

[环保·安全]

DOI:10.3969/j.issn.1005-2895.2023.04.012

基于海鸥算法的库区货位分配及优化研究

吴伟开^{1,2}, 何雪明^{1,2}, 黄育鹏^{1,2}

(1. 江南大学 机械工程学院, 江苏 无锡 214122;
2. 江南大学 江苏省食品先进制造装备技术重点实验室, 江苏 无锡 214122)

摘要:针对自动化立体仓库库区货位分配存在的出入库速度慢、货架稳定性差以及货品相关性低的问题,课题组提出了一种优化方法。首先,进行数学模型的搭建,为各目标函数分配权重,对比较重要的目标函数进行人为增量,提升目标函数的计算值敏感程度,达到自我分辨的目的;其次,通过引入非线性动态移动参数,提出一种改进的海鸥算法,并采用6种基准测试函数来验证改进算法的寻优能力。结果表明改进后的算法在精度以及寻优速度上都优于原始算法。该模型能够提升货物的出入库速度和货品相关性,保证货架稳定性,可有效提升立体仓库利用率。

关键词:立体仓库;货位分配;库区优化;海鸥算法

中图分类号:TH187 文献标志码:A 文章编号:1005-2895(2023)04-0091-08

Research on Warehouse Space Allocation and Optimization Based on Seagull Algorithm

WU Weikai^{1,2}, HE Xueming^{1,2}, HUANG Yupeng^{1,2}

(1. School of Mechanical Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China;
2. Jiangsu Provincial Key Laboratory of Advanced Food Manufacturing Equipment Technology,
Jiangnan University, Wuxi, Jiangsu 214122, China)

Abstract: An optimization method was proposed to solve the problems such as the low speed of loading and unloading, the low stability of shelves and the low correlation of goods. First of all, the mathematical model was built, weights were assigned to each objective function, and artificial increment was made to the more important objective function to improve the sensitivity of the calculated value of the objective function, so as to achieve the purpose of self-discrimination. Secondly, by introducing nonlinear dynamic motion parameters, an improved seagull algorithm was proposed, and six kinds of benchmark test functions were used to verify the optimization ability of the improved algorithm. The results show that the improved algorithm is superior to the original algorithm in result accuracy and optimization speed. This model can improve the speed of goods in and out of storage and the correlation of goods, ensure the stability of shelves, and effectively improve the utilization rate of three-dimensional warehouse.

Keywords: three-dimensional warehouse; space allocation; reservoir optimization; seagull algorithm

自动化立体仓库的自动存储/检索系统(automatic storage/retrieval system, AS/RS)能够根据指令自动完成货物存取和记录,对库存进行自动管理。与传统仓库相比,自动化立体仓库具有储货量大、占地面积小、

出入库迅捷和企业管理与生产效率提高等优点^[1]。针对自动化立体仓库的货位分配优化问题,国内外的许多专家学者进行了大量的探索和深究。

Wang等^[2]研究了鱼骨机器人移动过滤系统

收稿日期:2022-11-26;修回日期:2023-04-20

基金项目:国家自然科学基金(51275210);国家自然科学基金(51975251);江苏省食品先进制造装备技术重点实验室自主研究课题资助项目(FMZ2018Y2);“六大人才高峰”(1076010241131350)。

第一作者简介:吴伟开(1997),男,安徽黄山人,硕士研究生,主要研究方向为滚动轴承动态性能、信息管理系统。E-mail: 986121511@qq.com

(fishbone robot mobile filtration system, FRMFS) 的存储分配优化问题,设计了一种自适应遗传算法(adaptive genetic algorithm, AGA)对提出的模型进行求解,并验证了其有效性。Lanza 等^[3]提出具有一些有效不等式的混合整数线性规划(mixed integer linear programming, MILP)模型,并将该模型用在仓库的货位分配中。Yuan 等^[4]研究了机器人移动履约系统(robotic mobile fulfillment system, RMFS)的存储分配优化问题,设计了贪心算法与改进模拟退火相结合的2阶段混合算法对模型进行求解。Tu 等^[5]提出了一种基于云的信息物理系统(cyber-physical system, CPS)的拣选通行系统(pick-and-pass system, PKPS),采用启发式多目标遗传算法求解基于电子商务的仓库拣货作业的 NP-hard 存储指派问题(storage assignment problem, SAP)。Keung 等^[6]提出了一种基于数据驱动的 RMFS 区域聚类 and 存储位置分配分类方法。Mirzaei 等^[7]提出了一种基于产品周转率和从历史客户订单中获取的亲合力集成簇分配(integrated cluster allocation, ICA)策略,以最小化零件到拣货系统的检索时间。Tabatabaei 等^[8]针对现代仓储中自动化存储和检索过程,提出了一种新的存储分配优化算法(storage assignment optimization technique, SAO),以实现存储机制的敏捷性。He 等^[9]提出了一种新的粒子群优化(particle swarm optimization, PSO)方法解决出库箱位指派问题。

有关货位分配方面的研究大多集中在对算法进行改进,较少对货位分配的模型进行分析和研究。对于仓储中普遍存在的出入库速度慢、货架稳定性差及货

品相关性低的问题,课题组将智能优化算法与数学模型结合起来解决实际仓储过程中存在的货位分配问题。

1 库区货位分配问题描述

对于自动化立体仓库,入库时存储位置就是出库时拿取的位置,因此在进行储位分配规划时只需确定出库时的条件即可。特定情况下,可能需要对所有位置进行一次重新分配,此时存储位置可能发生变化。

货位分配时,一般要遵守下面这几个准则:最先进入仓库的产品,也应当最先出去,减少产品的在库时间,周转率高的产品应当放在离出库口较近的位置;为使得货架使用寿命较长,应采用“上轻下重”的摆放方式;尽量将相关性较大的产品安放位置靠近。

因此,货位分配的目标可以总结为提高货品的出入库效率、保证货架的稳定性、提高存放货品之间的相关性。课题组针对产品的出库货位分配问题,根据上面总结的3个目标来进行货位分配的研究。

2 库区货位分配的优化数学模型

2.1 库区货位分配模型参数

模型参数是在货位分配模型中需要用到的各种变量和常量,应根据实际作业的情况赋予这些参数相应的值,货位分配模型的参数定义如表1所示。

2.2 库区货位分配模型目标函数

2.2.1 货品出库效率

由于立体仓库中的堆垛机速度是恒定的,可以通过缩短产品位置到出库口之间的距离(即缩短产品出库时间)来提高产品的出库效率。一般情况下,堆垛机

表1 模型参数

Table 1 Model parameters

参数	说明	参数	说明	参数	说明
a	货架的排数	l	单元货格的长度	M_i	第 i 类产品的单位质量
b	货架的列数	w	单元货格的宽度	$n_{xyz,i}$	货位 (x,y,z) 上第 i 类产品的数量
c	货架的层数	h	单元货格的高度	m_i	第 i 类产品的坐标均值
x	货位的排号	H	巷道的宽度	(x_{ij}, y_{ij}, z_{ij})	第 i 类产品的第 j 个所在储位坐标
y	货位的列号	f_{xyz}	货位 (x,y,z) 上产品的周转率	d_{ij}	第 i 类产品的第 j 个到 m_i 的距离
z	货位的层号	n	所有产品共 n 类	d	所有 n 类产品的类内离散度距离和
v_X	输送机的运输速度	i	第 i 类产品, $i \in [1, n]$	m	所有 n 类产品的均值坐标
v_Y	堆垛机沿 y 方向运输速度	k_i	第 i 类产品的个数	D	所有产品到 n 类产品中心 m 的离散度和
v_Z	堆垛机沿 z 方向运输速度	j	第 i 类产品的第 j 个, $j \in [1, k_i]$	D_m	所有产品均值坐标到出库口的距离

在巷道中运行是2个方向同时运动,因此有2个速度:一个是纵向的,另一个是横向的。根据这种实际情况,则需要得到这2个方向作业过程中花费的时间的最大值,即 $\max(y \cdot l/v_y + z \cdot h/v_z)$ 。堆垛机在运输带上所花费的时间可表示为 x/v_x 。则该模型的出库效率函数公式为:

$$\min f_1(x, y, z) = \sum_{x=1}^a \sum_{y=1}^b \sum_{z=1}^c t_{xyz} \cdot f_{xyz} \quad (1)$$

$$\text{其中, } t_{xyz} = \frac{x}{v_x} + \max\left(\frac{y \cdot l}{v_y} + \frac{z \cdot h}{v_z}\right)。$$

2.2.2 货品的相关性分析

将一起入库的相关性大的产品放在较为靠近的位置,并归为一类品种,想要得到这样的效果就要使得相关产品之间距离之和累加最小。

类内的离散度主要是指同品种的2个产品摆放的距离,那么用数学抽象化就是2个点之间的距离。将产品坐标除以产品数量就可以得到该类产品的坐标均值,即所有第*i*类货品的坐标均值为:

$$m_i = \frac{1}{k_i} \sum_{j=1}^{k_i} (x_{ij}, y_{ij}, z_{ij}) \quad (2)$$

计算同一品种的一个产品的坐标与这个类坐标均值之间的距离,即第*i*类货品的第*j*个到 m_i 的距离,公式表示为:

$$d_{ij} = \sqrt{(x_{ij} - m_i(x))^2 + (y_{ij} - m_i(y))^2 + (z_{ij} - m_i(z))^2} \quad (3)$$

将算出来的距离值进行累加,所得的值越小则说明同一品种的产品摆放得越集中,即货品之间的类内离散度 d 为所有 n 类货品到 m_i 的距离之和:

$$d = \sum_{i=1}^n \sum_{j=1}^{k_i} d_{ij} \quad (4)$$

最后计算类间离散度。同理先将所有种类的货品的类坐标均值算出来,即所有 n 类货品的均值坐标为:

$$m = \frac{1}{n} \sum_{i=1}^n m_i \quad (5)$$

然后将所有货品到 n 类货品均值坐标 m 的距离累加之和作为货品类间离散度之和:

$$D = \sum_{i=1}^n (m_i - m) = \sum_{i=1}^n \sqrt{(m_i(x) - m(x))^2 + (m_i(y) - m(y))^2 + (m_i(z) - m(z))^2} \quad (6)$$

综合考虑这2种离散度,才能使所摆放的货品之间的相关性最大。即货品应均匀地摆放在货架中,而且 m 点应尽量离出库口较近。

$$D_m = \sqrt{m(x)^2 + m(y)^2 + m(z)^2} \quad (7)$$

可以建立货品相关性的目标函数:

$$\min f_2 = \frac{d + D_m}{D} \quad (8)$$

2.2.3 货架稳定性

考虑到货物存放的安全问题,在存放物品时应当将重的产品放在货架的靠下层,较轻的产品则放在货架的较上层,用数学的方法来抽象就是每一层的货品的质量、数量和所在高度的乘积除以每一层的货品的质量和数量的乘积所得到的值越小,则重心越低,即目标函数公式如下:

$$\min f_3(x, y, z) = \frac{\sum_{x=1}^a \sum_{y=1}^b \sum_{z=1}^c M_i \cdot n_{xyz} \cdot z}{\sum_{x=1}^a \sum_{y=1}^b \sum_{z=1}^c M_i \cdot n_{xyz}} \quad (9)$$

综上所述,货位分配的数学模型表达式为:

$$\left. \begin{aligned} \min f_1(x, y, z) &= \sum_{x=1}^a \sum_{y=1}^b \sum_{z=1}^c \left(\frac{x}{v_x} + \max\left(\frac{y \cdot l}{v_y} + \frac{z \cdot h}{v_z}\right) \right) \cdot f_{xyz}; \\ \min f_2 &= \frac{d + D_m}{D}; \\ \min f_3(x, y, z) &= \frac{\sum_{x=1}^a \sum_{y=1}^b \sum_{z=1}^c M_i \cdot n_{xyz} \cdot z}{\sum_{x=1}^a \sum_{y=1}^b \sum_{z=1}^c M_i \cdot n_{xyz}} \end{aligned} \right\} \quad (10)$$

在上述的货位分配模型函数公式中,目标函数式(1)保证了货品的出库效率,同时也缩短了周转率较高的产品的出库时间;目标函数式(8)使得在货架中放入的相关性较大或者同一品种的产品能够放在一起,保证了产品之间的相关性;目标函数式(9)约束了入库产品摆放按照质量“上小下大”的原则进行摆放。最后将这几个目标函数统一在一起,得到最后的目标函数式(10)。该问题属于一种复杂的多项式问题,可以采用海鸥算法对该问题进行优化求解。

2.3 库区货位分配模型约束条件

企业成品库存储区共有 a 排货架,每排货架共有 c 层 b 列。因此货位坐标的 (x, y, z) 值应当不能超过 $(a,$

b, c) 的值。同时货位坐标应为整数。所以模型的约束条件如下:

$$\text{s. t. } \begin{cases} 1 \leq x \leq a, \\ 1 \leq y \leq b, \\ 1 \leq z \leq c; \end{cases} \quad (11)$$

x, y, z 为整数。

联立式(10)和(11),可得货位分配模型。

上述模型是根据实际问题建立的,但是在实际运用与计算的时候发现根据该模型得到的结果并不是很好,为解决这一问题,提出为各目标函数分配权重,以及对比较重要的目标函数进行人为增量的方法,使该目标函数得到的数值再乘以一个固定的敏感系数,使得函数值增大。通过这样的方法使程序对该目标函数的计算值敏感,然后达到自我分辨的目的。

对于目标 1 的函数,乘以一个固定的敏感系数 10,得到修改后的公式为:

$$\min f_1(x, y, z) = \sum_{x=1}^a \sum_{y=1}^b \sum_{z=1}^c \left(\frac{x}{v_x} + \max \left(\frac{y \cdot l}{v_y} + \frac{z \cdot h}{v_z} \right) \right) \cdot f_{xyz} \cdot 10. \quad (12)$$

对于目标 2 的函数,在计算类内离散度的公式中乘以一个固定的敏感系数 50,可得到修改后的公式:

$$d = \sum_{i=1}^n \sum_{j=1}^{k_i} (d_{ij} \cdot 50). \quad (13)$$

则目标 2 的函数公式为:

$$\min f_2 = \frac{\sum_{i=1}^n \sum_{j=1}^{k_i} (d_{ij} \cdot 50) + D_m}{D}. \quad (14)$$

将各个目标函数加起来得到一个总的目标函数,再为各目标函数赋予合适的权重得到总目标函数公式:

$$F_{\text{total}} = (\min f_1 + \min f_2 + \min f_3) \cdot w. \quad (15)$$

式中 w 为权重数组 $[0.3, 0.6, 0.1]$ 。

3 库区货位分配的海鸥算法

海鸥算法(seagull optimization algorithm, SOA)是 Gaurav Dhiman 和 Vijay Kumar 通过研究海鸥的生物行为时提出的^[10],能够使用相应的数学模型来表征这些行为,该算法的模型特征主要包括觅食迁徙和攻击猎物的行为,如图 1 所示。

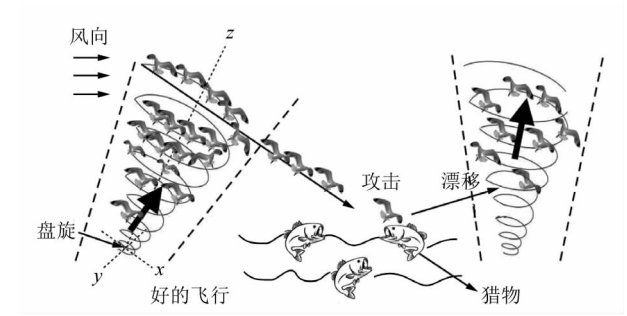


图 1 海鸥算法模型

Figure 1 Seagull optimization algorithm model

3.1 海鸥优化算法特征

3.1.1 迁徙特征

迁徙模型就是表示海鸥在全体移动寻找食物时产生的一种迁徙行为,在这种大范围,并且个体较为紧密的迁徙过程中,海鸥应当满足以下几个条件:

1) 海鸥之间避免相互碰撞。海鸥们的位置不能够重叠,因此为防止海鸥之间产生相互碰撞,在位置更新的过程中加入一个移动参数 A 。海鸥避免相互碰撞如图 2 所示,公式如下:

$$C_s = A \cdot P_s(X). \quad (16)$$

式中: $A = f_c - (X \cdot (f_c / M_q))$; $X = 0, 1, 2, \dots, M_q$; C_s 为新的海鸥位置,与其他海鸥位置不产生冲突; P_s 为当前海鸥位置; X 为到目前位置迭代更新次数; A 为海鸥在迁徙过程中搜索猎物的移动参数; f_c 为控制移动参数的变化, f_c 一般取 2; q 为迭代更新次数; M_q 为设置的总迭代更新次数。

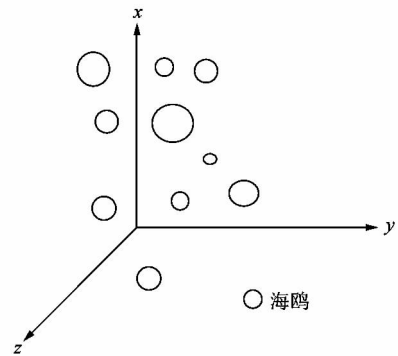


图 2 海鸥之间避免相撞过程

Figure 2 Collision avoidance process between seagulls

2) 向目前离猎物最近的海鸥靠近。在解决碰撞的问题后,海鸥们会向目前最靠近猎物的海鸥靠近,公

式如下:

$$\left. \begin{aligned} M_s &= B \cdot (P_{\text{best}}(x) - P_s(x)), \\ B &= 2 \cdot A^2 \cdot R. \end{aligned} \right\} \quad (17)$$

式中: M_s 为当前海鸥向最靠近猎物海鸥的位置飞行方向; B 为随机变量, 扰动新的海鸥的搜寻位置, 增强全局寻优能力; R 为一个随机数, 范围是 $[0, 1]$; $P_{\text{best}}(X)$ 为当前最靠近猎物的海鸥位置。

其他海鸥向最靠近猎物的海鸥位置靠近的过程如图 3 所示。

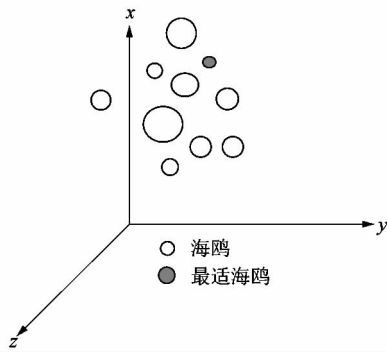


图 3 海鸥向最佳海鸥移动

Figure 3 Movement of seagull towards optimal seagull

3) 靠近最佳海鸥后更新位置。当向最佳海鸥靠近后, 为避免碰撞会更新自己的位置且进行其他位置的探索, 公式为:

$$D_s = |C_s + M_s|。 \quad (18)$$

式中 D_s 为当前海鸥与最佳海鸥之间的距离。

海鸥向最佳海鸥方向靠近的过程并更新位置描述如图 4 所示。

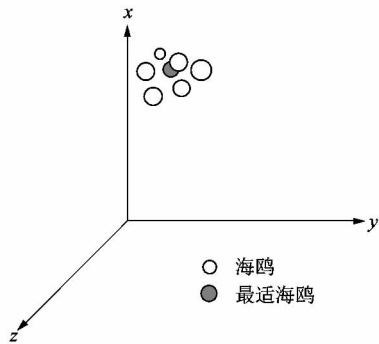


图 4 靠近最佳海鸥后位置

Figure 4 Close to optimum position behind seagull

3.1.2 攻击特征

海鸥们在捕食时的主要的行为特点是以一种螺旋式的飞行轨迹进行捕食, 不断控制改变下降的角度以及飞行的半径, 如图 5 所示。其数学描述如下:

$$\left. \begin{aligned} X' &= r \cdot \cos k; \\ y' &= r \cdot \sin k; \\ z' &= r \cdot k; \\ r' &= u \cdot e^{kv}; \\ P_{s+1}(X) &= (D_s(X) \cdot X' \cdot y \cdot z) + P_{\text{best}}(X). \end{aligned} \right\} \quad (19)$$

式中: r 为在下降过程中的飞行半径; k 为 $[0, 2\pi]$ 内的一个随机值; u, v 为螺旋飞行行为的常数变量; e 为自然对数的底数; $P_s(X)$ 为当前更新后的最优海鸥位置。

海鸥算法在迭代更新的过程中, 海鸥们能够向当前的最优海鸥靠近并更新自己的位置, 增强全局的探索能力。

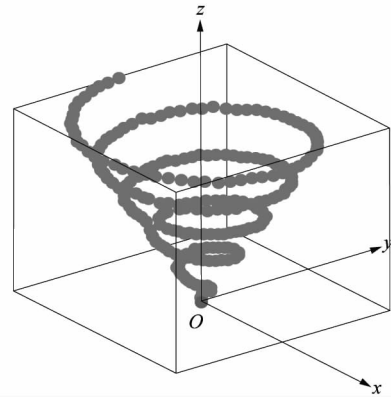


图 5 捕猎模型

Figure 5 Hunting model

3.2 海鸥优化算法流程

算法流程如下:

- 1) 代入初始变量的值进行计算;
- 2) 根据式(11)~(18)计算海鸥当前位置;
- 3) 计算每只海鸥个体的适应度值;
- 4) 根据式(19)更新海鸥的位置;
- 5) 计算目前最优海鸥的位置以及适应度值, 将迭代数加 1, 进行下次迭代;
- 6) 若当前迭代数小于最大的迭代数, 则执行下一步流程, 否则跳转至 3);

7) 得到最后最优的海鸥位置。

3.3 海鸥优化算法改进

通过对 SOA 算法的原理以及数学公式推导可以得知:SOA 算法的模型参数很少,算法的流程也较为简单易懂,比较容易实现,并且能够较好地模拟海鸥的迁徙和捕食行为;但是在迁徙的过程中并不能很好地更新自己的位置,这样就会使得算法在运行初期收敛较慢或者陷入局部最优。因此通过修改移动参数的方式设计了非线性动态移动海鸥算法(parallel seagull optimization algorithm, PSOA),该想法是从国内外学者对粒子群算法进行改进的方法中产生的。课题组将这 2 种算法的性能进行了对比,改进后的算法增强了算法的收敛速度,保证了算法的简易性和精确性。

引入非线性动态移动参数,改进后海鸥算法的移动参数 A 更新公式为:

$$A = f_c - f_c \cdot \left(\sec \left(\left(\frac{X}{M_q} \right)^2 \cdot \frac{\pi}{3} \right) - 1 \right), X = 0, 1, 2, \dots, M_q. \quad (20)$$

改进后的移动参数 A 能够非线性地从 f_c 变化到 0,这样使得海鸥在更新位置时能够扩大全局的搜索能力,同时随着 A 不断地变小,群体不断地向最优个体位置靠近,增强了算法的局部搜索能力。改进后的移动参数 A 与原移动参数 A 对比的趋势如图 6 所示。

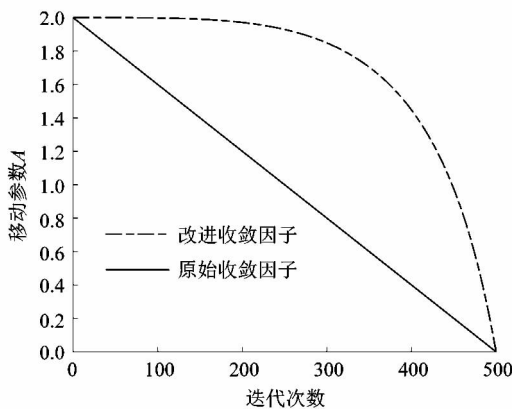


图 6 移动参数 A 随迭代次数变化趋势

Figure 6 Changes of movement parameter A with number of iterations

3.4 改进后海鸥优化算法流程

图 7 所示为改进后的海鸥算法执行流程。

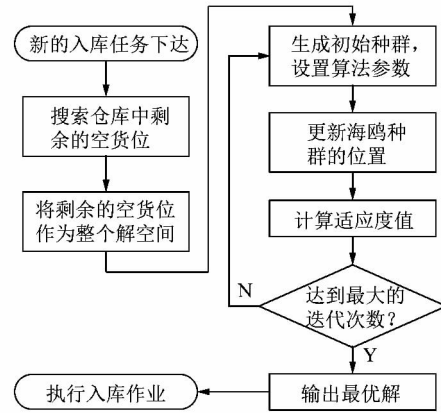


图 7 执行流程图

Figure 7 Execution flow chart

将式(14)改为式(20)代入计算海鸥当前位置,其中计算适应度值就是计算货位分配模型的总目标函数,将生成的货位坐标代入函数中即可。

3.5 性能测试

为了验证 PSOA 的算法的性能,选取 6 个常用的基准测试函数对 PSOA, SOA 和 PSO 进行性能对比测试,验证改进后的性能。

其中函数 $f_1 \sim f_3$ 为单峰测试函数, $f_4 \sim f_6$ 为多峰测试函数。单峰测试函数可以用来评估算法的开发能力以及寻优能力,多峰函数可以用来评估算法的全局寻优能力。单峰测试函数见式(21)~(23),多峰测试函数见式(24)~(26)。

1) Sphere 函数

$$\left. \begin{aligned} f_1(x) &= \sum_{i=1}^n x_i^2; \\ \min f_1(x) &= f_1(0, 0, \dots, 0) = 0. \end{aligned} \right\} \quad (21)$$

2) Schwefel 函数

$$\left. \begin{aligned} f_2(x) &= \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2; \\ \min f_2(x) &= f_2(0, 0, \dots, 0) = 0. \end{aligned} \right\} \quad (22)$$

3) Rosenbrock 函数

$$\left. \begin{aligned} f_3(x) &= \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]; \\ \min f_3(x) &= f_3(0, 0, \dots, 0) = 0. \end{aligned} \right\} \quad (23)$$

4) Rastrigin 函数

$$\left. \begin{aligned} f_4(x) &= \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]; \\ \min f_4(x) &= f_4(0,0,\dots,0) = 0. \end{aligned} \right\} \quad (24)$$

5) Ackley 函数

$$\left. \begin{aligned} f_5(x) &= -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e; \\ \min f_5(x) &= f_5(0,0,\dots,0) = 0. \end{aligned} \right\} \quad (25)$$

6) Griewank 函数

$$\left. \begin{aligned} f_6(x) &= \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1; \\ \min f_6(x) &= f_6(0,0,\dots,0) = 0. \end{aligned} \right\} \quad (26)$$

单峰函数测试结果见图 8~10,多峰函数测试结果见图 11~13,详细的结果对比如表 2 所示。实验设置种群初始规模为 50,迭代次数为 1 000。

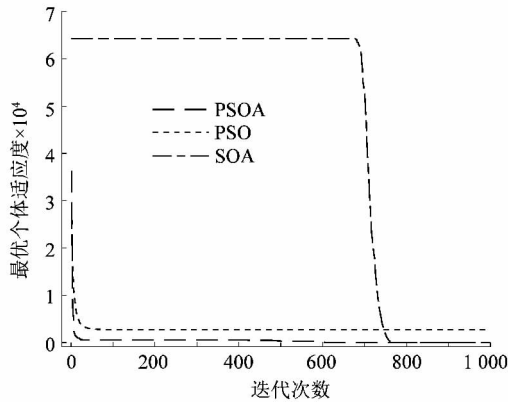


图 8 Sphere 优化结果曲线

Figure 8 Sphere optimization result curve

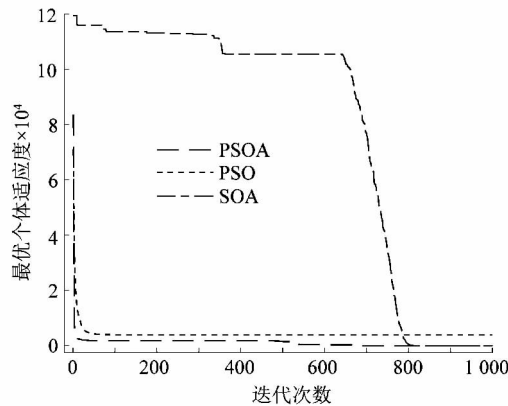


图 9 Schwefel 优化结果曲线

Figure 9 Schwefel optimization result curve

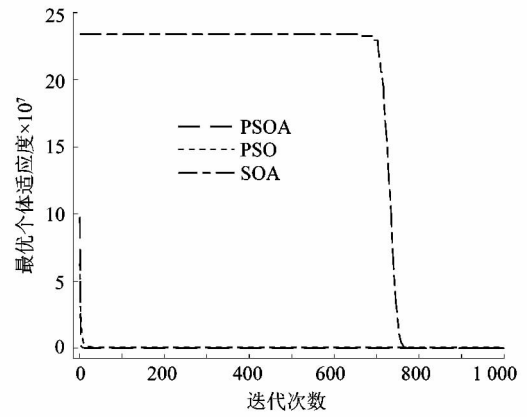


图 10 Rosenbrock 优化结果曲线

Figure 10 Rosenbrock optimization result curve

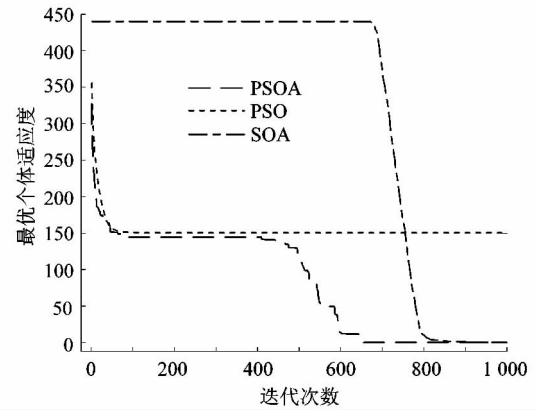


图 11 Rastrigin 优化结果曲线

Figure 11 Rastrigin optimization result curve

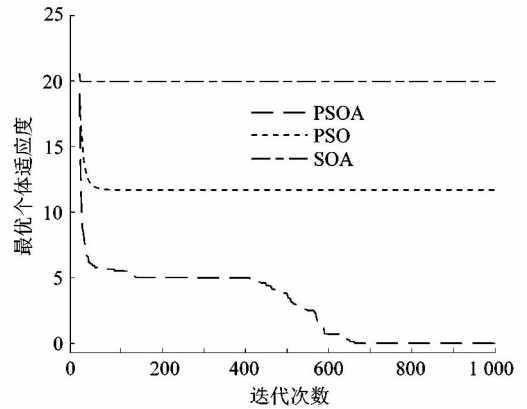


图 12 Ackley 优化结果曲线

Figure 12 Ackley optimization result curve

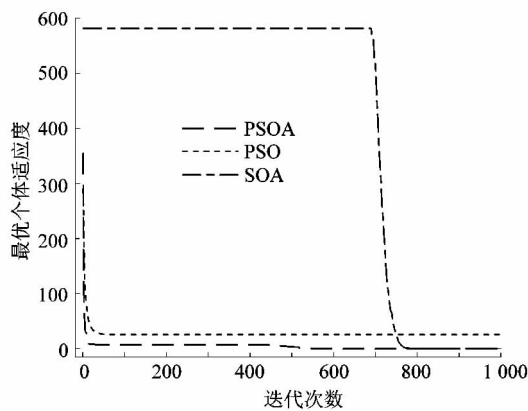


图 13 Griewank 优化结果曲线

Figure 13 Griewank optimization result curve

表 2 性能测试对比

Table 2 Performance test comparison

测试函数	最优值			收敛迭代数		
	PSOA	PSO	SOA	PSOA	PSO	SOA
Sphere	0.00	0.35	0.00	32	38	750
Schwefel	0.00	0.73	0.00	653	47	821
Rosenbrock	0.00	0.11	0.00	757	23	33
Rastrigin	0.00	157.50	0.03	701	86	822
Ackley	0.00	13.50	21.60	679	63	17
Griewank	0.00	39.20	0.00	587	46	765

从图 8 ~ 12 中可以看出 PSOA 的收敛速度均优于 PSO 和 SOA, 由表 2 可知 PSOA 的算法精度要优于 PSO 和 SOA; 同时, 从图中可看出 SOA 存在收敛速度慢、有时不易得到最优解等缺点, 这在 PSOA 的测试结果得到了明显的改善, 说明了改进算法的有效性。课题组从移动参数方面进行的改进取得了较好的结果。

4 结论

为解决库区货位分配的出入库速度慢、货架稳定性差、货品相关性低 3 个问题, 课题组建立货位分配数学模型, 采用海鸥优化算法对该模型进行求解, 结果表明:

1) 通过在算法更新策略的公式中引入非线性动态参数, 增强了算法的收敛速度、精确性以及早熟性。

2) 用 6 个基准测试函数对算法的寻优精确度以及避免陷入局部最优的能力进行测试, 改进后的算法

在寻优精度和寻优速度上都优于原始算法, 同时具备良好的避免陷入局部最优的能力。

后续的研究中应考虑将具体行业仓储物品的特殊性也加入模型中, 同时进一步优化算法性能。

参考文献:

- [1] 王军锋. 自动化立体仓库储位分配策略优化的研究[D]. 北京: 北京邮电大学, 2010:1.
- [2] WANG Y Y, MAN R J, ZHAO W M, et al. Storage assignment optimization for fishbone robotic mobile fulfillment systems[J]. Complex & Intelligent Systems, 2022, 8(6): 4587 - 4602.
- [3] LANZA G, PASSACANTANDO M, SCUTELLA M G. Assigning and sequencing storage locations under a two level storage policy: optimization model and matheuristic approaches[J]. Omega: the International Journal of Management Science, 2022, 108: 102565.
- [4] YUAN R P, LI J T, WANG W, et al. Storage assignment optimization in robotic mobile fulfillment systems[J]. Complexity, 2021, 2021: 1071 - 1081.
- [5] TU M R, YANG M F, KAO S L, et al. Using a heuristic multi-objective genetic algorithm to solve the storage assignment problem for CPS-based pick-and-pass system[J]. Enterprise Information Systems, 2021, 15(9): 1238 - 1259.
- [6] KEUNG K L, LEE C K M, JI P. Data-driven order correlation pattern and storage location assignment in robotic mobile fulfillment and process automation system[J]. Advanced Engineering Informatics, 2021, 50: 101369.
- [7] MIRZAEI M, ZAERPOUR N, DE KOSTER R. The impact of integrated cluster-based storage allocation on parts-to-picker warehouse performance[J]. Transportation Research Part E: Logistics and Transportation Review, 2021, 146: 102207.
- [8] TABATABAEI S K, VALILAI O F, ABEDIAN A, et al. A novel framework for storage assignment optimization inspired by finite element method[J]. PeerJ Computer Science, 2021, 7(1): e378.
- [9] HE Y Y, WANG A H, SU H L, et al. Particle swarm optimization using neighborhood-based mutation operator and intermediate disturbance strategy for outbound container storage location assignment problem[J]. Mathematical Problems in Engineering: Theory, Methods and Applications, 2019, 2019: 1 - 13.
- [10] DHIMAN G, KUMAR V. Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems[J]. Knowledge-Based Systems, 2019, 65: 169 - 196.