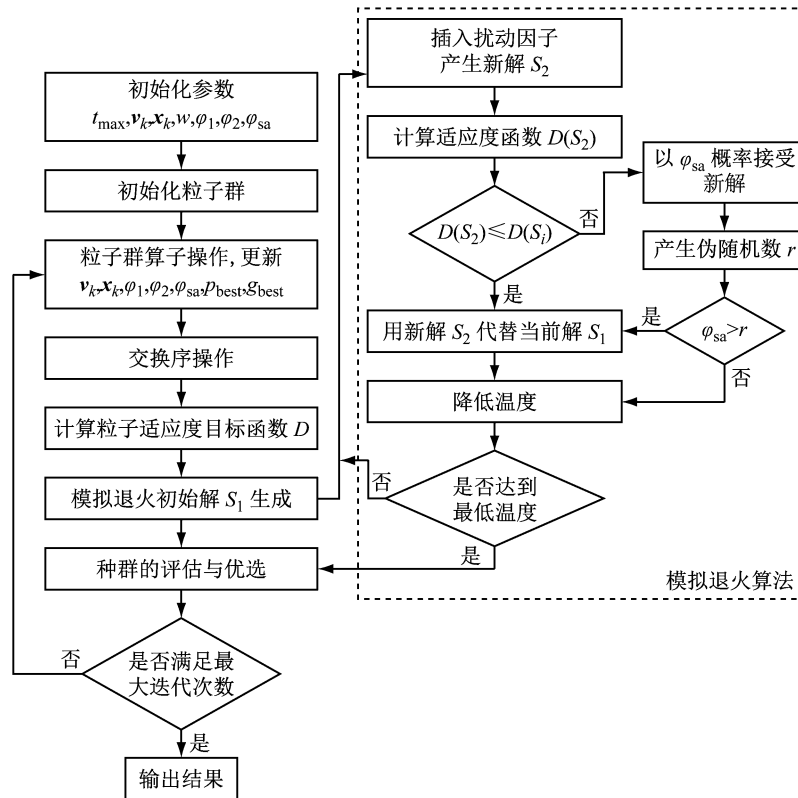


图 2 混沌模拟退火粒子群算法流程图
Fig.2 Flowchart of chaotic SAPSO

⑥ 降低温度;

⑦ 判断当前温度是否达到最低温度. 若已达到, 则退出模拟退火操作, 并转至步骤 7. 若未达到, 则返回①.

步骤 7 种群评估与优选. 根据个体的最佳适应值, 选出种群的最佳位置和最佳适应值, 并保存最佳适应值.

步骤 8 判断算法终止. 若满足最大迭代次数或目标函数减少幅度趋于收敛的预设条件, 则输出拣选作业路径总距离以及对应的具体路径方案, 算法终止; 否则, 返回步骤 3.

3 仿真及分析

仿真对象为鱼骨型仓库布局, 其各项参数取值参见 1.1 问题参数与假设条件. 由于遗传算法的全局搜索能力强, 所以选用 GAPSO (genetic particle swarm optimization)^[14] 与混沌 SAPSO 算法对比, 以验证混沌 SAPSO 算法所得优化结果是否具有全局最优的特点. 同时, PSO 算法具有收敛速度快的优点, 则选用 PSO 算法与混沌 SAPSO 算法对比迭代次数与收敛速度.

仿真实验在 Inter® Core™ i5-7300HQ, CPU

主频为 2.50 GHz、8.00 GB 内存、Windows 10 操作系统下进行, 并利用 MATLAB 2016 仿真工具编程实现. 算法参数设置如表 1 所示.

表 1 PSO、混沌 SAPSO 及 GAPSO 算法参数设置
Tab.1 Parameters of PSO, chaotic SAPSO, and GAPSO

算法	参数	问题规模			
		10	20	30	40
PSO	S	100	100	100	100
	t_{max}	50	100	200	200
	w	0.8	0.8	0.8	0.8
	φ_1	0.5	0.5	0.5	0.5
	φ_2	0.7	0.7	0.7	0.7
混沌 SAPSO	S	100	100	100	100
	t_{max}	50	100	200	200
	w	0.8	0.8	0.8	0.8
	w_{max}	0.95	0.95	0.95	0.95
	w_{min}	0.4	0.4	0.4	0.4
	φ_1	0.5	0.5	0.5	0.5
	φ_2	0.7	0.7	0.7	0.7
GAPSO	S	100	100	100	100
	t_{max}	50	100	200	200
	P_c	0.8	0.8	0.8	0.8
	P_m	0.08	0.08	0.08	0.08

注: S 表示初始种群规模; P_c 表示交叉概率; P_m 表示变异概率.

拣选点样本设置(以 10 个拣选点为例)如表 2 所示.

表 2 10 个拣选点坐标样本

Tab.2 Sample coorditions of 10 picked locations

拣选点编号	坐标	拣选点编号	坐标
0	(0,0,0,0)	6	(3,3,1,9)
1	(1,2,0,13)	7	(3,4,0,10)
2	(1,4,0,1)	8	(4,4,0,5)
3	(2,2,1,13)	9	(4,2,0,9)
4	(2,2,0,4)	10	(4,2,0,18)
5	(2,1,0,17)		

PSO、混沌 SAPSO 和 GAPSO 算法对鱼骨型仓

库布局下的拣选路径优化模型的求解结果如表 3 所示. 从表中可以看出,混沌 SAPSO 算法在 4 种问题规模下的适应度值和收敛速度都要优于 PSO 和 GAPSO 算法,可见混沌 SAPSO 算法避免了 PSO 算法易陷入局部最优的缺点,同时适应度值比 GAPSO 算法所求得的结果更优. 另外,混沌 SAPSO 相对 PSO 平均运行时间在问题规模为 10、20、30、40 时分别提高了 42. 01%、10. 13%、4. 18%、9. 29%,混沌 SAPSO 相对 GAPSO 平均运行时间分别提高了 2. 74%、39. 50%、54. 46%、53. 58%.

表 3 PSO、混沌 SAPSO 及 GAPSO 算法的性能比较

Tab.3 Performance comparison of PSO, chaotic SAPSO, and GAPSO in 10 experiments

序号	问题规模为 10						问题规模为 20					
	PSO		混沌 SAPSO		GAPSO		PSO		混沌 SAPSO		GAPSO	
	适应度值	迭代次数	适应度值	迭代次数	适应度值	迭代次数	适应度值	迭代次数	适应度值	迭代次数	适应度值	迭代次数
1	154.94	39	146.45	6	146.45	6	274.69	84	267.97	42	278.45	120
2	146.45	32	146.45	20	146.45	4	288.69	82	267.97	12	317.66	162
3	146.45	26	146.45	12	146.45	10	286.69	54	272.21	46	276.21	110
4	146.45	39	146.45	4	146.45	13	290.69	70	267.97	20	309.18	124
5	146.45	26	146.45	13	146.45	4	297.94	76	270.45	20	293.18	104
6	146.45	26	146.45	7	146.45	16	309.42	94	267.97	17	282.94	168
7	154.94	34	146.45	16	146.45	12	288.45	84	267.97	32	293.69	118
8	146.45	40	146.45	20	146.45	3	306.21	64	267.97	13	290.45	128
9	146.45	39	146.45	13	146.45	6	274.21	58	267.97	12	289.18	100
10	156.45	32	146.45	10	146.45	5	297.45	64	270.21	50	293.42	162
平均值	149.15	34	146.45	12	146.45	8	291.44	73	268.86	26	292.44	129
最小值	146.45	26	146.45	4	146.45	3	274.21	54	267.97	12	267.21	100

序号	问题规模为 30						问题规模为 40					
	PSO		混沌 SAPSO		GAPSO		PSO		混沌 SAPSO		GAPSO	
	适应度值	迭代次数	适应度值	迭代次数	适应度值	迭代次数	适应度值	迭代次数	适应度值	迭代次数	适应度值	迭代次数
1	327.91	72	306.42	56	357.15	152	517.46	132	329.18	28	436.66	168
2	398.42	76	306.42	29	386.12	173	447.31	130	344.94	33	425.58	156
3	418.42	76	306.42	58	378.66	186	517.92	124	352.18	77	413.94	118
4	377.39	75	310.45	11	389.94	142	499.28	118	360.42	32	367.15	163
5	396.82	86	303.18	26	385.39	176	522.46	126	340.94	69	442.91	121
6	376.86	78	318.18	33	379.91	174	471.93	100	333.91	34	351.63	126
7	338.45	94	306.94	16	374.12	184	464.18	124	341.66	62	394.63	164
8	382.33	98	304.18	22	342.45	186	502.03	84	351.69	28	381.88	162
9	400.42	74	306.42	23	363.91	108	434.52	142	338.94	50	418.91	141
10	416.88	98	310.18	19	335.45	124	488.30	113	326.69	82	375.66	142
平均值	383.39	83	307.88	30	369.31	161	486.54	120	342.06	50	400.90	147
最小值	327.91	72	303.18	11	333.45	108	434.52	84	326.69	28	351.63	118

PSO、混沌 SAPSO 和 GAPSO 算法的适应度曲线如图 3 所示. 从图中可以看出,在 4 种问题规模下混沌 SAPSO 算法的收敛速度都比 PSO 和 GAPSO 算法要快,同时振荡现象得到了明显改善.

混沌 SAPSO 算法求解所得路径解的拣选顺序如下:

10 个拣选点顺序为 0→10→9→8→7→6→4→3→1→2→5→0.

20 个拣选点顺序为 0→1→11→10→6→7→5→

4→3→2→8→9→13→14→16→17→15→18→19→10→12→0.

30 个拣选点顺序为 0→15→16→5→12→11→9→10→8→6→7→4→3→13→14→23→24→25→22→21→27→26→28→29→30→2→1→20→19→17→18→0.

40 个拣选点顺序为 0→22→39→40→23→21→24→25→26→36→34→28→27→29→30→31→32→33→35→38→37→2→1→18→3→4→7→6→5→16

→14→13→11→9→10→8→12→15→17→19→20→ 4 所示.

0. 以 10 个拣选点为例,优化后拣货路径示意图如图

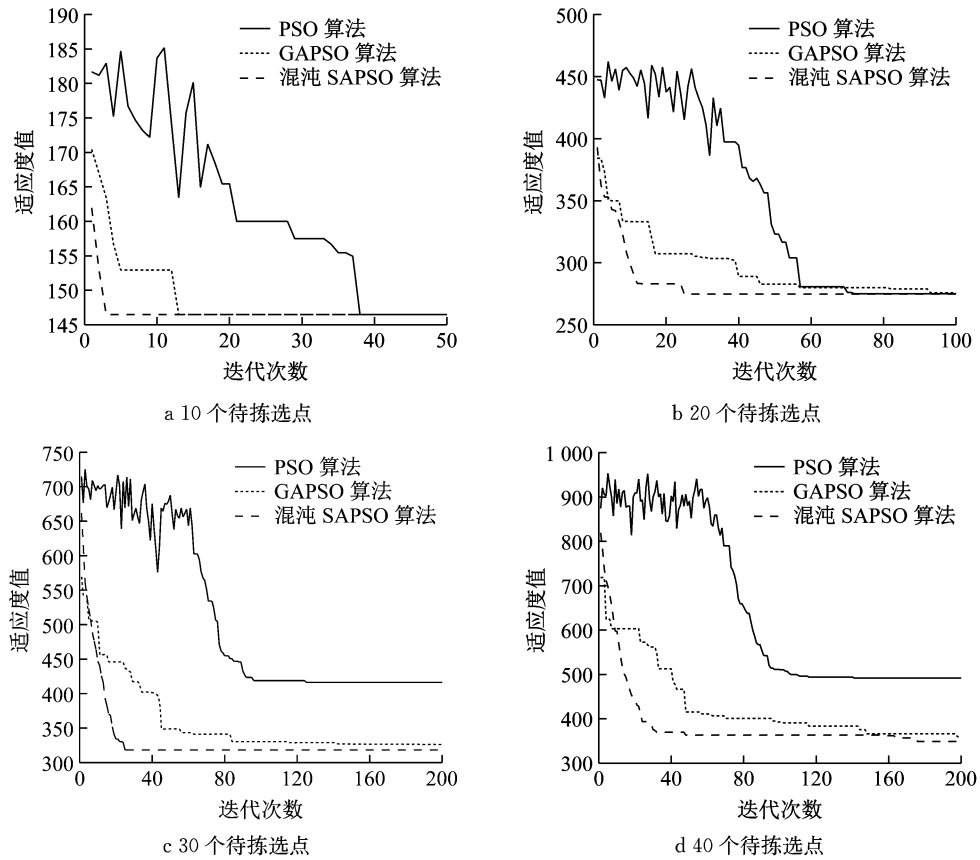


图 3 PSO、混沌 SAPSO 及 GAPSO 算法适应度曲线比较

Fig. 3 Fitness comparison of PSO, chaotic SAPSO, and GAPSO in 10 experiments

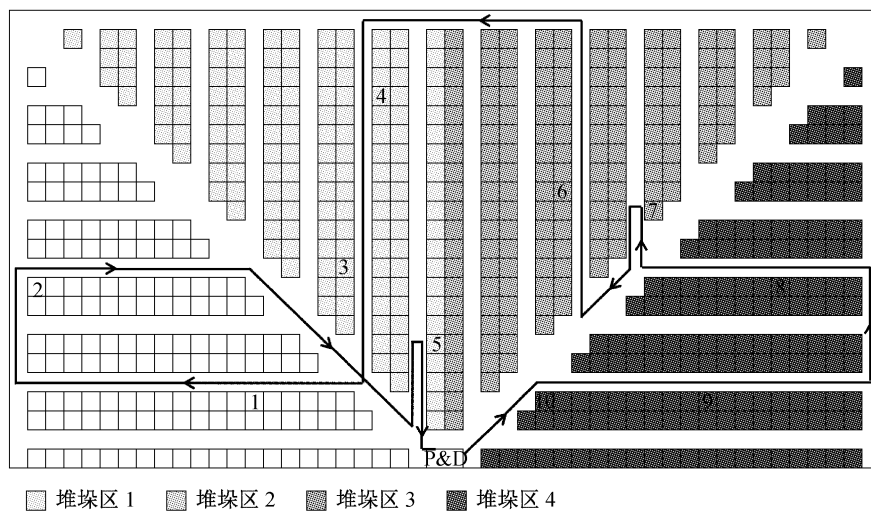


图 4 10 个拣选点的拣货路径示意图

Fig. 4 Order picking route of 10 picked locations

4 结论

本文的研究基于鱼骨型仓库布局的拣货路径问

题,对建立的拣货距离计算和拣货路径优化模型,采用混沌模拟退火粒子群算法进行求解,混沌模拟退火粒子群算法采用混沌理论对粒子群优化算法中的参数 r_1 和 r_2 进行动态调整,同时引入模拟退火算法

对粒子群优化算法中的每个粒子位置局部寻优,并设置不同的问题规模,对算法的性能进行了分析。仿真结果表明,混沌模拟退火粒子群算法在适应度值、迭代次数、收敛速度、运行时间和振荡效果方面都优于粒子群优化算法和遗传粒子群算法。

本文的拣货距离计算和拣货路径优化模型可为鱼骨型仓库布局的研究提供决策参考,在后续研究中可进一步考虑鱼骨型仓库布局下的订单分批问题。

参考文献:

- [1] HACKMAN S T. Allocating space in a forward pick area of a distribution center for small parts[J]. IIE Transactions, 2008, 40(11): 1046.
- [2] GUE K R, MELLER R D. Aisle configurations for unit-load warehouses[J]. IIE Transactions, 2009, 41(3): 171.
- [3] POHL L M, MELLER R D, GUE K R. Turnover-based storage in non-traditional unitload warehouse designs [J]. IIE Transactions, 2011, 43(10): 703.
- [4] 蒋美仙, 冯定忠, 赵宴林, 等. 基于改进 Fishbone 的物流仓库布局优化[J]. 系统工程理论与实践, 2013, 33(11): 2920. JIANG Meixian, FENG Dingzhong, ZHAO Yanlin, *et al.* Optimization of logistics warehouse layout based on the improved Fishbone layout[J]. System Engineering -Theory & Practice, 2013, 33(11): 2920.
- [5] 刘艳秋, 张义华, 焦妮. 基于 Fishbone 的仓储货位分配优化[J]. 物流科技, 2014, 37(12): 66. LIU Yanqiu, ZHANG Yihua, JIAO Ni. Slotting optimization allocation of storage based on fishbone[J]. Logistics Sci-Tech, 2014, 37(12): 66.
- [6] CARDONA L F, SOTO D F, RIVERA L. Detailed design of fishbone warehouse layouts with vertical travel [J]. International Journal of Production Economics, 2015, 170(C): 825.
- [7] 刘权, 杨鹏辉, 刘润茜, 等. 基于遗传算法的仓库布局优化模型及最优角度的确定[J]. 河北北方学院学报, 2016, 32(3): 21. LIU Quan, YANG Penghui, LIU Runqian, *et al.* Optimization model of warehouse layout and determination of optimal angle based on genetic algorithm[J]. Journal of Hebei University (Natural Science Edition), 2016, 32(3): 21.
- [8] 刘少华. 多种智能算法在鱼骨布局拣选路径决策中的比较研究[D]. 北京: 北京物资学院, 2017. LIU Shaohua. A comparative study on multi-intelligence algorithm in route selection of fishbone layout[D]. Beijing: Beijing Wuzi University, 2017.
- [9] 陈雪. 基于改进粒子群算法的 A 集团自动化立体仓库的优化研究[D]. 北京: 北京交通大学, 2018. CHEN Xue. Research on optimization of a group automated warehouse based on improved particle swarm optimization algorithm[D]. Beijing: Beijing Jiaotong University, 2018.
- [10] 刘爱军, 杨育, 李斐, 等. 混沌模拟退火粒子群优化算法研究及应用[J]. 浙江大学学报(工学版), 2013, 47(10): 1722. LIU Aijun, YANG Yu, LI Fei, *et al.* Chaotic simulated annealing particle swarm optimization algorithm research and its application[J]. Journal of Zhejiang University(Engineering Science), 2013, 47(10): 1722.
- [11] 李乐, 曾德贵. 贯通式货架系统和 Fishbone 布局方法结合的物流仓库布局优化[J]. 物流技术, 2015, 34(2): 201. LI Le, ZENG Degui. Layout optimization of logistics warehouse based on drive-in rack system and fishbone planning[J]. Logistics Technology, 2015, 34(2): 201.
- [12] 张晓东, 王茜. 多目标服务工作流混合粒子群调度算法[J]. 东南大学学报(自然科学版), 2010, 40(3): 491. ZHANG Xiaodong, WANG Qian. Hybrid particle swarm optimization algorithm for multi-objective scheduling in service work flows [J]. Journal of Southeast University (Natural Science Edition), 2010, 40(3): 491.
- [13] 姜建国, 田旻, 王向前, 等. 采用扰动加速因子的自适应粒子群优化算法[J]. 西安电子科技大学学报, 2012, 39(4): 74. JIANG Jianguo, TIAN Min, WANG Xiangqian, *et al.* Adaptive particle swarm optimization via disturbing acceleration coefficients [J]. Journal of Xidian University, 2012, 39(4): 74.
- [14] 樊明, 郭艺, 负超, 等. 基于自适应混合算法的智能存取系统动态路径规划[J]. 系统仿真学报, 2013, 25(7): 1543. FAN Ming, GUO Yi, YUN Chao, *et al.* Adaptive hybrid algorithm for dynamic path planning problem of intelligent access system[J]. Journal of System Simulation, 2013, 25(7): 1543.