

六向穿梭机仓储系统多目标货位分配优化算法研究

陈宏¹, 杨晨², 包建东²

¹苏州弘远机械制造股份有限公司, 江苏 苏州

²南京理工大学自动化学院, 江苏 南京

收稿日期: 2025年9月20日; 录用日期: 2025年10月1日; 发布日期: 2025年10月22日

摘要

本文针对六向穿梭机仓储系统多目标货位分配优化算法进行研究。首先选择基于货物出入库效率原则、货架稳定性原则以及穿梭机作业均衡原则的随机存储策略作为仓库的存储方式, 并建立六向穿梭机仓储入库货物货位分配数学模型; 随后改进第三代快速非支配排序遗传算法(NSGA-III)进行模型求解, 通过引入反向学习机制和随机扰动提高初始种群多样性, 引入自适应差分变异平衡全局搜索和局部收敛能力。最后, 结合实例进行仿真实验。通过与各个目标的单目标优化结果比较, 验证构建模型的有效性和算法的实用性。

关键词

六向穿梭机, 智能仓储, 货位分配, 遗传算法

Multi-Objective Storage Location Assignment Optimization Algorithm for Six-Way Shuttle Automated Storage and Retrieval Systems

Hong Chen¹, Chen Yang², Jiandong Bao²

¹Suzhou Hongyuan Machinery Manufacturing Co., Ltd., Suzhou Jiangsu

²School of Automation, Nanjing University of Science and Technology, Nanjing Jiangsu

Received: September 20, 2025; accepted: October 1, 2025; published: October 22, 2025

Abstract

This paper investigates the multi-objective storage location assignment optimization problem for

文章引用: 陈宏, 杨晨, 包建东. 六向穿梭机仓储系统多目标货位分配优化算法研究[J]. 动力系统与控制, 2025, 14(4): 446-463. DOI: 10.12677/dsc.2025.144045

six-way shuttle automated storage and retrieval systems. First, a random storage strategy is adopted as the storage method based on the principles of storage/retrieval efficiency, rack stability, and shuttle operation balance. Accordingly, a mathematical model of storage location assignment is established. We then enhanced the third-generation nondominated sorting genetic algorithm (NSGA-III) to solve the model. By embedding an opposition-based learning mechanism and random perturbation we boosted the diversity of the initial population, while an adaptive differential-mutation operator was introduced to balance global exploration and local convergence. Finally, a real-world instance was used for simulation experiments. Comparisons with single-objective optima of each target confirm the validity of the proposed model and the practicality of the algorithm.

Keywords

Six-Way Shuttle, Intelligent Warehouse, Storage Location Assignment, Genetic Algorithm

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

制造业是经济发展的核心动力, 不仅提供各类产品满足消费需求, 也支撑产业协同发展[1]。近年来, 随着物联网和大数据的发展, 制造业加速向数字化、智能化转型[2]。作为关键环节, 传统人工仓储因效率低、容错率高、空间利用率低而难以满足现代供应链对效率、准确性与灵活性的要求, 故智能仓储与数字化转型成为主流趋势[3]。

高层货架仓储是智能仓库中的重要一环。智能化立体仓库由高层货架、智能设备与网络控制系统构成, 具备高效率、低成本和高安全性[4]。六向穿梭机系统自动化程度高、设备灵活; 核心作业为入库/出库, 货位分配直接影响效率与空间利用率并关系货架稳定性。多穿梭机协同需合理调度以提升设备利用率、降低空载并实现任务均衡; 同时路径规划对安全与效率至关重要, 可避免冲突与碰撞[5]。

仓储作业的货位分配策略对入库效率、空间利用率及货架稳定性具有决定性影响。早期研究总结了随机存储、基于周转率存储和分类存储等策略[6], 并且学者在优化算法方面开展了大量研究[7]-[11]。在穿梭机仓储中, 也有学者考虑设备特性建立行程时间模型或多目标货位分配模型, 利用改进的混合算法进行优化[12]-[16]。

本文的主要贡献包括: 1) 基于多货位分配原则建立货位分配数学模型; 2) 基于货物出入库效率、货架稳定性和穿梭机作业均衡三个原则构建多目标货位分配优化模型; 3) 设计自适应 DE-NSGA-III算法, 引入反向学习机制和随机扰动提高初始种群多样性, 引入自适应差分变异平衡全局搜索和局部收敛能力。

2. 六向穿梭机仓储货位分配问题分析

2.1. 六向穿梭机仓储存储策略研究

随机存储指由仓储管理系统将到货动态分配至任一可用货位, 不设固定位置。其优点是空间利用率高、动态分配位置, 能随货种与数量波动保持高灵活性; 缺点是高度依赖信息系统, 且可能使高周转货位于远端或关联货物分散, 从而影响出库效率。本文选择随机存储作为货位分配策略。

2.2. 六向穿梭机仓储货位分配原则

货位分配原则建模为:

1) 货物出入库效率原则

在先进先出前提下,将周转率高的货物优先放置在距离出库口更近的位置,减少穿梭机的运行距离,减少货物出入库时间,提高出入库效率。

2) 货架稳定性原则

以安全为先:纵向上,大重量货物优先放置在底层,而小重量货物优先放置在高层,货架整体重心偏低。同时,横向上,同一层的货物尽量保证均匀分布,保证货架受力均匀,防止货架倾斜。

3) 穿梭机作业均衡原则

系统含多巷道与多出库口时,为防止个别巷道/设备过载,在货位分配时,要使同一订单的货物尽量均匀分布在各个巷道上,保证多机并行作业。

3. 多目标货位分配模型构建

3.1. 模型假设

六向穿梭机仓储整体货架布局见图 1。

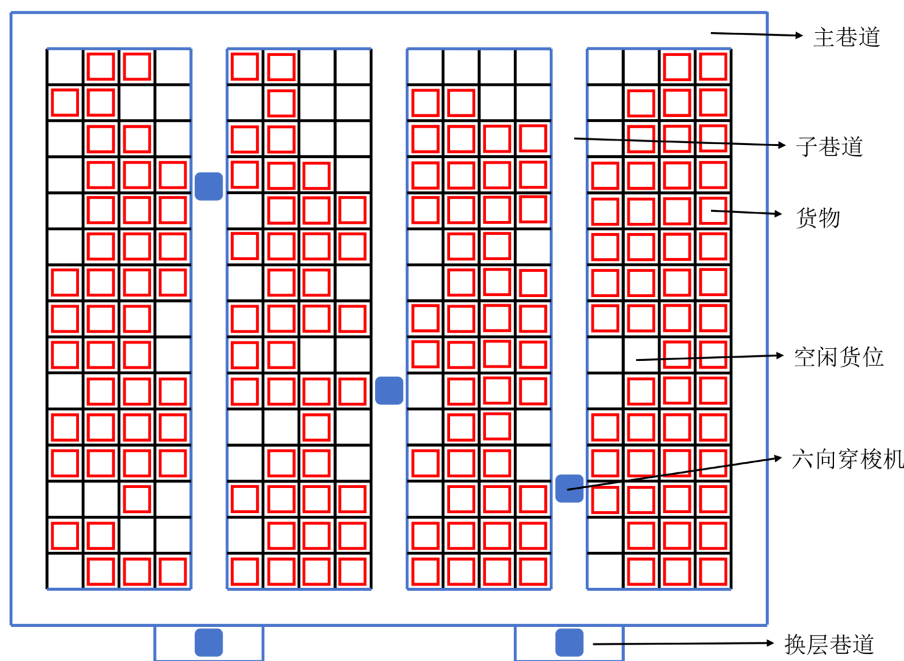


Figure 1. Plan view of the storage racking

图 1. 仓储货架俯视图

为了便于研究和合理地构建货位分配模型,现做出如下假设:

- 1) 同规格,单货位仅存放单个货物,容量可满足所有货类;
- 2) 假设所有穿梭机最大承载能力满足所有货物的存放需求;
- 3) 各车单载运行,即一台穿梭机只搭载一个货物;
- 4) 考虑加速度,且穿梭机在换层巷道以及水平巷道做匀加速、匀速和匀减速运动;
- 5) 不考虑巷道拥堵;
- 6) 转向、进出/等待换层等时间视为定值并在效率计算中忽略;
- 7) 只考虑货物质量,不考虑穿梭机质量;

8) 假设货物周转率、货物是否空闲等信息已知。

3.2. 参数符号定义

文中用到的相关参数及定义见表 1。

Table 1. Model parameters and definitions

表 1. 模型相关参数及定义

参数或变量	含义
$x_i / y_i / z_i$	第 i 个货物所在的列数/排数/层数
p_i / w_i	第 i 个货物的周转率/质量
$C / R / L$	货架总列数/总排数/总层数
m	总巷道数
$L_a / L_b / L_h$	货位长度/宽度/高度
W	货架能承受的最大重量
a_E / a_L	换层巷道/水平巷道中穿梭机加/减速度
$v_{E \max} / v_{L \max}$	换层巷道/水平巷道中穿梭机最大速度
n_l	第 l 层已占用的货位数
$\bar{n}$	货位分配后平均每层已占用货位数
t_k	第 k 个巷道总作业时间
$\bar{t}$	平均每个巷道作业时间

3.3. 货位分配模型构建

1) 货物出入库效率模型

作为衡量立体仓储性能的重要指标之一，货物出入库效率是非常重要的货位分配原则之一。以货物的入库时间为优化目标，在货位分配时，穿梭机在换层巷道中运行时间和在水平巷道中运行的时间和最短，仓储运行效率越高。

在这个过程中，假设共有 N 个货物入库，第 i 个货物被分配到坐标为 (x_i, y_i, z_i) 货位，其中 $x_i \in [1, C]$ ， $y_i \in [1, R]$ ， $z_i \in [1, L]$ 。其入库作业经过第 j 个换层巷道，其横坐标为 x_j 。则入库时间 T_i 为换层巷道行驶时间 T_E^i ，纵深巷道行驶时间 T_X^i 以及水平巷道行驶时间 T_Y^i 三者之和，即有式

$$T_i = T_E^i + T_X^i + T_Y^i \quad (1)$$

考虑穿梭机在换层巷道和水平巷道的加速度，故存在两种运动过程：穿梭机加速到最大速度 $v_{\max}$ 运行；穿梭机未加速到最大速度 $v_{\max}$ 运行。两种运动过程速度 - 时间曲线如图 2 所示。

a) 穿梭机在换层巷道上行驶

当其不能达到最大速度 $v_{\max}$ 时，提升时间 T_E^i 为：

$$T_E^i = 2 \sqrt{\frac{z_i L_h}{a_E}} \quad (2)$$

当其能够达到最大速度 $v_{\max}$ 时，提升时间 T_E^i 为：

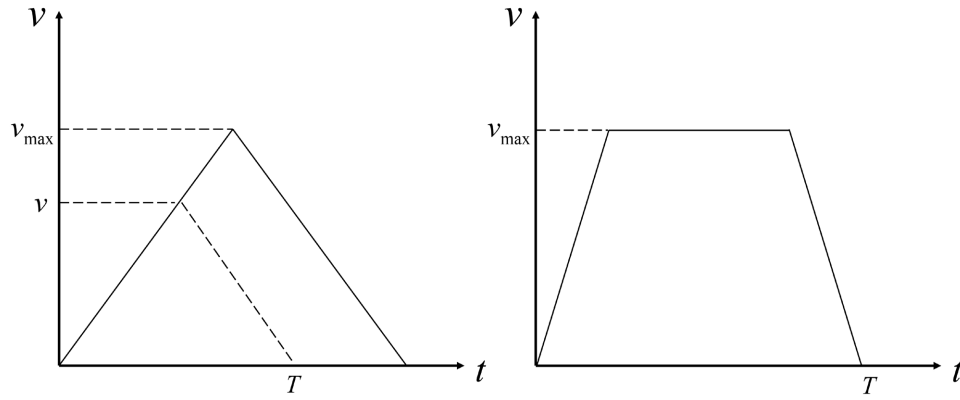


Figure 2. Velocity-time curves for two motion profiles
图 2. 两种运动过程速度 - 时间曲线图

$$T_E^i = 2 \frac{v_{E \max}}{a_E} + \frac{z_i L_h - \frac{v_{E \max}^2}{a_E}}{v_{E \max}} \quad (3)$$

即提升时间 T_E^i 的计算方法为:

$$T_E^i = \begin{cases} 2 \sqrt{\frac{z_i L_h}{a_E}} & z_i L_h \leq \frac{v_{E \max}^2}{a_E} \\ 2 \frac{v_{E \max}}{a_E} + \frac{z_i L_h - \frac{v_{E \max}^2}{a_E}}{v_{E \max}} & z_i L_h > \frac{v_{E \max}^2}{a_E} \end{cases} \quad (4)$$

b) 穿梭机在主巷道行驶

当穿梭机到达 z_i 层后, 易知其沿纵深巷道方向位移为 $|x_i - x_j| L_a$, 则同理可得纵深巷道行驶时间 T_X^i 的计算方法为:

$$T_X^i = \begin{cases} 2 \sqrt{\frac{|x_i - x_j| L_a}{a_L}} & |x_i - x_j| L_a \leq \frac{v_{L \max}^2}{a_L} \\ 2 \frac{v_{L \max}}{a_L} + \frac{|x_i - x_j| L_a - \frac{v_{L \max}^2}{a_L}}{v_{L \max}} & |x_i - x_j| L_a > \frac{v_{L \max}^2}{a_L} \end{cases} \quad (5)$$

c) 穿梭机在子巷道行驶

易知穿梭机沿水平巷道方向位移为 $|y_i - 1| L_b$, 则同理可得水平巷道行驶时间 T_Y^i 的计算方法为:

$$T_Y^i = \begin{cases} 2 \sqrt{\frac{|y_i - 1| L_b}{a_L}} & |y_i - 1| L_b \leq \frac{v_{L \max}^2}{a_L} \\ 2 \frac{v_{L \max}}{a_L} + \frac{|y_i - 1| L_b - \frac{v_{L \max}^2}{a_L}}{v_{L \max}} & |y_i - 1| L_b > \frac{v_{L \max}^2}{a_L} \end{cases} \quad (6)$$

根据货位分配原则, 需要将周转率高的货物放置在距离出入库口较近的货位。因此将入库时间 T 与货物周转率 p_i 的乘积表征货物出入库效率, 乘积越小, 表示出入库效率越高。其表示方法为:

$$f_1 = \sum_{i=1}^N T_i p_i = \sum_{i=1}^N (T_E^i + T_X^i + T_Y^i) p_i \quad (7)$$

2) 货架稳定性模型

货架稳定性也是衡量仓储性能的一个重要指标。在进行货位分配时，应尽量使整个货架的重心尽可能低，即“下重上轻”，将大重量的货物放置在底层，稳定重心，避免发生倾斜、倒塌。在这个过程中，不考虑货架本身重量，货架稳定性评估目标通常为货物的质量和货物所在货位层乘积之和，函数值越小，稳定性越高。其计算方法为：

$$f_2 = \sum_{i=1}^N w_i z_i \quad (8)$$

3) 层级作业均衡模型

对于六向穿梭机仓储系统而言，轨道畅通对仓储运行效率至关重要。穿梭机在密集式仓储内大多沿轨道按直线行走，如果仅考虑货架稳定性原则，那么货物将优先聚集于低层货位，多台穿梭机频繁在某些层巷道内行走，容易造成某层穿梭机拥堵而其他层闲置率较高的情况。因此，建立穿梭机作业均衡模型，使用每层占用货位的标准差作为评估函数，来反映货物分配的均衡程度，函数值越小，作业越均衡，其计算方法为：

$$f_3 = \sqrt{\frac{1}{L} \sum_{l=1}^L (n_l - \bar{n})^2} \quad (9)$$

其中， n_l 表示第 l 层分配的货物数量，引入决策变量 δ_{xyl} 表示坐标为 (x, y, l) 的货位是否被占用，则 n_l 的计算方法如式(10)所示：

$$n_l = \sum_{x=1}^C \sum_{y=1}^R \delta_{xyl} \quad (10)$$

$$\delta_{xyl} = \begin{cases} 0 & \text{货位}(x, y, l) \text{ 未被占用} \\ 1 & \text{货位}(x, y, l) \text{ 被占用} \end{cases} \quad (11)$$

注 1：若考虑使用各层作业时间的标准差作为评估函数，来反映货物分配的均衡程度，函数值越小，作业越均衡，其计算方法为：

$$f_3 = \sqrt{\frac{1}{L} \sum_{l=1}^L (T_l - \bar{T})^2}$$

其中， T_l 表示第 l 层的作业时间， $\bar{T}$ 表示货位分配后平均每层作业时间， t_{xyl} 为将一件货物分配到 (x, y, l) 时对应的作业时间，则 T_l 的计算方法为：

$$T_l = \sum_{x=1}^C \sum_{y=1}^R \delta_{xyl} t_{xyl}$$

4) 巷道均衡模型

巷道均衡模型用于保证各个巷道中的总作业时间均衡，将穿梭机作业时间均匀地分布在各个巷道中，防止某个巷道拥堵而其他巷道闲置率较高的情况，使用各个巷道总作业时间的标准差作为评估函数，函数值越小，巷道作业越均衡，其计算方法为：

$$f_4 = \sqrt{\frac{1}{m} \sum_{k=1}^m (t_k - \bar{t})^2} \quad (12)$$

5) 模型约束条件

货物唯一性约束, 表示每个货物只能存放在一个货位上, 引入决策变量 V_{xyz} 表示第 i 个货物是否存放在坐标为 (x, y, z) 的货位上。 V_{xyz} 的表示方法为:

$$V_{xyz} = \begin{cases} 0 & \text{货物 } i \text{ 未放在 } (x, y, z) \text{ 上} \\ 1 & \text{货物 } i \text{ 放在 } (x, y, z) \text{ 上} \end{cases} \quad (13)$$

货架承重能力约束, 所有货物的质量和不能超过货架能承载的最大重量, 即:

$$\sum_{i=1}^N w_i \leq W \quad (14)$$

货架容量约束, 用坐标 (x, y, z) 表示货位, 其中 x 、 y 、 z 分别不能超过货架的最大列数、排数、层数, 即:

$$\begin{cases} 1 \leq x \leq C \\ 1 \leq y \leq R \\ 1 \leq z \leq L \end{cases} \quad (15)$$

综上所述, 根据货物出入库效率、货架稳定性、穿梭机层级和巷道级作业均衡原则, 建立货物入库时货位分配的数学模型为:

$$\begin{cases} F_1 = \min f_1 = \min \sum_{i=1}^N (T_E^i + T_X^i + T_Y^i) p_i \\ F_2 = \min f_2 = \min \sum_{i=1}^N w_i z_i \\ F_3 = \min f_3 = \min \sqrt{\frac{1}{L} \sum_{i=1}^L (n_i - \bar{n})^2} \\ F_4 = \min f_4 = \min \sqrt{\frac{1}{m} \sum_{k=1}^m (t_k - \bar{t})^2} \end{cases} \quad (16)$$

约束条件为:

$$\begin{cases} V_{xyz} \in \{0, 1\} \\ \sum_{i=1}^N w_i \leq W \\ 1 \leq x \leq C \\ 1 \leq y \leq R \\ 1 \leq z \leq L \end{cases} \quad (17)$$

4. 自适应优化算法设计

六向穿梭机的入库货位分配问题综合考虑货物入库效率原则、货架稳定性原则、穿梭机作业均衡原则等多个原则, 隶属于多目标优化问题。本文研究的六向穿梭机入库货位分配问题拟采用遗传算法来解决, 遗传算法具有强大的全局搜索能力和适应性。传统的遗传算法并不能解决多目标优化问题, 而 DebK 等人[17]在 2002 年提出快速非支配排序遗传算法(NSGA-II), 在传统遗传算法的框架上, 引入了快速非支配排序遗传算法, 极大地提高了多目标优化问题的求解效率, 具有较强的灵活性和适应性, 同时保持了收敛性和多样性的平衡。但是 NSGA-II 一般用于解决 3 个以下优化目标的多目标优化问题, 而本文研究的内容包含 4 个优化目标, 因此采用 NSGA-III 来求解。NSGA-III 是 DebK 等[18]人于 2014 年提出的, 是对 NSGA-II 的进一步改进, 引入了参考点的策略, 大大提高了解决高维度多目标优化问题的效率。

4.1. 编解码设计

本文对入库货位分配采用整数编码：仓库有 21 列 17 排 8 层，其中第 1，6，11，16 列和第 1 排为巷道。按栅格化将全部货位与巷道统一编号 1~2856；第一层编号如图 3 所示。

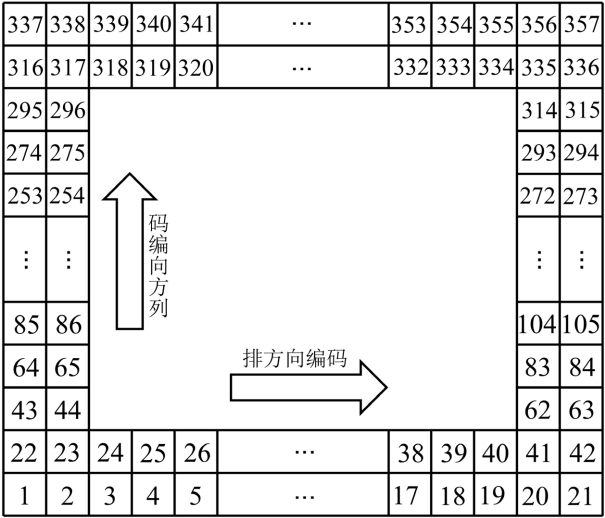


Figure 3. Coding scheme for first-level storage locations
图 3. 第一层货位编码

建立货位坐标和货位编号的一一对应关系，式(18)可将货位编号解码成货位坐标 (x, y, z) ：

$$\begin{cases} x = (n - 1) \% 21 + 1 \\ y = (n - 1) // 21 \% 17 + 1 \\ z = (n - 1) // 357 + 1 \end{cases} \quad (18)$$

其中，坐标 (x, y, z) 表示第 x 列第 y 排第 z 层的货位， n 为货位编号， $//$ 表示取商， $\%$ 表示取模。

按入库货物数设置染色体长度，每个等位基因为一个货位编号，一条染色体表示一个货位分配方案。如图 4 所示，10 个货物的分配方案可表示为(42, 137, 55, 178, 212, 87, 872, 528, 7, 1258)，其对应坐标见表 2。

Table 2. Correspondence between selected location numbers and coordinates
表 2. 部分货位编号与坐标对应关系

货物编号	货位编号	货位坐标
1	42	(21, 2, 1)
2	1926	(15, 7, 6)
3	1080	(9, 1, 4)
4	178	(10, 9, 1)
5	2101	(1, 16, 6)
6	965	(20, 12, 3)
7	872	(11, 8, 3)
8	528	(3, 9, 2)
9	1682	(2, 13, 5)
10	1258	(19, 9, 4)

货物编号	1	2	3	4	5	6	7	8	9	10
货位编号	42	1926	1080	178	2101	965	872	528	1682	1258

Figure 4. Schematic of chromosome encoding

图 4. 染色体编码示意图

4.2. 改进策略

1) 初始化种群

初始化种群是 NSGA-III 的关键步骤, 直接影响搜索效率和优化结果。考虑局部最优情形, 引入反向学习机制, 同时考虑当前解及其反向解, 以扩大初始种群的覆盖范围、提高多样性。考虑极端情形, 对部分个体施加随机扰动, 即对其货位分配方案中的部分货物随机重新分配货位编号。

假设初始化种群大小为 N , 共有 Q 个货物入库, 随机生成包含 N 个个体的种群 X , 设 X_i 为种群 X 中的一个个体, 则:

$$X_i = \{x_1, x_2, \dots, x_j, \dots, x_Q\}, i = 1, 2, \dots, N \quad (19)$$

其中 X_i 为第 i 个个体的染色体基因编码, x_j 为编码值, $x_j \in [1, A]$, $j = 1, 2, \dots, Q$, $[1, A]$ 表示基因编码取值范围, x_j 表示给第 j 个货物分配在货位编号为 x_j 的货位上。基于反向学习机制, 取种群 X 中的 50% 个个体进行反向计算:

$$X_i^* = A - X_i = \{A - x_1, A - x_2, \dots, A - x_j, \dots, A - x_Q\}, i = 1, 2, \dots, \frac{N}{2} \quad (20)$$

则由反向计算后的个体和剩余的原有个体组成新的种群 X^* 。

为提升多样性, 引入随机扰动。在 X^* 中随机选 20% 个体, 并将其 20% 基因重新随机赋值, 保证货位编号不重复。将 $0.2N$ 个染色体中的 $0.2Q$ 个货物的货位编号进行二次随机分配, 最终得到规模为 N 的初始种群 X_t (见图 5)。

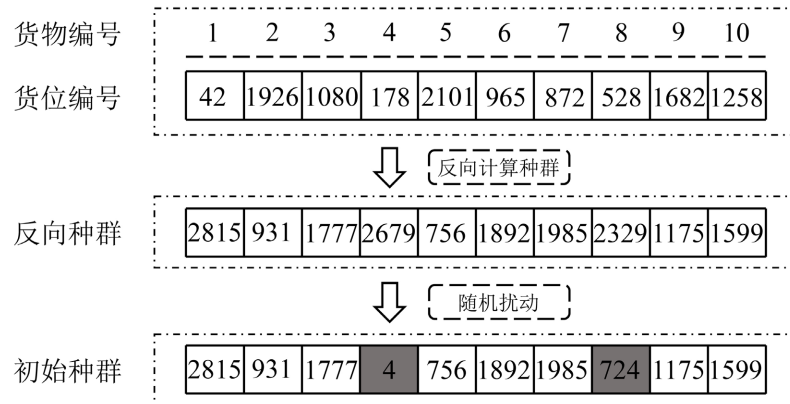


Figure 5. Schematic of initial population generation

图 5. 初始化种群示意图

2) 自适应差分进化算法

差分进化基于差分分量生成变异向量 v_i^t , 表示为:

$$v_i^t = x_{r1} + F \cdot (x_{r2} - x_{r3}) \quad (21)$$

$$F = \frac{F_{i1} + F_{i2}}{2} \quad (22)$$

$$F_{i1} = F_{\max} e^{\frac{F_{\max} - t}{F_{\min} T}} \quad (23)$$

$$F_{i2} = F_{\max} \left(1 - \frac{d(x_i^t, Z_k)}{d_{\max}} \right) \quad (24)$$

其中, x_{r1} , x_{r2} , x_{r3} 是种群中随机选取的 3 个个体, 三者互不相同, i 为种群中不同于 x_{r1} , x_{r2} , x_{r3} 的个体向量的索引, F 是变异因子, F_{i1} 为基于时间的自适应变异因子, t 为当前迭代次数, T 为迭代总次数, $F_{\max}$ 为变异因子最大值, $F_{\min}$ 为变异因子最小值。 F_{i1} 为基于迭代次数的分因子: 通过指数函数使变异因子非线性递减, 前期取大以提升全局搜索、避免陷入局部最优; 后期取小以保护优质解、强化局部搜索。 F_{i2} 为基于参考点距离的分因子: $d(x_i^t, Z_k)$ 是个体 x_i^t 到其参考点 Z_k 的距离, $d_{\max}$ 为种群中个体到参考点的最大距离, 按差异自适应调节变异强度, $d(x_i^t, Z_k)$ 大(远离 Pareto 前沿)则增强变异, $d(x_i^t, Z_k)$ 小(接近前沿)则减弱变异。两者结合形成自适应策略, 在不同时段兼顾种群多样性与算法收敛性。

父代种群变异后, 通过交叉操作生成试验向量 $u_i^t = \{u_{i,1}^t, u_{i,2}^t, \dots, u_{i,j}^t, \dots, u_{i,Q}^t\}$, 通过结合变异向量 v_i^t 和当前个体 x_i^t , 将父代个体的某些特性传递到下一代:

$$u_{i,j}^t = \begin{cases} v_{i,j}^t & \text{rand}(0,1) \leq CR \text{ 或 } j = j_{\text{rand}} \\ x_{i,j}^t & \text{其他} \end{cases} \quad (25)$$

$$CR = CR_{\min} + (CR_{\max} - CR_{\min}) \cdot \frac{t}{T} \quad (26)$$

其中, $u_{i,j}^t$ 表示第 t 代种群中第 i 个个体的第 j 维试验向量, CR 为交叉算子, 用于控制在一定概率下将变异向量 v_i^t 中的某些分量传递给试验向量 u_i^t , CR 的计算方式采用基于迭代次数的线性增长方式, 降低前期交叉概率, 尽量保留当前个体信息, 后期逐步增大交叉概率, 扩大搜索范围。 j_{rand} 为范围 $[1, Q]$ 的随机整数, 起到保证作用, 即使当 $\text{rand}(0,1) > CR$ 时, 在 $j = j_{\text{rand}}$ 时, 仍然会发生交叉操作。

在多目标优化中, 传统遗传算法每代从与种群规模相同的父代产生子代, 个体选择需兼顾适应度与收敛性。NSGA-II 在此框架下引入快速非支配排序与拥挤度, 但当目标数增多时效果受限; NSGA-III 进一步加入参考点机制, 提升高维目标下的选择效率与解集分布质量。

个体之间的支配关系是通过如下方式定义的, 假设个体 A 支配个体 B, 如果: 1) 个体 A 的所有目标函数值都不劣于个体 B; 2) 个体 A 在至少一个目标函数上优于个体 B。

而非支配等级是指根据 Pareto 原则将种群分级: 第一级为不被任何个体支配的解(相对最优集); 第二级被第一级支配但不被其他个体支配, 依此类推。

传统非支配排序需对个体两两比较并逐层重复, 种群规模为 N , 优化目标个数为 M 个时, 传统非支配排序算法时间复杂度为 $O(MN^3)$ 。因此, 当种群规模过大时, 计算复杂度也会大大提升。快速非支配排序算法改进了传统算法, 通过支配集合 $S(p)$ 和支配计数 n_p 记录每个个体, 其中 $S(p)$ 表示 p 支配的个体的集合, n_p 表示支配 p 的个体的数量, 通过判断 n_p 是否为 0 来决定 p 是否属于当前层, 并持续递归迭代, 直到所有个体都分层, 快速非支配排序算法具体实现步骤如下:

Step 1: 遍历种群中所有个体 p , 并计算 n_p 和 $S(p)$;

Step 2: 对于种群中的每个个体 p , 如果 $n_p = 0$, 将 p 加入到第 1 层 F_1 ;

Step 3: 对于当前层 F_i , 如果 F_i 非空, 那么对于当前层中的每个个体 $p \in F_i$, 遍历 $S(p)$ 中的所有个体 q , 令 $n_q = n_q - 1$;

Step 4: 如果 $n_q = 0$, 将 q 加入下一层 F_{i+1} , 并令 $i = i + 1$;

Step 5: 重复 Step 3 和 Step 4 , 直到所有个体都被分配到某一层。

在差分进化的选择环节, NSGA-III 先比较变异向量 u_i^t 与当前个体 x_i^t 的非支配等级, 取等级更高者; 若等级相同, 则启用参考点机制, 选择距参考点更近的个体进入下一代。参考点用于在多目标(高维)下保持解分布均匀并引导进化, 距离越小通常越接近 Pareto 前沿。

4.3. 自适应 DE-NSGA-III 算法

自适应 DE-NSGA-III 通过反向学习提升初始多样性, 结合 DE 与 NSGA-III, 并依据迭代进度与个体差异自适应调节变异因子与交叉因子(不破坏优质解), 从而兼顾前期全局搜索与后期快速收敛。自适应 DE-NSGA-III 算法流程图如图 6 所示, 初始化种群先进行种群随机生成, 其次引入反向机制, 再将随机扰动加入, 通过自适应差分进化算法进行复制、变异和交叉, 并通过快速非支配排序以及参考点机制进行新一代种群选择。

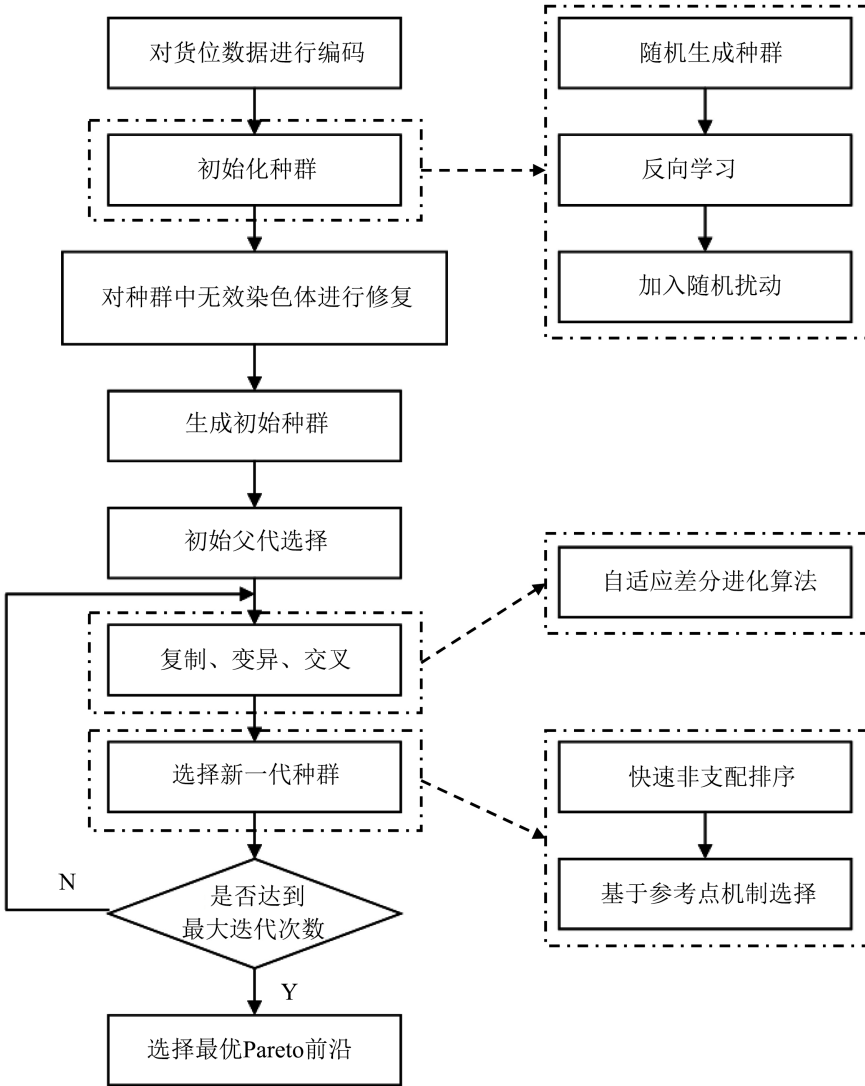


Figure 6. Workflow of the adaptive DE-NSGA-III algorithm
图 6. 自适应 DE-NSGA-III 算法流程图

5. 实验验证与分析

5.1. 实验平台及设计

根据 H 集团六向穿梭机密集式立体仓库设计，货架总数为 17 排 21 列 8 层，除去巷道位置，总货位数为 1920 个。仓库用于存储各种型号整箱产品，仓储设备各项参数如表 3 所示。

Table 3. Parameters of the warehouse equipment
表 3. 仓储设备参数表

参数名称	符号表示	参数值
货架总列数/总排数/总层数	$C/R/L$	21 列/17 排/8 层
穿梭机水平/垂直方向加/减速度	a_L/a_E	$0.5\text{ m/s}^2; 0.5\text{ m/s}^2$
穿梭机水平/垂直方向最大速度	$v_{L\max}/v_{E\max}$	2 m/s; 1.5 m/s
货位长度/宽带/高度	$L_a/L_b/L_h$	1 m/1 m/0.6 m

使用 Python 进行实验验证，实验平台配置如表 4 所示。

Table 4. Configuration of the experimental platform
表 4. 实验平台配置

设备	配置
操作系统	Windows 11 家庭中文版
CPU	Intel(R) Core(TM) i5-9300H
主频	2.40 GHz
内存	8.00 GB
开发环境	PyCharm 2023.1

基于上述参数和实验平台，随机生成 200 个待分配的货物，货物属性包括重量、周转率。为验证模型有效性，在相同条件下进行对比：用自适应 DE-NSGA-III 对四个目标函数做多目标优化；用遗传算法对四个目标函数做单目标优化。两种方法各进行 30 次实验，取平均结果进行分析。

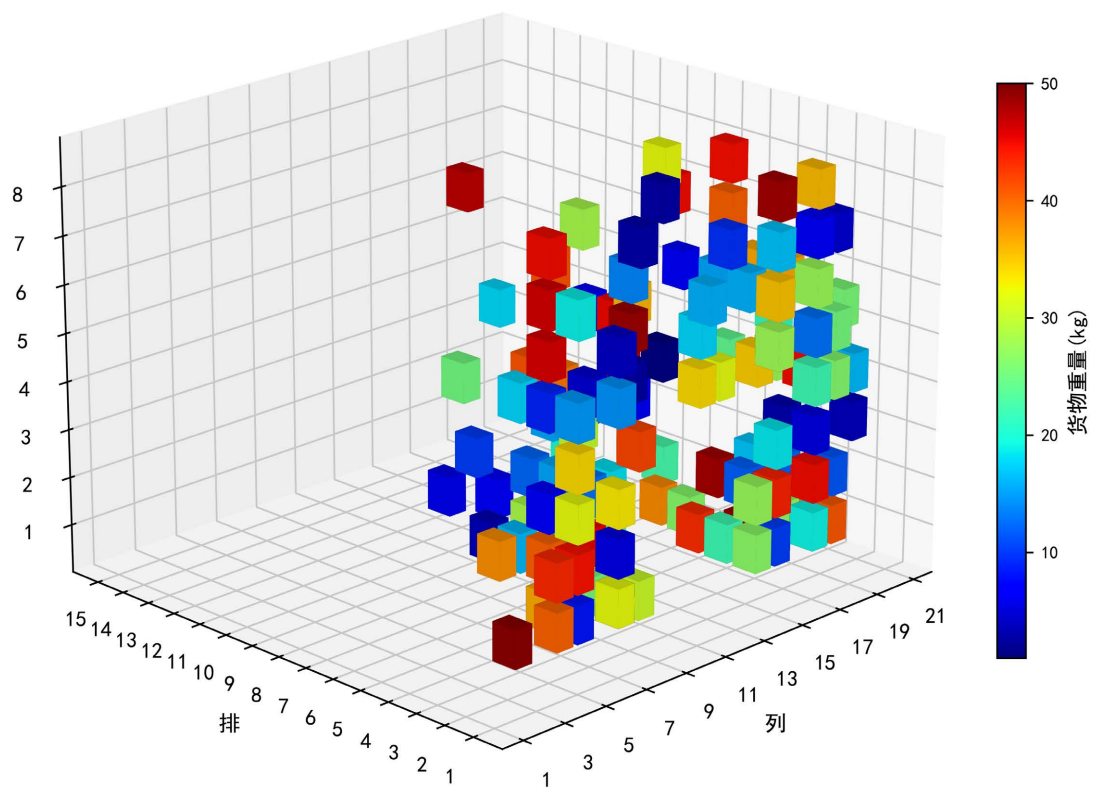
5.2. 实验结果与分析

图 7 给出了使用遗传算法进行优化的货位分配结果，红色/蓝色分别表示重量较大/较小的货物。

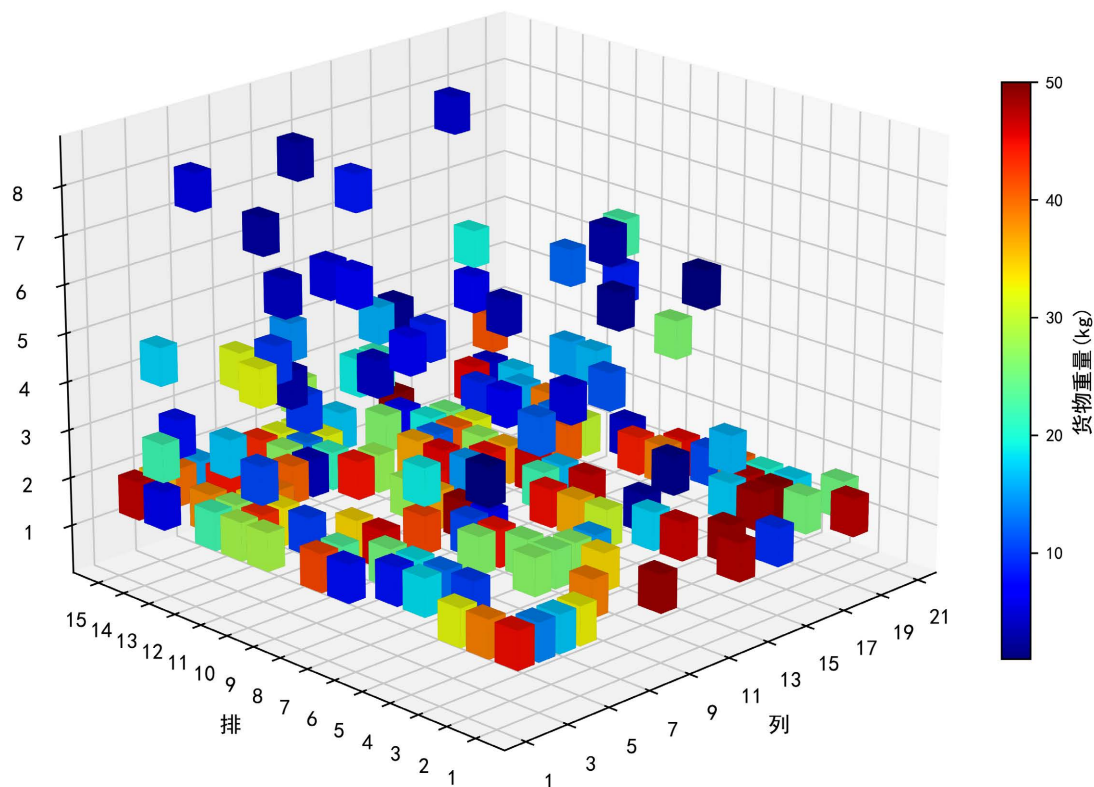
图 7(a)为基于货物出入库效率模型的分配结果，可以看出货物集中在靠近出库口位置，总入库时间较小；但未考虑稳定性，部分重货在上层，重心偏高。

图 7(b)为基于货架稳定性模型的分配结果，可以看出重货集中底层、顶层少货，重心低；但层级分布不均，底层任务过多，易增大冲突、降低效率。

图 7(c)和图 7(d)分别为基于层级作业均衡模型和巷道级作业均衡模型的分配结果，可以看出分配总体均匀、近似随机，导致总入库时间增加。



(a) f_1 优化



(b) f_2 优化

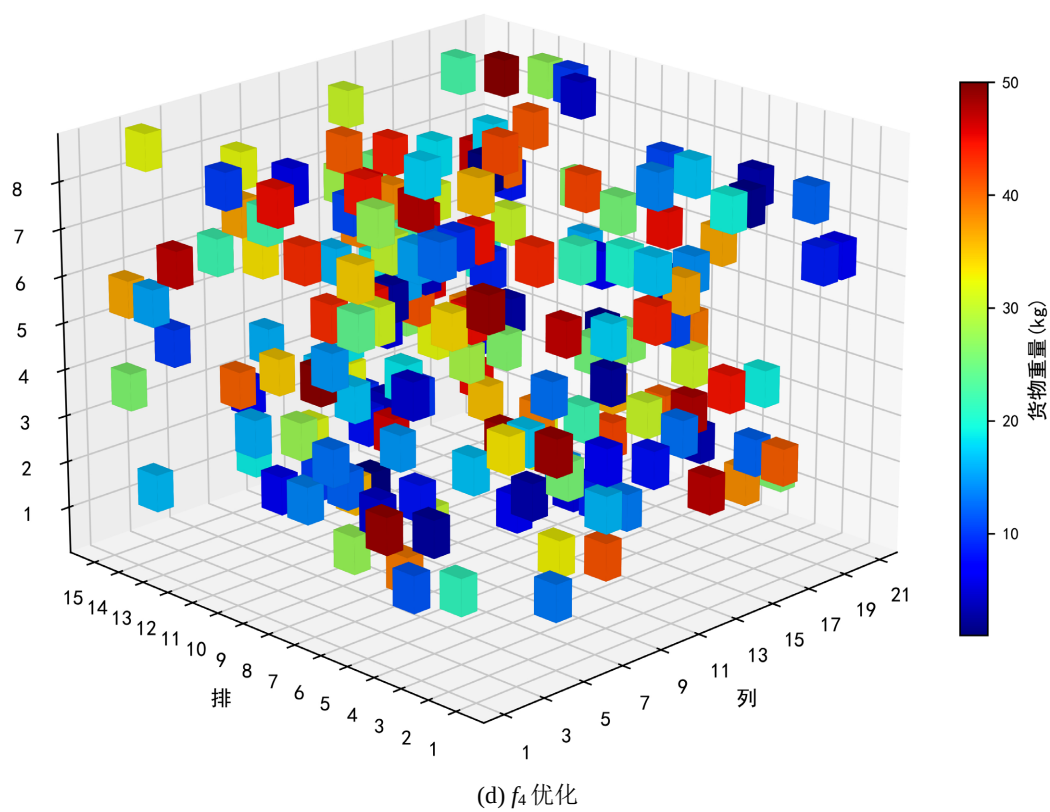
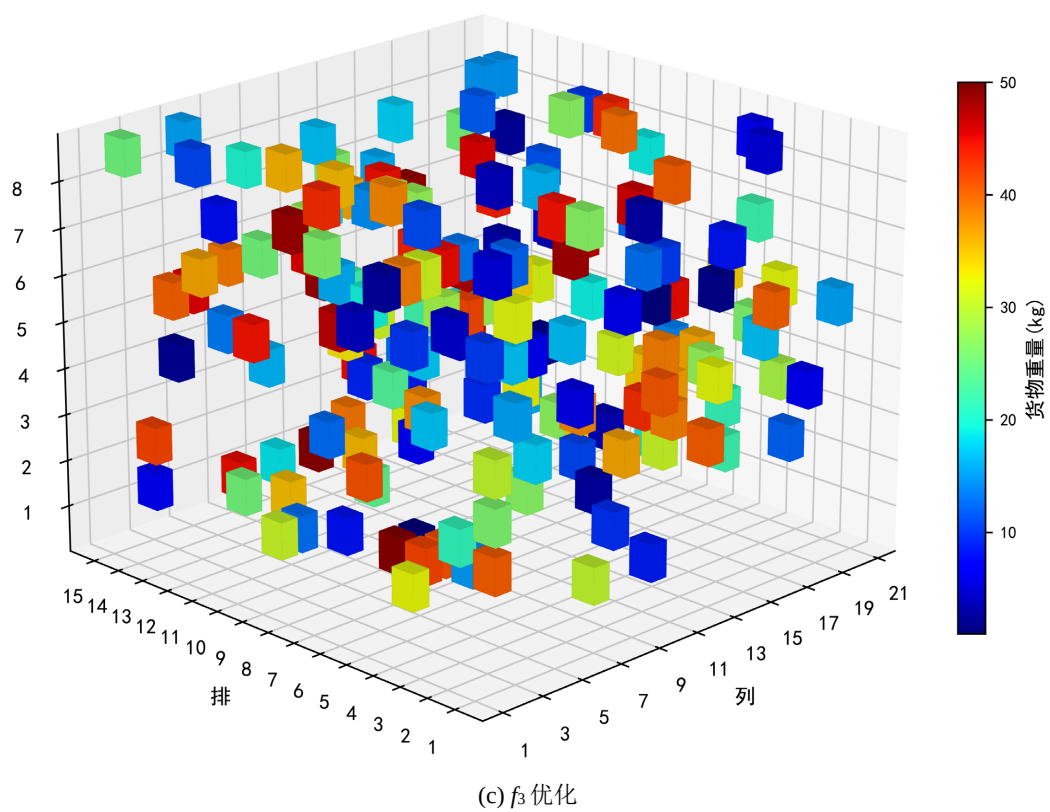


Figure 7. Results of single-objective optimization
图 7. 单目标优化结果

在相同属性的 200 个货物上采用自适应 DE-NSGA-III 进行分配。每次实验从得到的 Pareto 解集中选取综合表现最佳的解，重复 30 次取四个目标的平均值，并与单目标 GA 结果对比(见表 5)。表明在多目标权衡下，出入库效率、货架稳定性、巷道均衡三目标的表现均接近各自单目标最优；层级均衡目标也较好，但仍不及“巷道均衡”单目标优化的数值。原因是每层巷道数固定，优化巷道均衡会同时影响层级均衡，二者正相关。

Table 5. Objective function values
表 5. 目标函数值

优化目标	f_1	f_2	f_3	f_4
多目标	897.19	13671.83	9.25	0.99
f_1	459.06	15930.80	15.82	3.83
f_2	1286.63	6168.40	48.57	1.03
f_3	1426.11	21740.12	0.50	2.76
f_4	1436.28	22721.45	5.50	0.00

自适应 DE-NSGA-III 输出的是 Pareto 解集，最终方案可按偏好选择。图 8 为综合各目标的方案：货物整体靠近出库口，重货下层、轻货上层，层级与巷道分布较均匀。综合表现良好，优于单目标优化；图 9 为分别取四个目标的最优方案。对比可见：图 9(a)货物更靠近入库口；图 9(b)重心更低、货架更稳；图 9(c)和图 9(d)分布更均匀，层级/巷道作业更均衡，降低拥堵。

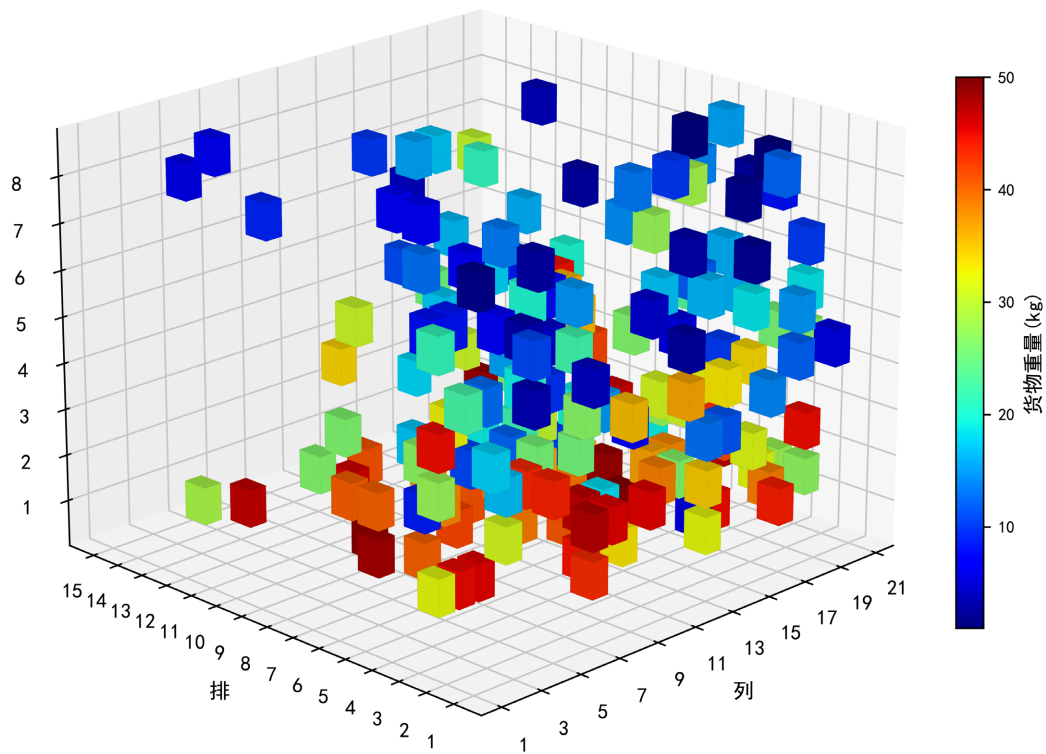
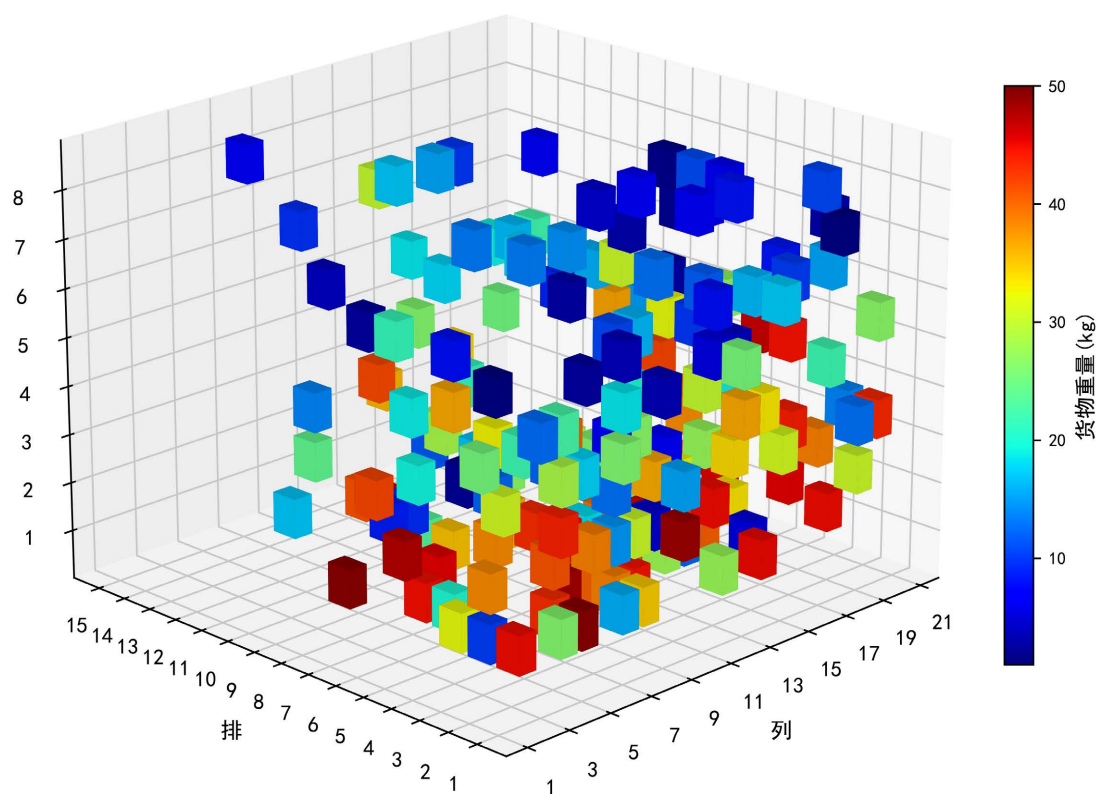
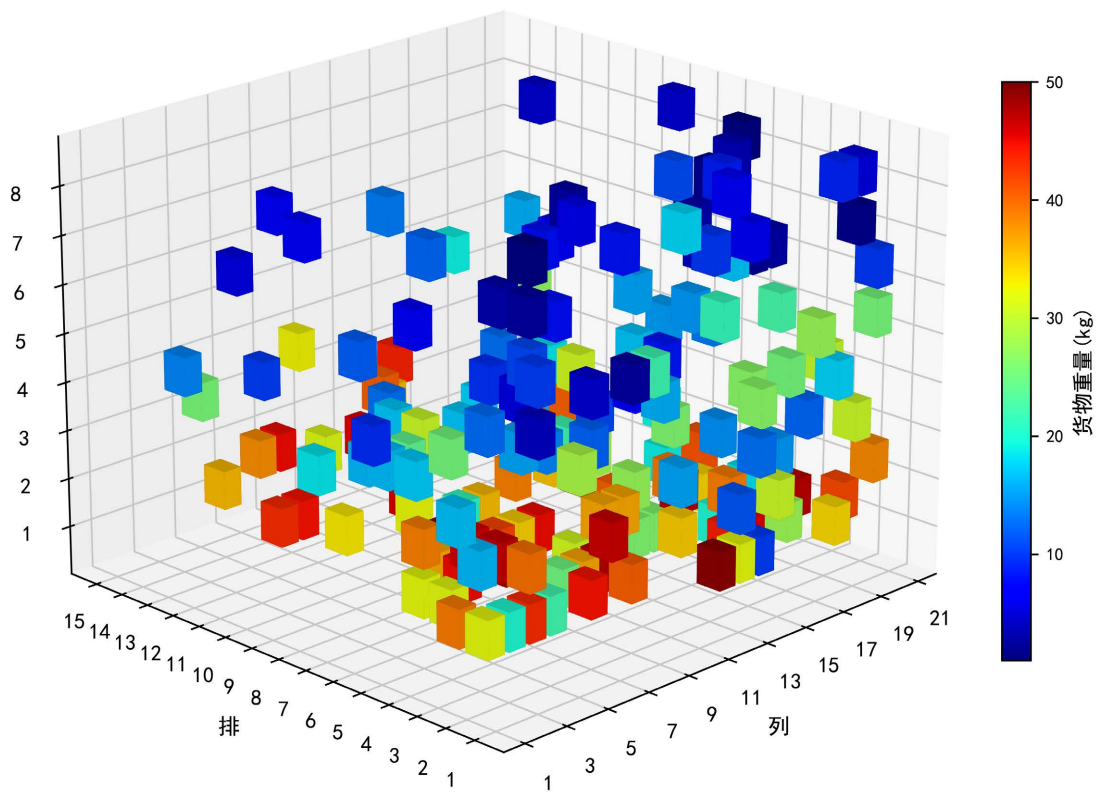


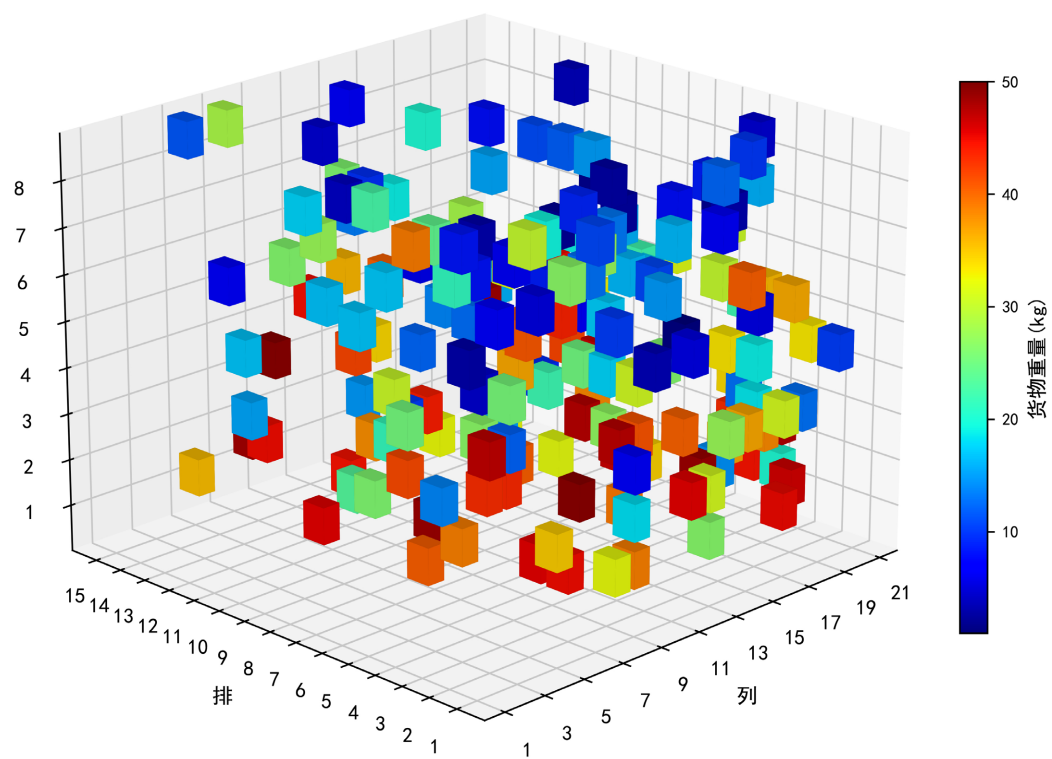
Figure 8. Results of multi-objective storage location assignment
图 8. 多目标优化货位分配结果



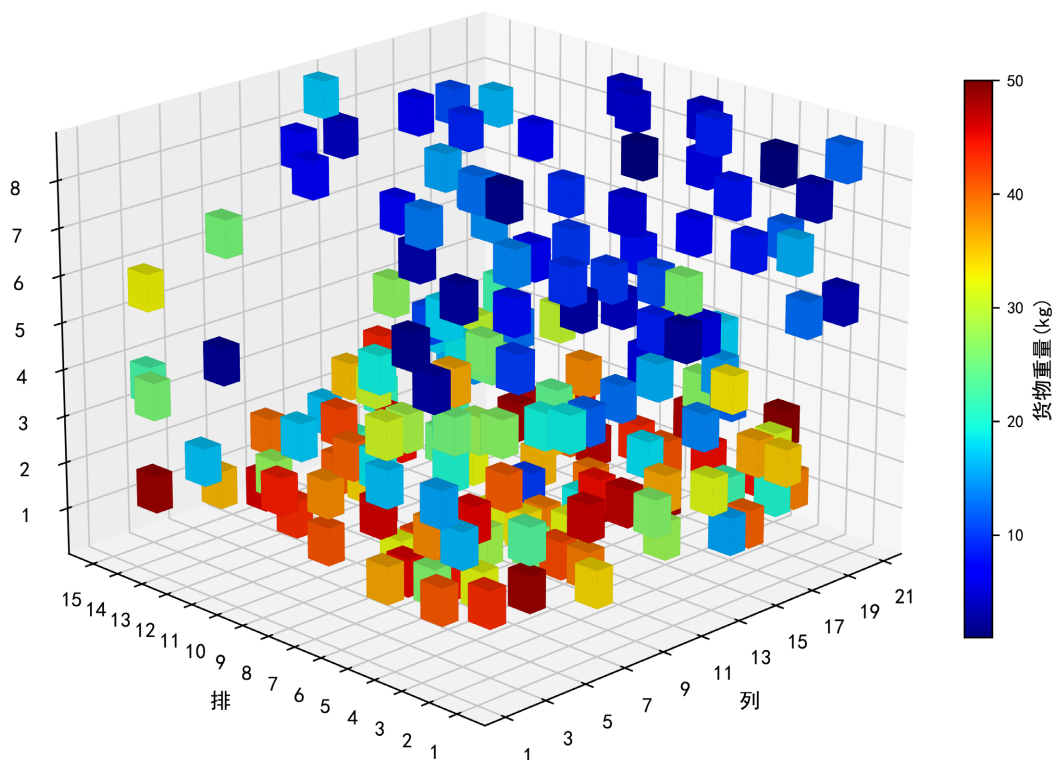
(a) 货物出入库效率最优



(b) 货架稳定性最优



(c) 层级作业均衡最优



(d) 巷道作业均衡最优

Figure 9. Storage location assignment schemes
图 9. 货位分配方案

6. 结论

本文针对入库货位分配问题, 首先分析了随机存储策略, 同时考虑货物出入库效率、货架稳定性和穿梭机作业均衡三个核心原则, 构建了多目标货位分配优化模型。为了求解该模型, 设计了自适应 DE-NSGA-III 算法, 通过引入反向学习机制和随机扰动提高初始种群多样性, 引入自适应差分变异平衡全局搜索和局部收敛能力。实验结果表明该算法能够得到一组综合性能良好的货位分配方案。

参考文献

- [1] ElMaraghy, H., Monostori, L., Schuh, G. and ElMaraghy, W. (2021) Evolution and Future of Manufacturing Systems. *CIRP Annals*, **70**, 635-658. <https://doi.org/10.1016/j.cirp.2021.05.008>
- [2] Kasilingam, S., Yang, R., Singh, S.K., Farahani, M.A., Rai, R. and Wuest, T. (2024) Physics-Based and Data-Driven Hybrid Modeling in Manufacturing: A Review. *Production & Manufacturing Research*, **12**, Article ID: 2305358. <https://doi.org/10.1080/21693277.2024.2305358>
- [3] Kern, J. (2021) The Digital Transformation of Logistics: A Review about Technologies and Their Implementation Status. In: Sullivan, M. and Kern, J., Eds., *The Digital Transformation of Logistics: Demystifying Impacts of the Fourth Industrial Revolution*, The Institute of Electrical and Electronics Engineers, Inc., 361-403.
- [4] Palkina, E. (2022) Transformation of Business Models of Logistics and Transportation Companies in Digital Economy. *Transportation Research Procedia*, **63**, 2130-2137. <https://doi.org/10.1016/j.trpro.2022.06.239>
- [5] 国务院关于印发《中国制造 2025》的通知[J]. 中华人民共和国国务院公报, 2015(16): 10-26.
- [6] Hausman, W.H., Schwarz, L.B. and Graves, S.C. (1976) Optimal Storage Assignment in Automatic Warehousing Systems. *Management Science*, **22**, 629-638. <https://doi.org/10.1287/mnsc.22.6.629>
- [7] Jiang, W., Liu, J., Dong, Y. and Wang, L. (2020) Assignment of Duplicate Storage Locations in Distribution Centres to Minimise Walking Distance in Order Picking. *International Journal of Production Research*, **59**, 4457-4471. <https://doi.org/10.1080/00207543.2020.1766714>
- [8] Kübler, P., Glock, C.H. and Bauernhansl, T. (2020) A New Iterative Method for Solving the Joint Dynamic Storage Location Assignment, Order Batching and Picker Routing Problem in Manual Picker-to-Parts Warehouses. *Computers & Industrial Engineering*, **147**, Article ID: 106645. <https://doi.org/10.1016/j.cie.2020.106645>
- [9] 胡祥培, 丁天蓉, 张源凯, 等. 基于关联网路的机器人移动货架系统货位分配方法[J]. 工程管理科技前沿, 2022, 41(1): 56-64.
- [10] Xu, X. and Ren, C. (2020) Research on Dynamic Storage Location Assignment of Picker-to-Parts Picking Systems under Traversing Routing Method. *Complexity*, **2020**, Article ID: 1621828. <https://doi.org/10.1155/2020/1621828>
- [11] 庄燕玲, 孙玉姣, 朱涛, 等. 移动货架系统多机器人“存-取货架”调度优化方法[J]. 系统工程理论与实践, 2023, 43(2): 488-508.
- [12] Lerher, T., Ekren, B.Y., Dukic, G. and Rosi, B. (2015) Travel Time Model for Shuttle-Based Storage and Retrieval Systems. *The International Journal of Advanced Manufacturing Technology*, **78**, 1705-1725. <https://doi.org/10.1007/s00170-014-6726-2>
- [13] Lerher, T. (2015) Travel Time Model for Double-Deep Shuttle-Based Storage and Retrieval Systems. *International Journal of Production Research*, **54**, 2519-2540. <https://doi.org/10.1080/00207543.2015.1061717>
- [14] 李佳, 何非, 谢刚伟, 等. 四向穿梭车式密集仓储入库货位分配方法研究[J]. 计算机工程与应用, 2023, 59(24): 328-335.
- [15] 吴自松, 茆道方, 盖宇春. 基于两阶段混合算法的四向穿梭车式密集仓储系统货位分配优化[J]. 系统仿真学报, 2025, 37(5): 1234-1245.
- [16] 谢书达. 智能仓储系统中货位分配及任务调度算法研究与模拟[D]: [硕士学位论文]. 上海: 东华大学, 2020.
- [17] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**, 182-197. <https://doi.org/10.1109/4235.996017>
- [18] Deb, K. and Jain, H. (2014) An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation*, **18**, 577-601. <https://doi.org/10.1109/tevc.2013.2281535>