

新型无人仓 AutoStore 的货物合箱方法研究

杨玮, 张子涵, 张晓楠, 马伟
(陕西科技大学 机电工程学院, 西安 710021)

摘要: **目的** 为减少 AutoStore 系统在共同存储策略下的订单处理时间, 解决经营品类繁多, 但仓储面积有限的企业货物存储问题。**方法** 提出货物合箱的存储方法, 设计关联规则挖掘算法与混沌种子优化算法的混合优化算法进行求解。从机器人翻箱操作、料箱的分配规则、系统布局三方面进行分析, 结合 AutoStore 运作规律, 建立以机器人拣货行走距离最短为目标函数的数学模型, 通过 Apriori 算法输出货物热度, 输入混沌种子算法, 得到储位分配结果, 实现联动决策。在基本种子算法的迭代中加入混沌序列, 帮助种子避开局部极小值。**结果** 实例分析表明, 货物合箱方法能够显著减少共同存储下 AutoStore 系统的订单处理时间, 缩小与指定存储策略订单处理时间的差距, 与粒子群算法和基本种子优化算法相比, 混沌种子优化算法的优化效率更高、稳定性更好。**结论** 文中的研究为经营品类繁多, 但仓储面积有限的企业提供了一种更适合的货物存储策略, 具有较强的操作性和实际意义。

关键词: AutoStore 系统; 存储策略; Apriori 关联规则挖掘算法; 混沌种子优化算法

中图分类号: F274 **文献标识码:** A **文章编号:** 1001-3563(2022)17-0174-10

DOI: 10.19554/j.cnki.1001-3563.2022.17.022

Cargo Consolidation Method in New Unmanned Warehouse AutoStore

YANG Wei, ZHANG Zi-han, ZHANG Xiao-nan, MA Wei

(School of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology,
Xi'an 710021, China)

ABSTRACT: The work aims to reduce the order processing time of the AutoStore system under the common storage strategy and solve the storage problem encountered by companies with a wide range of goods but limited storage area. A storage method of cargo consolidation was proposed, and a hybrid optimization algorithm of association rule mining algorithm and chaotic seed optimization algorithm was designed for solution. Then, the analysis was carried out from the three aspects of robot turning operation, bin distribution rules, and system layout. Combined with AutoStore operation rules, a mathematical model with the shortest walking distance of robot picking as the objective function was established to output the heat of the goods through the Apriori algorithm, and input the chaotic seed algorithm to get the storage allocation result, thus realizing the linkage decision-making. The chaotic sequence was added in the iteration of the basic seed algorithm to help the seed avoid local minima. The case analysis showed that the cargo consolidation method could significantly reduce the order processing time of the AutoStore system under common storage, and narrow the gap with the specified storage strategy order processing time. Compared with the particle swarm optimization algorithm and the basic seed optimization algorithm, the chaotic seed optimization algorithm had higher optimization efficiency and better stability. This research provides a more suitable cargo storage strategy for companies with a wide range of business categories but limited storage area, which has strong operability and practical significance.

收稿日期: 2021-11-02

基金项目: 国家自然科学基金青年科学基金 (71802120); 西安市未央区项目 (201939)

作者简介: 杨玮 (1972—), 女, 博士, 教授, 主要研究方向为智能物流系统优化。

KEY WORDS: AutoStore system; storage strategy; Apriori association rule mining algorithm; chaotic seed optimization algorithm

机器人履行系统在世界各地的仓库中变得越来越普遍, AutoStore 系统作为新兴的移动机器人履行系统 (Robotic Mobile Fulfillment System, RMFS), 具有超高空间利用率、高流量、运行稳定的优势, 适用于目前蓬勃发展的电商企业、店仓一体化的大型零售商超, 以及有大量拆零拣选或小批量多频次拣选任务的企业, AutoStore 系统在国内外都具有广阔的使用空间^[1]。典型的使用案例包括英国零售商 Asda、美国全渠道电子商品零售商 Best buy、德国西门子公司等。国内首个 AutoStore 系统于 2020 年 6 月在北京投入使用。与传统自动化立体仓库不同, AutoStore 不必留有巷道空间, 而是由一列列形状规格相同的料箱叠加码放形成的高密度存储区, 人们把一列料箱称为一个堆栈。机器人在网格货架的顶部沿 x - y 方向移动, 配合提升机抓取和释放料箱, 拣选网格货架中的货物。若待拣选货物所在料箱不位于货架顶层, 而是位于堆栈的较深层次, 则机器人需要进行翻箱。仓储系统的存储策略决定了仓库所需的存储空间和机器人执行系统的效率, 由于 AutoStore 系统出现时间短, 目前关于它的研究仍处于起步阶段, 很少有文献可以指导决策者为操作这样的系统选择正确的存储策略, 因此对系统存储策略展开研究十分必要。

过往针对存储策略的研究主要围绕随机存储和分类存储 2 种网格分区的方法展开, 朱杰等^[2]研究了随机存储策略下 M/G/1 随机服务系统的服务效率问题, 提出 2 种货物拣选方式及其适用条件。Francisco 等^[3]针对具有随机存储和不同类型叉车仓库中的存储和检索问题, 将其分解为托盘集合位置、叉车的任务分配、作业顺序等 3 个子问题, 分别设计 3 个启发式算法进行求解。刘恺文等^[4]研究了自动化立体仓库在随机存储和定位存储 2 种策略下的出入库调度问题。Eder^[5]以基于穿梭车的存储和检索系统为研究对象, 通过应用基于类的存储策略提高系统最大吞吐量, 提高系统性能。闫军等^[6]使用多色集合理论得到仓库中货位的最优分区。分析发现, 以上研究对于最小存储单元, 如自动化立体仓库中托盘上货物的摆放规则, 或 AutoStore 中不同料箱内货物的摆放规则并没有研究。所做内容均为在某一存储策略下讨论系统结构设计或系统运行时间, 且未比较各存储策略之间的优劣, 综合权衡系统效率和存储空间。

为解决上述问题, Gharehgozli 等^[7]首次在集装箱自动化堆场系统 (Container Stack System, CSS) 中提出了针对最小存储单元存储规则的指定存储与共同存储策略的概念。随后, Zerpour 等^[8]在 3DAS/RS 系统中研究了指定存储与共同存储策略, 以最小化系统作业时间

为目标建立模型。Fumi 等^[9]提出了一种在指定存储策略下减少所需仓储空间的方法。Felix 等^[10]针对 Kiva 系统提出了一种最短路径的存储方法, 通过实验证明, 最短路径存储方法得出的结果优于基于规则的简单分配存储方法的结果。Singbal 等^[11]针对 AS/RS 系统, 设计了 3 种基于周转频率的存储策略, 将产品分配到对应通道, 得到结论: 3 种存储策略在不同评价标准下, 基于类的存储分配策略的性能优于基于周转频率的存储策略的性能。目前, 针对 AutoStore 系统, Beckschäfer 等^[12]提出了一种离散事件模拟方法, 讨论指定存储和每个料箱内存放 2 种商品的共同存储策略, 结果表明, 除非挖掘出产品一起订购的可能性, 否则分法不会显著提高效率, 选择合适的商品投入策略可以提高拣选效率达到 5%。Zou 等^[13]同样研究了指定存储与共同存储策略, 结果表明, 指定存储比共同存储的系统订单处理时间提高了约 24%, 但多出约 52% 的储位面积, 约 50% 的系统成本, 且在指定存储策略下, 系统的充填率较低, 空间浪费严重。随着企业产品数量和规模的增加, 企业不得不设置更多的存储空间, 花费更多的存储成本, 因此对于经营品类繁多或仓储空间有限的企业, 更愿意选择共同存储策略。

文中在过往 AutoStore 研究的基础上, 针对共同存储策略(每个料箱中存放的货物种类数大于 1, 每个堆栈中存放多种货物)订单吞吐时间长的问题, 提出 AutoStore 系统货物的合箱方法。以机器人拣货过程中行走总距离最短为目标函数, 建立基于货物关联规则的合箱模型, 设计料箱的分配规则以减少翻箱时间, 提出一种基于 Apriori 关联规则挖掘算法与混沌种子优化算法联动决策的混合优化算法。

1 问题描述和数学建模

1.1 问题描述及相关假设

AutoStore 系统中货物的存储方式分为指定存储和共同存储, 货物合箱是在应用共同存储策略的情况下, 设计货物存放规则和料箱分配规则, 使双指令任务下机器人拣选行走时间、翻箱时间最小化, 作业效率最大化。在 AutoStore 系统中, 订单处理的工作流程见图 1, 具体过程如下。

1) 系统接收客户订单, 为其分配处在空闲状态的机器人, 机器人接收订单指令后从待位点行走至目标料箱处, 若料箱不位于货架顶层, 则机器人进行翻箱操作。

2) 机器人携带目标料箱送至距离最近的工作站, 由拣选人员从中选取需要的商品。

3) 工作站操作完成交换料箱, 机器人携带存储料箱到指定存储位置, 并在该处停留。

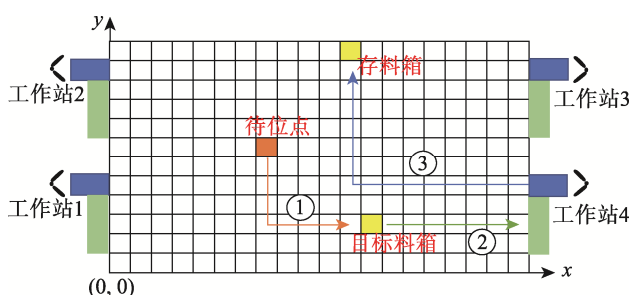


图1 AutoStore 系统顶层作业流程
Fig.1 Top-level operation flowchart of AutoStore system

根据以上订单处理工作流程, 可以得到系统中机器人搬运货物、完成订单的距离由 3 部分构成: 机器人完成上一个订单停留位置与最新指令订单所在料箱位置之间的距离 d_{dr} 、新指令订单所在料箱与工作站之间的最小距离 $d_{r,pi}$ 、以及为提取出目标料箱所需的翻箱距离 d_r 。指定存储是“一位一货”, 即每个料箱中只存放一种货物, 料箱堆码形成堆栈, 每个堆栈中也只存放一种货物。共同存储是指, 每个料箱中存放的货物种类数大于 1, 每个堆栈中存放多种货物, 因此, 在处理相同订单时二者产生的 d_{dr} 和 $d_{r,pi}$ 不同。在指定存储策略中, 机器人每次只需从堆栈顶端抓取目标料箱, 因此其 d_r 为 0, 而共同存储策略中, 若目标料箱位于堆栈的较深层次, 则机器人需要进行翻箱操作。

为不失一般性, 在建立数学模型时进行如下假设。

1) 机器人翻箱操作时, 四周堆栈未达到最大堆叠高度。

2) 机器人采用专用机器人服务规则, 一个工作站分配一个固定的机器人。

3) 每个料箱有固定尺寸标准, 料箱中容纳的所有货物的质量和体积不能超过限定值。

4) 忽略机器人行走过程中的转向时间对优化结果的影响。

1.2 数学建模

在建立模型之前需明确翻箱操作, 并将系统布局进行数学化表达。

1.2.1 机器人翻箱操作

如图 2 所示, 假设目标料箱位于最多能存放 H 个料箱堆栈的第 r 层, 该堆栈中目前存放了 i 个料箱, 则机器人需要将 $r+1$ 层到 i 层 ($r=1, 2, \dots, I; i=1, 2, \dots, H$) 的阻挡料箱移除后才能取出目标料箱。 $d_{r,el}$ 为机器人将阻挡料箱依次移到相邻堆栈顶部的距离; d_{lu}^s

为机器人将目标料箱抓起放在相邻堆栈顶部, 完成翻箱后再抓起目标料箱的距离; $d_{r,e2}$ 为机器人将阻挡料箱放回原堆栈的距离, 3 种距离可分别用式 (1) — (3) 计算。

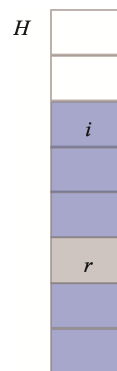


图2 机器人翻箱示意图
Fig.2 Schematic diagram of robot turning cargo

$$d_{r,el} = \sum_{j=0}^{i-r} \{2[H - (r+1) + j]h + 2jb\} \quad (1)$$

$$d_{lu}^s = 2(H-r)h \quad (2)$$

$$d_{r,e2} = \sum_{j=0}^{i-r} [2(H-r+j)h + 2jb] \quad (3)$$

式中: j 为阻挡料箱的数量; h 为单元料箱的高度; b 为单元料箱的宽度。

1.2.2 系统布局

AutoStore 标准布局下的网格货架为 14 行 20 列 16 层, 工作站有 4 个, 均匀分布在系统两侧。设工作站的坐标为 (x_{pi} , y_{pi} , h_{pi}), 计算见式 (4)。

$$x_{pi} = \begin{cases} 0 & i=1, 2, \dots, \frac{P}{2} \\ b_1 & i=\frac{P}{2}+1, \frac{P}{2}+2, \dots, P \end{cases}$$

$$y_{pi} = \begin{cases} \frac{L}{\frac{P}{2}+1} \cdot i & i=1, 2, \dots, \frac{P}{2} \\ \frac{L}{\frac{P}{2}+1} \cdot (i - \frac{P}{2}) & i=\frac{P}{2}+1, \frac{P}{2}+2, \dots, P \end{cases} \quad (4)$$

$$h_{pi} = 3h$$

式中: P 为工作站的数量; b_1 、 L 分别为系统的宽度和长度; h 为单元料箱的高度。

1.2.3 数学模型

在分析了上述相关问题后建立合箱模型, 包括商品存放策略和料箱分配规则 2 个子模型。模型中使用的决策变量定义如下, 其他参数变量定义见表 1。

表 1 模型中使用的参数变量
Tab.1 Parametric variables used in the model

变量	含义	变量	含义
$C_{kk'}$	基于关联规则挖掘算法计算出的历史订单中货物 k 和 k' 之间的关联度	$d_{kk'}$	已分配商品 k 和待分配商品 k' 即将被分配位置之间的距离
p_k	基于关联规则挖掘算法计算出的商品 k 的支持度 (代表商品需求频率)	D_i	料箱所处位置 i 距离网格货架最顶层的距离
r_i	料箱所在位置 i 距离工作站的水平距离	K	待分配商品种类数
k'	待分配商品	P	料箱 i 已分配到的商品种类数
V_i	料箱 i 的容积	n_{ki}	已分配到料箱 i 的商品 k 的数量
n_{ki}	待分配到料箱 i 的商品数量	σ	料箱的实际利用率
V_k 、 $V_{k'}$	商品 k 和 k' 的体积	m_k 、 $m_{k'}$	商品 k 和 k' 的质量
Q	待分配的所有商品总数	m_i	料箱 i 的最大载质量
(x, y, z)	已分配商品 k 的位置	(x', y', z')	待分配商品 k' 的位置

决策变量 x_{ki} ($0 \sim 1$): 商品 k 是否分配到料箱 i 中。

$$\min \sum_{i=1}^R \sum_{k=1}^K (\sum_{k'=1}^P C_{kk'} \cdot d_{kk'} \cdot x_{ki} + p_k \cdot D_i \cdot x_{ki} + p_k \cdot r_i \cdot x_{ki}) \quad (5)$$

条件:

$$\sum_{i=1}^R x_{ki} = 1 \quad k = 1, 2, \dots, K \quad (6)$$

$$\sum_{k=1}^K x_{ki} \cdot n_{ki} \cdot V_k + \sum_{k'=1}^P n_{ki} \cdot V_{k'} \leq V_i \cdot \sigma \quad i = 1, 2, \dots, R \quad (7)$$

$$\sum_{k=1}^K x_{ki} \cdot n_{ki} \cdot m_k + \sum_{k'=1}^P n_{ki} \cdot m_{k'} \leq L_i \quad (8)$$

$$\frac{n_{ki}}{Q} \approx p_k \quad (9)$$

$$d_{kk'}(x, y, z, x', y', z') = |x - x'| \cdot b + |y - y'| \cdot l + |z - z'| \quad (10)$$

文中的合箱模型是以使关联货物被分配到的料箱间的距离、顾客复购率最高货物到货架顶层距离, 以及顾客复购率最高货物到工作站的水平距离之和的最小化为优化目标。约束条件 (6) 表示确保在分配过程中, 每件货物只能被分配到一个货位, 其中 K 指待分配货物种类数; 约束条件 (7) 表示由于系统中的每个料箱容量都有限制, 且货物的存放不是紧密存储, 因此设定料箱实际利用率为 σ , 已经分配货物的体积加上将要分配的货物体积, 不能超过料箱容积与其实际利用率的乘积; 约束条件 (8) 表示由于系统中的每个料箱都有最大载质量限制, 因此待分配货物质量加上已分配货物质量之和不能超过料箱最大载质量; 约束条件 (9) 表示待分配货物数量与其货物总数之间的比例应近似等于货物的需求频率 (即货物支持度), 这是因为若存储某种货物数量过高而需求频率较低, 会造成浪费; 反之, 又会造成频繁补货。

约束条件 (10) 表示已分配商品 k 和待分配商品 k' 之间的距离用曼哈顿距离计算。

为满足模型中将需求频率大的货物优先放在距网格货架顶层最近, 且距工作站最近的地方的要求, 文中的第 2 个工作是设计料箱的分配规则, 将需求频率从高到低货物的放置顺序与料箱的分配顺序一一对应。具体分配规则: 不同层之间由上至下, 先分配距离网格顶部最近的层, 同层之间按照距离工作站的远近对称分配。图 3 中绘制了 AutoStore 标准布局的四分之一, 根据式 (4) 确定工作站的位置。数字 1—35 即为同层间料箱的分配顺序。图 3 中相同颜色的格子代表到工作站单位距离相同的料箱, 相同颜色内料箱的分配顺序又与料箱的单位宽度与长度有关。如分配顺序为 2、3、4 的 3 个料箱到工作站均为 2 个单位距离, 其中 2 号料箱为沿宽度方向 2 个单位长度, 3 和 4 料箱为沿长度方向、沿宽度方向 1 个单位长度, 而 AutoStore 系统中料箱的长度为 649 mm、宽度为 449 mm, 因此优先分配 2 号料箱, 其余层按照同样方法。这样的分配方式保证了高需求货物 (高周转率料箱) 集中在货架顶部且靠近工作站处, 减少机器人翻箱的工作量; 同时, 以工作站所在行对称分布的方式, 减少了翻箱过程中可能造成的拥堵和干涉, 分散仓储区域的局部作业热度, 减少了机器人的等待时间。

将上述料箱分配规则转化为料箱的分配优先度, 设任意一个料箱的坐标为 (x_b, y_b, z_b) , 则同层内料箱的分配优先度可由式 (11) 计算。 x_b 、 y_b 的绝对值越大, 表示料箱距离工作站的单位距离越大, 分配优先度越小; 当单位距离相同时, x_b 方向数值越大, 到工作站的距离越长, 分配优先度越小。按照料箱的分配优先度从大至小进行货位分配。

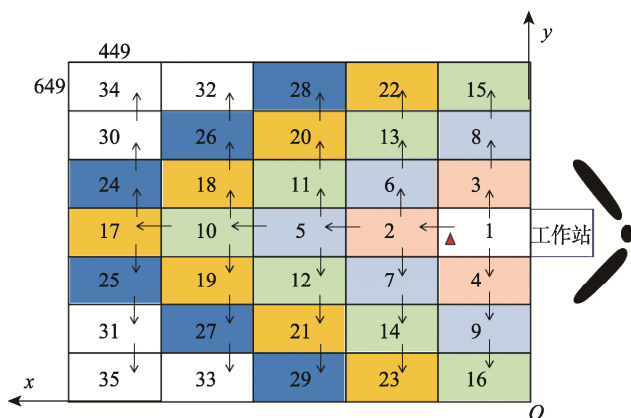


图3 最顶层料箱分配顺序
Fig.3 Distribution order of top bin

$$\alpha = \frac{1}{(|x_b| + |y_b|) + \frac{|x_b|}{\max(x_b)}} \quad (11)$$

2 算法设计

文中算法是基于 Apriori 关联规则挖掘算法与混沌种子优化算法联动决策的混合优化算法 (Apriori Chaotic Bean Optimization Algorithm, ACBOA) 的求解合箱模型。文中算法的设计分为 2 个阶段, 首先利用 Apriori 关联规则挖掘算法挖掘历史订单数据, 实现对货物间关联规则的挖掘, 其次将 Apriori 算法的结果输入混沌种子优化算法中; 求解货物的合箱结果。

2.1 混沌种子优化算法

种子算法是张晓明等^[14]受自然界种子传播方式的启发, 提出的一种进化算法, 他通过仿真实验证明了该算法较粒子群算法 (PSO) 具有较好寻优性能, 并通过数学计算和测试函数证明了其能够以概率 1 收敛于全局最优解。基本种子优化算法的初始化过程随机, 个体质量无法得到保证, 容易陷入局部最优。混沌理论近年来被广泛研究, 认为其可以加快算法收敛速度、提高求解精度, 因此, 文中针对基本种子算法存在的容易陷入局部最优、寻优精度低等问题, 设计混沌种子优化算法, 在基本种子算法的迭代中加入混沌序列, 产生多个区域的局部最优解点, 帮助种子避开局部极小值, 快速找到最优解。

2.2 种子个体编码

对于货物分配问题, 其解的结构包括起点 $|K|$ (可理解为待分配的货物) 和终点 $|L|$ (可理解为料箱), 这样可以得到一个 $|K|$ 行 $|L|$ 列的分配矩阵。为了能够确定唯一的分配矩阵, 需要确定优先满足哪一个终点, 以及确定优先由哪一个起点满足该终

点。由于文中已经设计了料箱的分配顺序, 因此将文中终点的优先级作为已知, 采用直接编码的方式, 将种子个体位置的维度设置为待分配货物的数量 N , 每一维位置向量都代表一个货物, 且各向量的分量不重复。设种子个体的向量 $X = (x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_n)$, 则 $x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n$ 表示分配到同一料箱中的商品为 $1, 2, \dots, n$, 其中 $x_i, x_j = 1, 2, \dots, n$ 且 $x_i \neq x_j (i \neq j)$ 。

2.3 适应度函数及解码

根据系统官网给出的机器人水平方向运行速度 v_r 和轨道提升机速度 v_h , 将优化目标转化为所有订单处理总时间最短, 则适应度函数见式 (12)。其中 $t_{\max}$ 为订单处理时间最大值, 以其作为适应度参数值, 既满足适应度函数最大序列的作业时间最小的要求, 又能淘汰所有拣选作业中完成时间最长的序列。

$$f(t) = t_{\max} - g(t) \quad (12)$$

根据合箱模型的约束条件, 解码执行步骤如下。

1) 参数初始化。设置最大迭代次数、种群大小和父种间的距离阈值。

2) 混沌初始化种子的位置, 随机生成一个介于 $[0, 1]$ 的 n 维向量值, $x = (x_1, x_2, \dots, x_n)$, n 为目标函数的变量数量。然后得到 n 个混沌向量 $Z_1, Z_2, \dots, Z_n$, 根据逻辑映射方程, 将 Z_i 的每个分量映射到要优化的变量值的空间。子代个体的进化方程见式 (13) 和式 (14), 其中 $X[i]$ 指种子 i 的位置, X_{fi} 是种子 i 父代的位置, $Y(i)$ 为父代的传播向量函数, 种群规模用 N 表示, 每个种群中的个体都可用向量 X 表示。 α 、 β 、 γ 表示常数向量, a 、 b 指每一个父代周围分布子代的比例。在每次迭代中, 根据模型的目标函数, 选择剩余种子中适应度最大的种子作为父代, 整个种子群体按照进化方程循环进化。

$$X[i] = \begin{cases} X[i] & X[i] \text{ 为父代} \\ X_{fi} + \text{rand}() \cdot Y(i) & X[i] \text{ 不为父代} \end{cases} \quad (13)$$

$$Y(i) = \begin{cases} \alpha & \text{若 } i \leq (N \cdot a) \\ \beta & \text{若 } (N \cdot a) \leq i \leq (N \cdot b) \\ \dots & \dots \\ \gamma & \text{其他} \end{cases} \quad (14)$$

3) 计算每个种子的适应度值, 找到适应值最优的种子, 将其选为候选父代。

4) 选择一个适应度值最优的种子作为 1 号父代种子, 从剩余的种子中选择具有最优适应值的种子, 并判断它与所选择的父代之间的近似程度 (文中使用 Apriori 算法产生的结果)。

5) 根据种群分布 (文中为料箱的位置、质量和

体积约束),在每个父代的扩散范围内生成相应的后代种子组。

6) 混沌搜索。如果迭代次数超过阈值,则根据式(15),由当前父代种子对子代种子添加混沌干扰,之后根据式(16)映射到优化变量,返回步骤3,其中 σ 是一个可调参数, $0 \leq \sigma \leq 1$; Z 为映射生成的混沌序列向量, $Z = \{Z_1, Z_2, \dots, Z_d\}$; φ 为当前最优解向量映射到 $[0, 1]$ 内形成的最优混沌向量。

$$Z = (1 - \sigma)\varphi + \beta Z \quad (15)$$

$$\varphi = (X - a) / (b - a) \quad (16)$$

7) 判断是否满足终止条件,如果满足,结束算法;否则,执行步骤3。

8) 输出结果,结束算法。

图4为ACBOA的具体算法流程,其中前8步为Apriori算法的步骤,第9到18步为混沌种子优化算法(CBOA)的步骤。

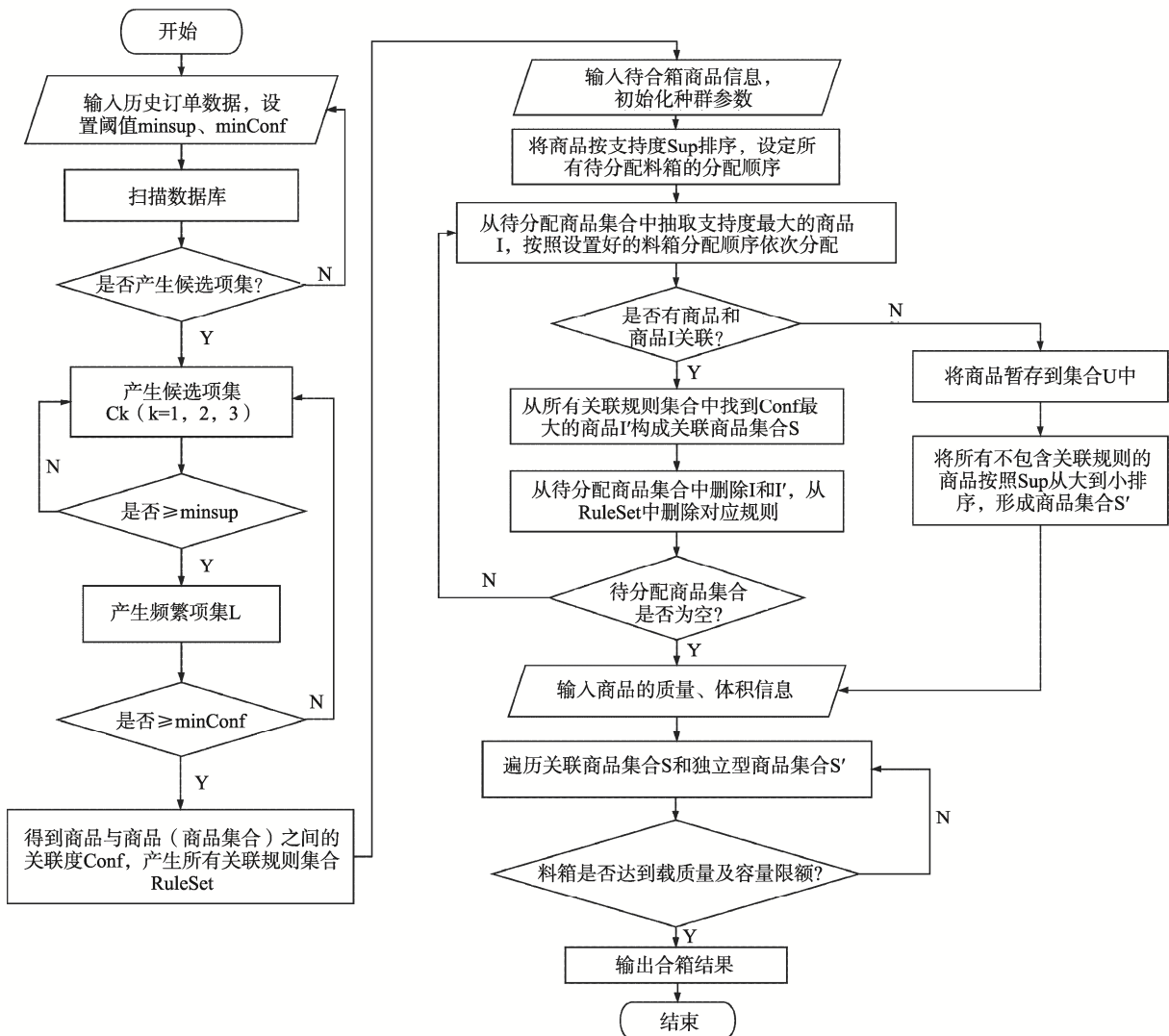


图4 ACBOA 算法流程
Fig.4 ACBOA algorithm flowchart

3 仿真实验

3.1 算例数据及算法参数设置

英国某在线零售商超使用 AutoStore 系统进行货物存储,根据其 2018 年的销售订单数据对货物合箱后系统订单吞吐时间进行分析,设计具有不同参数场景的问题进行数值实验,具体系统运行参数(来源于 AutoStore 系统介绍官网)和算法参数设置如下:网格货架行数为 7 行,网格货架列数为 5 列,网格货架层数为 8 层,料箱尺寸为 2 970 mm³,料箱最大载质量为 35 kg,料箱实际利用率 σ 为 85%,最小支持度为 0.005,最小置信度为 0.5,料箱的最大容量为 59 040 cm³,系统订单到达率 λ 为 0.1 个/s,机器人转向时间 t_d 为 2 s,工作站处订单到达率 λ_p 为 0.1 个/s,机器人每次抓取/释放料箱的时间 t_u 为 2 s,机器人垂直方向轨道提升速度 v_h 为 1.6 m/s,机器人水平方向行走速度 v_l 为 3.1 m/s。

3.2 仿真结果分析

以系统处理 500、1 000、3 000 张 3 种不同规模的订单数量为例进行分析。其中 3 000 张订单的处理结果：3 000 张订单共包含 28 722 条商品购买数据，15 000 种不同种类的商品，品类涵盖食品、药品、个护和运动等 4 个方面，共挖掘出货物关联规则 126 条，见表 2。求得货物的合箱结果见表 3。

为验证算法有效性，将混沌种子优化算法（CBOA）与基本种子优化算法（BOA）、粒子群算

法（PSO）（此处默认已使用 Apriori 算法挖掘出货物的关联规则）在不同作业规模下运行 50 次的结果进行对比，见表 4。适应度、收敛代数、收敛时间的箱线图比较见图 5。设置连续迭代 500 次最优值不变为收敛的判断条件，根据文献[15]，粒子群算法求解离散问题的参数设置：群体规模为 20，惯性权重因子为 1.0，学习因子为 2，微粒最小位置值为 0，最大位置值为 4，微粒最小速度值为-4 m/s，最大速度值为 4 m/s，进化次数为 500，种群规模为 100。

表 2 货物关联规则（部分）
Tab.2 Goods association rules (partial)

支持度最大的 货物集合	置信度最大的 货物集合	置信度	支持度最大的 货物集合	置信度最大的 货物集合	置信度
非洲辣椒酱	一袋有机香蕉	1.0	野餐垫	梨汁	1.0
有机甜洋葱	厨房四件套	1.0	姜饼人模具	一次性纸巾	1.0
蓝莓酸奶	桃子酸奶	1.0	水果罐头、蓝莓干	香蕉辅食、有机婴儿菠菜	1.0
覆盆子果脯	香蕉辅食	0.8	低脂牛奶 100%	有机婴儿菠菜	0.8

表 3 合箱结果（部分）
Tab.3 Consolidation result (partial)

料箱号	料箱坐标	箱内货物	数量	质量/kg	体积/cm ³
1	(1,4,1)	香蕉辅食	3	26.00	56 931.00
		有机苹果酱	1		
		有机婴儿菠菜	3		
		蓝莓干	1		
		有机婴儿菠菜	5		
2	(2,4,1)	香蕉辅食	3	31.45	57 115.00
		薄脆	1		
		水果罐头	1		

表 4 算法 benchmark 性能比较
Tab.4 Algorithm benchmark performance comparison

订单数量	算法	均值/m	平均偏差	求解时间/s	优化效率/%
500	PSO	452.972	2.852	3 566.20	17.31
	BOA	424.692	1.374	1 680.64	17.34
	CBOA	418.032	1.144	2 180.34	22.32
1 000	PSO	802.763	6.867	6 984.63	17.29
	BOA	794.873	3.387	2 580.91	17.36
	CBOA	779.063	1.837	3 283.03	22.34
3 000	PSO	1 869.905	9.160	12 185.54	18.22
	BOA	1 854.655	2.244	5 784.67	17.74
	CBOA	1 833.275	2.221	6 574.85	22.41

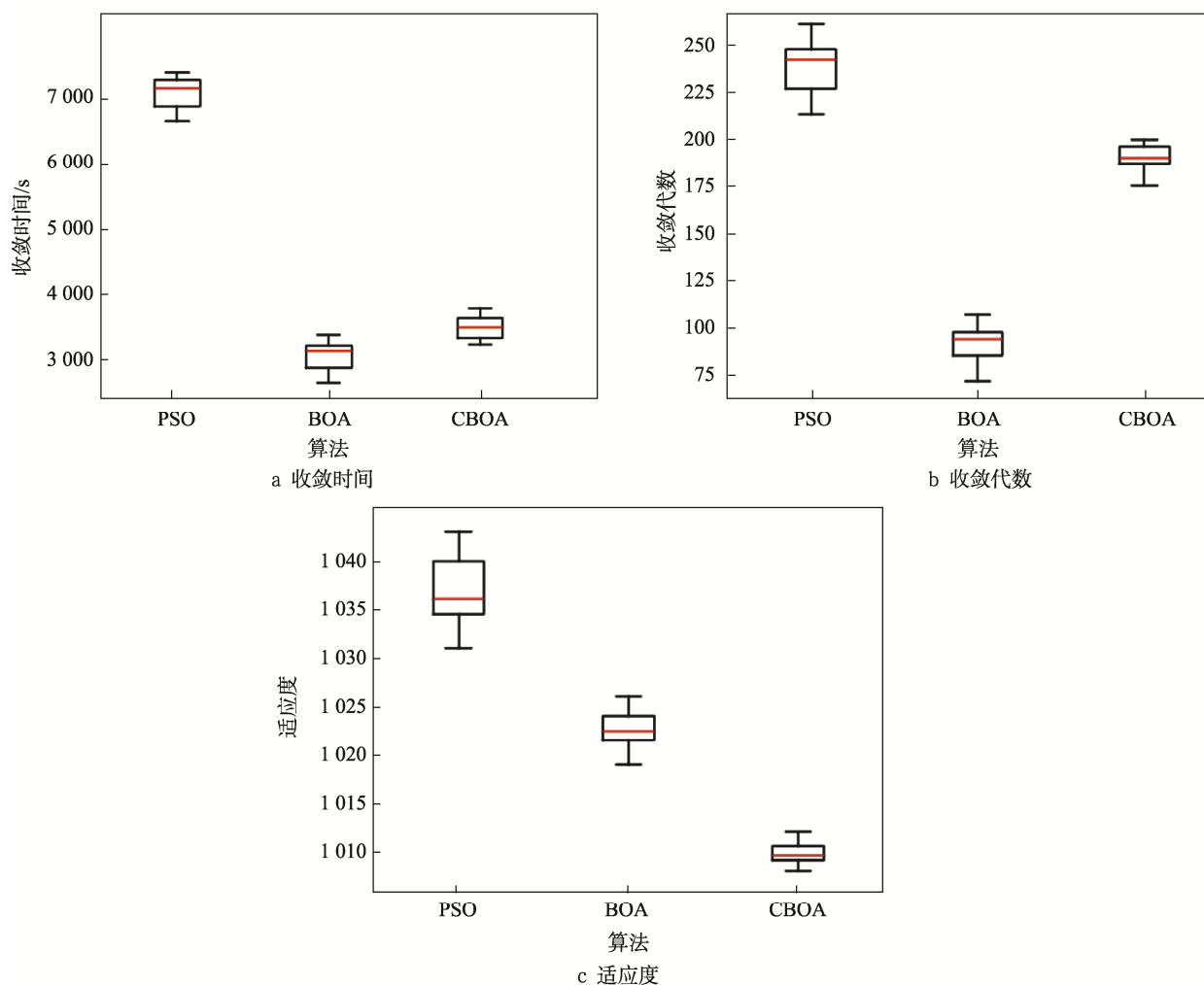


图 5 算法优化性能箱线图比较

Fig.5 Algorithm optimization comparison based on cargo plot

由图 5 可知, CBOA 的数据波动明显小于 PSO 和 BOA, 说明 CBOA 稳定性强。CBOA 的收敛时间和速度明显优于 PSO, 与 BOA 结果接近, 这是因为基本种子算法更易陷入局部最优, 从而更快收敛。算法求得的适应度值明显优于其他 2 种算法。

粒子群算法、基本种子优化算法和混沌种子优化算法这 3 种算法对合箱问题在 3 种作业规模下的算法收敛效果见图 6。

分析图 6 中的收敛曲线和数据可以看出, 粒子群算法收敛速度较慢, 所求结果较基本种子优化算法和混沌种子优化算法相比较差; 基本种子优化算法收敛速度很快, 但是由于初始化过程随机, 容易陷入局部最优, 无法使种群集中在最优值附近; 而文中提出的混沌种子优化算法, 在种子迭代中加入混沌序列, 产生多个区域的局部最优解点, 帮助种子避开局部极小值, 不仅避免算法过早收敛陷入局部最优, 而且算法求解精度高。

将按照网格分区原则划分的随机存储和分区存储与按照货物最小存储单元划分的制定存储和

共同存储两两组合, 得到 4 种常见存储策略: 共同分区存储、共同随机存储、指定分区存储和指定随机存储。分区存储是基于商品的订购频率进行分类分区, 周转率前 15% 的商品归为 A 类, 中间 35% 的商品归为 B 类, 剩下 50% 的商品归为 C 类, A 类商品存放在最靠近工作站处, B 类次之, C 类最远。将合箱存储的仿真结果与这 4 种存储方式进行对比, 共设置订单平均处理时间变化率、机器人行走总时间变化率、机器人翻箱总时间变化率、平均每个订单需要搬运的料箱数变化率等 4 个实验评价指标, 合箱存储相较于另外 4 种存储方式的变化百分比见表 5。

由于指定存储策略下机器人没有翻箱时间, 因此共同存储的机器人行走总时间和翻箱总时间不与指定存储比较。分析发现, 相较于共同存储策略下的分区和随机存储, 合箱存储在 4 个实验评价指标的数值均有所下降, 因此说明合箱存储策略能够有效降低共同存储策略下系统订单吞吐时间。合箱存储比指定随机存储的订单处理时间短 8.5%, 比指定分区存储的时

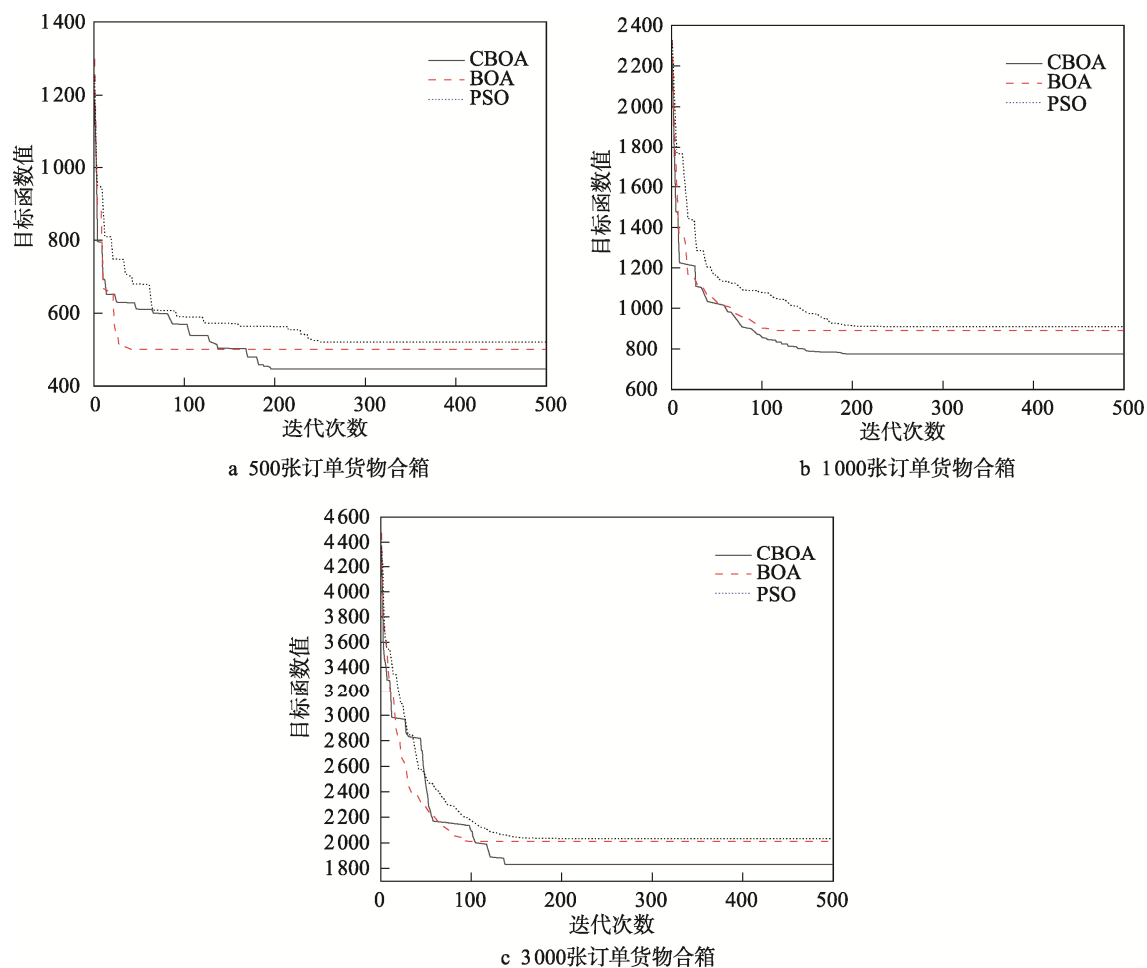


图6 不同作业规模下算法收敛效果

Fig.6 Convergence effect of algorithm under different operation scale

表5 合箱存储相较于另外4种存储方式改进值

Tab.5 Improved value of consolidated storage compared to the other four storage methods

存储方式	订单平均处理时间 变化率/%	机器人行走总时间 变化率/%	机器人翻箱总时间 变化率/%	平均每个订单需要搬运的料箱数 变化率/%
共同分区存储	7.55↓	54.63↓	3.76↓	9.17↓
共同随机存储	39.27↓	74.59↓	5.10↓	50.58↓
指定分区存储	19.92↑	—	—	53.14↓
指定随机存储	8.5↓	—	—	53.14↓

间长 19.92%。文献[12]中提到,共同存储与指定存储下系统订单处理时间相差约 24%,而在文中所得出的结果中,共同存储策略下的合箱存储与指定分区存储的时间差缩小到 19.92%。综上,合箱问题的提出能够缩小共同存储与指定存储订单处理时间的差距,进一步缩短共同存储策略的订单处理时间,为经营品类繁多或仓储空间有限的企业提供选择。

4 结语

共同存储策略与指定存储策略相比,具有系统成本低、吞吐时间长等特点,考虑如何缩短系统的吞吐时间,平衡系统订单处理能力与运行成本,为经营品类繁多但仓储面积有限的企业提供更适合的存储策

略。通过分析共同存储策略存在机器人翻箱动作、箱内货物摆放杂乱导致系统订单处理时间长的问題,结合 AutoStore 系统的工作原理,在订单处理过程中,建立以机器人行走距离最短为目标函数的合箱模型,设计了不同层之间由上至下,同层之间按照距离工作站远近进行先后分配的料箱分配规则,采用基于 Apriori 算法与混沌种子优化算法的复合优化算法进行求解。仿真结果表明,文中提出的混沌种子优化算法比粒子群算法和基本种子优化算法的运算速度更快、优化效率更高。文中提出的货物合箱方法为紧致化仓储系统提供了一种新的货物存储方式,广泛适用于电商、医药等经营品类繁多,但仓储面积有限的企业,未来会进一步探讨不同机器人服务规则和订单分批策略对货物合箱的影响。

参考文献:

- [1] Swisslog. 瑞仕格 AutoStore 落户上海为智慧用水打造物流之枢[J]. 现代制造, 2021(11): 40-41.
Swisslog. Swisslog AutoStore Settled in Shanghai to Create a Hub for Smart Water Logistics[J]. Maschinen Markt, 2021(11): 40-41.
- [2] 朱杰, 周丽, 郭键. 分类存储人工拣选随机服务系统效率研究[J]. 管理科学学报, 2012, 15(2): 59-71.
ZHU Jie, ZHOU Li, GUO Jian. Research on Random Service System Efficiency of Sorted Storage Manual Order Picking Manners[J]. Journal of Management Sciences in China, 2012, 15(2): 59-71.
- [3] BALLESTÍN F, PÉREZ Á, QUINTANILLA S. A Multi-stage Heuristic for Storage and Retrieval Problems in a Warehouse with Random Storage[J]. International Transactions in Operational Research, 2020, 27(3): 1699-1728.
- [4] 刘恺文, 曹政才. 基于改进灰狼优化算法的自动化立体仓库作业能量优化调度[J]. 计算机集成制造系统, 2020, 26(2): 376-383.
LIU Kai-wen, CAO Zheng-cai. Energy-Optimized Task Scheduling of Automated Warehouse Based on Improved Grey Wolf Optimizer[J]. Computer Integrated Manufacturing Systems, 2020, 26(2): 376-383.
- [5] EDER M. Analytical Model to Estimate the Performance of Shuttle-Based Storage and Retrieval Systems with Class-Based Storage Policy[J]. The International Journal of Advanced Manufacturing Technology, 2020, 107: 1-16.
- [6] 闫军, 王璐璐, 常乐. 多色集合在仓配一体化仓库货位分配建模中的应用[J]. 包装工程, 2019, 40(23): 203-208.
- [7] YAN Jun, WANG Lu-lu, CHANG Le. Application of Polychromatic Set in Cargo Space Allocation Modeling of Warehouse Integrated with Storage and Distribution[J]. Packaging Engineering, 2019, 40(23): 203-208.
- [8] GHAREHGOZLI A H, YU Yu-gang, KOSTER R, et al. A Decision-Tree Stacking Heuristic Minimising the Expected Number of Reshuffles at a Container Terminal[J]. International Journal of Production Research, 2014, 52(9): 2592-2611.
- [9] ZAERPOUR N, YU Yu-gang, KOSTER R B. Storing Fresh Produce for Fast Retrieval in an Automated Compact Cross-Dock System[J]. Production and Operations Management, 2015, 24(8): 1266-1284.
- [10] FUMI A, SCARABOTTI L, SCHIRALDI M M. Minimizing Warehouse Space with a Dedicated Storage Policy[J]. International Journal of Engineering Business Management, 2013, 5: 1-10.
- [11] WEIDINGER F, BOYSEN N, BRISKORN D. Storage Assignment with Rack-Moving Mobile Robots in KIVA Warehouses[J]. Transportation Science, 2018, 52(6): 1-10.
- [12] SINGBAL V, ADIL G K. Designing an Automated Storage/Retrieval System with a Single Aisle-Mobile Crane under Three New Turnover Based Storage Policies[J]. International Journal of Computer Integrated Manufacturing, 2021, 34(2): 212-226.
- [13] BECKSCHÄFER M, MALBERG S, TIERNEY K. Simulating Storage Policies for an Automated Grid-Based Warehouse System[C]// International Conference on Computational Logistics. Springer, Cham, 2017.
- [14] ZOU B, KOSTER M, XU X. Evaluating Dedicated and Shared Storage Policies in Robot-Based Compact Storage and Retrieval Systems[J]. ERIM Report Series Reference, 2016: 132-146.
- [15] 张晓明, 王儒敬, 宋良图. 一种新的进化算法——种子优化算法[J]. 模式识别与人工智能, 2008, 21(05): 677-681.
ZHANG Xiao-ming, WANG Ru-jing, SONG Liang-tu. A New Evolutionary Algorithm-Seed Optimization Algorithm [J]. Pattern Recognition and Artificial Intelligence, 2008, 21(05): 677-681.
- [16] 王凌, 刘波. 微粒群优化与调度算法[M]. 北京: 清华大学出版社, 2008: 1-219.
WANG Ling, LIU Bo. Particle swarm optimization and scheduling algorithms[M]. Beijing: Tsinghua University Press, 2008: 1-129.

责任编辑: 曾钰婵