

基于 AGA 与 MPSO 的非传统布局仓储货位分配优化^①

胡颖聪^{②*} 刘建胜^{③*} 张有功^{**}

(* 南昌大学经济管理学院 南昌 330031)

(** 南昌大学机电工程学院 南昌 330031)

摘 要 非传统布局是现代仓储管理的新热点,根据对非传统布局(Fishbone 型)特征分析,针对货位分配优化问题,提出以出入库效率和货架稳定性为优化目标,建立多目标优化模型。设计了自适应遗传算法(AGA)和改进的粒子群优化算法(MPSO)进行求解。AGA 采用动态自适应策略改进选择、交叉、变异算子,克服初期“早熟”,提高末期局部搜索,增强鲁棒性;考虑到 PSO 搜索过程的非线性复杂特征,引入非线性变化的惯性权重和时变加速的学习因子,提升早期全局搜索能力,改善末期收敛迟钝,优化算法整体性能。采用 Matlab 进行仿真实验,结合实例验证了本文方法的有效性与通用性。对比实验结果表明 AGA 在处理此类货位分配优化问题上优势更明显。

关键词 非传统布局, 货位分配优化, 自适应遗传算法(AGA), 改进粒子群优化算法(MPSO)

0 引 言

作为现代制造业重要组成部分,仓储是连接生产制造与销售服务的桥梁,高效的仓储运营不仅能及时响应客户需求,还能有效降低成本^[1]。而仓储效率又受多方面因素影响^[2],其中非传统布局仓储是当前重点议题之一。

Gue 等^[3]打破传统布局的思维禁锢,在一定条件下得出 Flying-V 型能比传统型平均减少11.2% 的拣货路径,而 Fishbone 型优化效果更明显,拣货路径可减少20.7%。这表明传统布局可能并不是最优的,从而也启发了业界对布局设计的重新思考和放松假设条件的研究。Pohl 等^[4]基于双命令模式下探讨了 Fishbone 型的最优布局设计,后来又对不同需求偏态下的布局设计进行了分析^[5]。Öztürkçülu 等^[6]

在 Fishbone 型基础上进一步放松假设,设计了某些场景下更优的 Chevron 型布局。Martínez^[7]则从最佳主通道角度对 Fishbone 型特点进行分析,着重研究了主通道斜率设计的问题。另外,Clark 等^[8]和 Cardona 等^[9]则考虑货架高度对作业时间的影响,提出了对作业时间模型进行修改。国内方面,周丽等^[10]在 2 种拣选方式下对 Fishbone 型和传统型的拣货行程距离进行了仿真,得出的结论与 Gue 等^[3]的相似。而蒋美仙^[11]和李乐等^[12]则在 Fishbone 型中结合贯通式思想,提出改进的 Fishbone 型设计,并结合实例进行了验证。由此不难看出非传统布局是当前仓储领域的新热点,不过,以上文献均是探讨非传统布局中通道的优化设计,都没有涉及到货位分配问题。

货位分配是仓储作业中极为重要的环节,良好的货位分配既显著减少货物存取的行程时间又能保证货架结构的稳定性^[13]。货位分配优化方法一直

① 国家自然科学基金(51565036)和江西省研究生创新专项资金(YC2017-S026)资助项目。
② 男,1993 年生,硕士生;研究方向:优化算法,现代仓储管理,物流优化技术;E-mail: 18070074135@163.com
③ 通信作者,E-mail: liujiansheng@ncu.edu.cn
(收稿日期:2018-05-13)

也是管理者密切关注的焦点也是学术界研究的热点,现已有相对丰富的成果^[14-16]。然而,非传统布局与传统仓储之间差异明显,其货位分配优化问题是一个更加复杂的大规模组合优化问题,现有的优化算法和传统仓储的货位分配方法为此提供了借鉴基础,但并不能直接适用在此处。因此,研究设计高效的非传统布局仓储货位分配优化方法具有一定的学术意义和实际应用价值。

1 问题描述与模型构建

1.1 Fishbone 型布局

Fishbone 型是一种典型的非传统布局方式(如图 1 所示),整个仓储可均分为 4 个货区,其中存取点(pickup and deposit point, P&D 点)位于仓储前端底部中心,左右两边有各有 1 条主通道(cross aisle),若干条拣货通道(pick aisle)与主通相交,由此整个仓储布局呈 Fishbone 型。

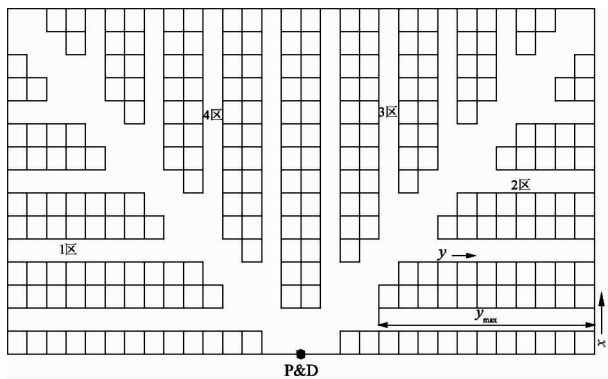


图 1 Fishbone 型布局

1.2 布局参数

该仓储中货位的长和宽均为 l , 高为 h 。 k ($k = 1, 2, 3, 4$) 为货区号, 从左下角区域开始按逆时针方向分别为 1 区、2 区、3 区、4 区; x ($x = 1, 2, \dots, x_{\max}$) 为货位的排数; y ($y = 1, 2, \dots, y_{\max}$) 为货位的列数; z ($z = 1, 2, \dots, z_{\max}$) 为货位的层数。位于 k 区 x 排 y 列 z 层的货位记 (k, x, y, z) , 例如 $(4, 2, 7, 3)$ 表示该货位位于 4 区 2 排 7 列 3 层。 i ($i = 1, 2, \dots, i_{\max}$) 为货品的编号, m_i 为货品的质量, r_i 为货品的存取频率, j_i ($j = 1, 2, \dots, j_{\max}$) 为第 i 类货品的数量。 v_1 为

AGV (automatic guided vehicle) 的水平运行速度, v_2 为其竖直运行速度。另外, 货架的列数 $y_{\max}$ 随着 x 不断变化, 其表达式为

$$y_{\max} = \begin{cases} Y - 1.5(x - 1) & x \text{ 为奇数} \\ Y - 1.5x + 1 & x \text{ 为偶数} \end{cases} \quad (1)$$

式(1)中, Y 为最大列数, 即第一排货架的列数。

L_x 为 AGV 小车从 P&D 点出发到目的货位过程在主通道的行驶距离, 其表达式为

$$L_x = \begin{cases} \sqrt{2}(1 + 1.5(x - 1))l + 2l & x \text{ 为奇数} \\ \sqrt{2}(2 + 1.5(x - 2))l + l & x \text{ 为偶数} \end{cases} \quad (2)$$

1.3 货位分配优化模型

货位分配优化的目的是根据货物以及货架的特性为待入库的货物分配合适的存储位置, 从而降低仓储作业成本, 提高作业效率, 使得仓储效益最大化。在进行货位分配时一般要遵循一些基本原则, 如先进先出原则、高近低远原则、上轻下重原则等^[16]。因此, 本文将出入效率和货架稳定性作为优化目标, 建立多目标货位分配优化模型。

模型假设:

- (1) 货位分配前待入库货物的数量、质量、存取频率等信息已知。
- (2) 同一类的货物可以存放于不同货位, 但一个货位只能存放一个货物。
- (3) v_1 与 v_2 已知且不考虑 AGV 的启动与制动时间。
- (4) 单次行程中 AGV 只能访问一个货位。
- (5) 拣货通道宽度与单排货架的宽度相等。

目标函数:

$$f_1 = \min \sum_{j=1}^{j_{\max}} \sum_{i=1}^{i_{\max}} r_{ij} \times \left(\frac{L_x}{v_1} + \frac{(y-1) \times l}{v_1} + \frac{(z-1) \times h}{v_2} \right) \quad (3)$$

$$f_2 = \min \frac{\sum_{j=1}^{j_{\max}} \sum_{i=1}^{i_{\max}} m_{ij} \times z \times h}{\sum_{j=1}^{j_{\max}} \sum_{i=1}^{i_{\max}} m_{ij}} \quad (4)$$

约束条件:

$$x \leq x_{\max} \quad (5)$$

$$y \leq y_{\max} \quad (6)$$

$$z \leq z_{\max} \tag{7}$$

式(3)表示以货物出入库效率最高建立的目标函数,式(4)表示以货架稳定性最优建立的目标函数。式(5)~(7)为 Fishbone 布局中仓储货位分配的约束条件。

1.4 多目标归一化

当前,处理多目标优化问题的方法比较多,普遍采用的是赋权法^[14,15],这种方法特征明显且简单易执行。但本问题中由于 2 个目标函数量纲不同,值域范围差别大,如果直接采用简单赋权法可能会造成某些目标被弱化,很难得到整体最优效果^[17]。为此,本文采用一种“量纲归一化”方法来消除量纲差异,弥补简单赋权法的缺陷,具体来说就是先求得单目标情况下的最优值,然后将各个单目标函数的量纲归一化。再由决策者根据实际情况对各个子目标函数进行赋权,最后可将多目标问题转化为单目标问题。

量纲归一化得到的子目标函数 F_1 、 F_2 ,总目标函数 F 及适应度函数 G 如下:

$$F_1 = \frac{\hat{f}_2}{\hat{f}_2 + \hat{f}_1} f_1 \tag{8}$$

$$F_2 = \frac{\hat{f}_2}{\hat{f}_2 + \hat{f}_1} f_2 \tag{9}$$

$$F = w_1 \times F_1 + w_2 \times F_2 \tag{10}$$

$$G = \frac{1}{F} \tag{11}$$

其中, $\hat{f}_1$ 表示以货物出入库效率最高时的单目标函数的最优值, $\hat{f}_2$ 表示以货架稳定性最优时的单目标函数最优值, w_1 和 w_2 分别表示决策者对 2 个优化目标赋予权重。

2 自适应遗传算法

2.1 遗传算法

遗传算法 (genetic algorithm, GA) 由 Holland 教授提出后得到长足发展,尤其在处理组合优化问题上更是被广泛运用。然而,该算法也存在着诸如进化初期易“早熟”、进化末期难收敛等问题^[18]。因

此,本文采用一种动态自适应策略即自适应遗传算法 (adaptive genetic algorithm, AGA) 对算法的选择、交叉、变异算子进行改进,使得算法在初期具备较强的全局搜索能力,而到后期又增强其局部搜索能力,从而有效解决上述问题。

2.2 AGA 设计

2.2.1 编码

编码是算法设计的关键一步,编码方式多种多样,需要针对求解问题及数学模型的特征设计相应高效的编码方式。本文此处采用矩阵式编码,即单个个体的基因由一个 $i_{\max} \times a$ 的“伪二进制”矩阵表示。其中,第 i 行代表第 i 个货物被分配的货位,而列长 a 则由仓储货区数 k 、货架的最大排数 $x_{\max}$ 、货架的最大列数 Y 以及货架的最大层数 $z_{\max}$ 共同决定。例如有 5 个货物需要存放,且仓储参数 $k=4$ 、 $x_{\max}=9$ 、 $Y=13$ 、 $z_{\max}=4$,则某个体基因编码方式如图 2 所示。

1	1	1	0	0	0	1	1	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0	0
1	0	0	0	1	0	1	0	0	1	0	1

图 2 个体编码矩阵

在个体编码矩阵中,第 1~2 列代表货位的货区数,第 3~6 列代表货位的排数,第 7~10 列代表货位的列数,第 11~12 列代表货位的层数。在上述矩阵中第 1 行代表的货位为 (4,9,13,4),而第 2 行代表的货位为 (1,1,1,1)。

2.2.2 自适应策略

轮盘赌法是常用的选择算子,但是这种方法会产生较大的抽样误差,容易造成“早熟”的现象。为此,本文在选择操作之前采用自适应策略对适应度值进行变换:

$$G_1(j) = G(j) + \frac{e - e^{\nu/T}}{e + e^{\nu/T}} (G_{\max} - G_{\min}) \tag{12}$$

式中 $G_{\max}$ 为当前种群最大适应度值, $G_{\min}$ 为当前种群最小适应值, t 为当前遗传代数, T 为终止遗传代数,再按照概率 $p(j) = \frac{G_1(j)}{\sum_{j=1} G(j)}$ 进行选择操作。

在进化初期 $\frac{e - e^{1/T}}{e + e^{1/T}} \approx 1$ 时, $G_1(j)$ 被选择概率弱化,有利于保持种群的多样性,避免“早熟”;而在进化后期 $\frac{e - e^{1/T}}{e + e^{1/T}} \approx 0$ 时, $G_1(j) \approx G(j)$, $G_1(j)$ 被选择概率增强,有利于加快收敛到最优解。

另外,在交叉与变异算子中也采用自适应策略动态调整参数,使交叉率和变异率随适应度值进行动态变化。如此,能够使算法在初期保持全局搜索能力强而后期又可充分发挥局部搜索能力,加速收敛到最优解。

$$P_c = \begin{cases} P_{cmax} - \frac{(P_{cmax} - P_{cmin})(\hat{G}(j) - G_{avg})}{G_{max} - G_{avg}} & G_{avg} \leq G(j) \\ P_{cmax} & G_{avg} > G(j) \end{cases} \quad (13)$$

$$P_m = \begin{cases} P_{mmax} - \frac{(P_{mmax} - P_{mmin})(G_{max} - G(j))}{G_{max} - G_{avg}} & G_{avg} \leq G(j) \\ P_{mmax} & G_{avg} > G(j) \end{cases} \quad (14)$$

式中, P_{cmax} 和 P_{cmin} 表示交叉率取值的上限与下限, P_{mmax} 和 P_{mmin} 表示变异率取值的上限与下限, G_{avg} 为当前种群平均适应值, $\hat{G}(j)$ 为交叉的 2 个个体中较大的适应度, $G(j)$ 为待变异个体的适应度。

2.3 AGA 算法实现

输入: $i_{max}, m_i, r_i, v_1, v_2, l, h, w_1, w_2$ 。

输出: 最优货位分配方式。

步骤 1 输入货位分配优化模型参数: $i_{max}, m_i, r_i, v_1, v_2, l, h, w_1, w_2$ 。

步骤 2 设置 AGA 参数: $T, J, P_{cmax}, P_{cmin}, P_{mmax}, P_{mmin}$ 。

步骤 3 算法开始 $t=1$, 运用 2.2.1 中矩阵编码方式生成初始种群 J 。

步骤 4 判断进化次数是否超过终止进化代数($t > T$), 若是则转步骤 5, 否则继续。

步骤 4.1 计算个体的目标函数值 F 以及适应度 G ;

步骤 4.2 选择算子, 采用式(12)对适应度值进行

G 变换为 G_1 ;

步骤 4.3 交叉算子, 采用式(13)对交叉率进行自适应调整;

步骤 4.4 变异算子, 采用式(14)对变异率进行自适应调整;

步骤 4.5 进化次数 $t = t + 1$ 。

步骤 5 算法结束, 输出最优货位分配方案。

3 改进的粒子群优化算法

3.1 粒子群优化算法

粒子群优化算法 (particle swarm optimization, PSO) 是一种基于鸟群迁徙行为的群体优化算法, 其迭代寻优是一种由粒子速度与位置共同决定的复杂非线性变化过程^[19]。同时, 速度与位置的更新又与惯性权重 (ω) 和学习因子 (c) 密切相关, 而传统 PSO 中这 2 个参数恒定或线性递减的设定并不能反映粒子这一复杂过程。因此, 本文采用非线性变化动态更新惯性权重以及及时变加速学习因子这 2 种策略对算法进行改进, 设计改进的粒子群优化算法 (modified particle swarm optimization, MPSO)。

3.2 MPSO 设计

货位分配优化问题的解在空间呈离散分布, 此处借鉴离散思想对 MPSO 进行编码^[20], 使粒子位置与货位分配方案进行一一映射。每个货位有 4 个维度, 则搜索空间维度为 $D(D=4 \times i_{max})$, 种群为 $X = (X_1, X_2, \dots, X_n)$, 其中 $N(n \in N)$ 为种群规模, 第 i 个粒子在第 t 代的位置与速度则表示为

$$X_i^t = (x_{i1}, x_{i2}, \dots, x_{iD})^T \quad (15)$$

$$V_i^t = (v_{i1}, v_{i2}, \dots, v_{iD})^T \quad (16)$$

并且在迭代选优过程中采用如下公式更新粒子状态:

$$V_{id}^{t+1} = \omega V_{id}^t + c_1 r_1 (P_{id}^t - X_{id}^t) + c_2 r_2 (P_{gd}^t - X_{id}^t) \quad (17)$$

$$X_{id}^{t+1} = X_{id}^t + V_{id}^{t+1} \quad (18)$$

其中, P_{id}^t 为粒子 i 的个体极值, P_{gd}^t 为种群的群体极值。 r_1, r_2 为介于 $[0, 1]$ 之间的随机数; ω 为惯性权重, 一般设为常数或线性递减函数; c_1, c_2 为学习因子, 一般为非负常数。然而考虑到 PSO 实际寻优是非线性变化

过程,本文根据群体最优解的变化采用非线性函数来更新惯性权重 ω (初始 $\omega^0 = 0.9$),保证算法能够跳出局部最优解而在快接近最优解时又能加快收敛到全局最优解,具体更新过程如下所示。

$$\omega^{i+1} = \begin{cases} \omega^i, P_g^i \leq P_g^{i+1} \\ \omega^i \times rand \times (1 - t/T)^2, P_g^i > P_g^{i+1} \end{cases} \quad (19)$$

由 PSO 理论可知, c_1 反映的是粒子自身学习能力,而 c_2 为群体学习能力,如果在迭代初期设置 $c_1 > c_2$, 粒子则能够趋向种群最优;而在迭代末期 $c_1 < c_2$, 则将有利于粒子收敛于全局最优解。因此,本文对于 c_1 、 c_2 的改进如下所示:

$$c_1 = c_{1i} + (c_{1f} - c_{1i}) \times \frac{t}{T} \quad (20)$$

$$c_2 = c_{2i} + (c_{2f} - c_{2i}) \times \frac{t}{T} \quad (21)$$

其中, c_{1i} 、 c_{2i} 、 c_{1f} 、 c_{2f} 为初始设置的常数, t 为当前迭代次数, T 为最大迭代次数。研究表明^[21], 当 c_1 从 2.5 变化到 0.5、 c_2 从 0.5 变化到 2.5 时,最有利于搜索到最优解,因此,本文设置 $c_{1i} = c_{2f} = 2.5$, $c_{2i} = c_{1f} = 0.5$ 。

3.3 MPSO 算法实现

输入: $i_{\max}, m_i, r_i, v_1, v_2, l, h, w_1, w_2$ 。

输出: 最优货位分配方式。

步骤 1 输入货位分配优化模型参数: $i_{\max}, m_i, r_i, v_1, v_2, l, h, w_1, w_2$ 。

步骤 2 设置 MPSO 参数: $T, N, c_{1i}, c_{2i}, c_{1f}, c_{2f}, \omega^0$ 。

步骤 3 算法开始 $t = 1$, 随机生产初始粒子群 N 的位置及速度。

步骤 4 判断迭代次数是否超过最大迭代次数 ($t > T?$), 若是则转步骤 5, 否则继续。

步骤 4.1 计算个体的目标函数值 F 以及适应度 G ;

步骤 4.2 更新粒子的个体最优 P_i^t 以及群体最优 P_g^t ;

步骤 4.3 根据式 (19) ~ (21) 更新惯性权重 ω 与学习因子 c_1, c_2 ;

步骤 4.4 根据式 (17) 与式 (18) 更新各个粒子的位置与速度;

步骤 4.5 迭代次数 $t = t + 1$ 。

步骤 5 算法结束, 输出最优货位分配方案。

4 仿真实验

4.1 实例背景

某汽车零部件制造企业的非传统布局仓储中心货架按如图 1 的 Fishbone 型摆放。其中, 货位的长和宽均为 $l = 1$ m, 货位的高为 $h = 1$ m, 第一排货架的货位数为 $x_{\max} = 9$, AGV 的水平速度 $v_1 = 1$ m/s, 竖直速度 $v_2 = 0.5$ m/s, 决策者偏好权重 $w_1 = w_2 = 1$ 。货物信息如附表 1 所示 (50 类物品共占 203 个货位), 这些货物按照人工方式进行存放得到的初始分配方案如附表 2 所示。由于算法参数的设置对最终优化结果优劣程度有较大影响, 为保证仿真结果的客观性, 根据初始实验本文设置 AGA 与 MPSO 的参数信息如表 1 所示。设计采用 Matlab2015b 编译, 运行环境 Intel (R) Xeon (R) 2.10 GHz 16 GB, 操作系统 Windows 7。

表 1 AGA 与 MPSO 参数值

AGA 参数	MPSO 参数
$T = 1000$	$T = 1000$
$J = 500$	$N = 500$
$P_{c\max} = 0.9$	$c_{1i} = 2.5$
$P_{c\min} = 0.4$	$c_{2i} = 0.5$
$P_{m\max} = 0.1$	$c_{1f} = 0.5$
$P_{m\min} = 0.01$	$c_{2f} = 2.5$

4.2 仿真结果

附表 3、4 分别是由 AGA 和 MPSO 得到货位分配优化方案。表 2 为 3 种不同方法得到货位分配方案的子目标函数值 f_1 、 f_2 以及归一化后子目标加权求和值 F 。数值越小表示货位分配方案的效果越优, 由表 2 可

表 2 3 种货位方案目标函数值

目标函数	初始方案	AGA 优化后方案	MPSO 优化后方案
f_1	118.01	90.54	92.85
f_2	16144	8950	11744
F	—	174.01	184.76

注: 初始方案无法求得其单目标最优值故无法进行归一化处理, 因此其 F 值为空缺。

知 AGA 对上述 2 个子目标函数值均有优化,优化率分别为 23.28% 与 44.56%;而 MPSO 的优化效果稍微弱一些,优化率为 21.32% 与 27.25%。由此说明本文提出的 2 种方法对于提升非传统布局仓储出入库效率以及货架稳定性两方面均有效可行,并且优化效果明显。

在相同环境下运行,AGA 与 MPSO 2 种算法总运行耗时分别为 412 s 与 527 s,其中图 3、图 4 分别给出了 2 种算法的收敛过程,AGA 在第 604 代开始收敛于最优值,而 MPSO 则于 627 代后开始收敛。AGA 整体的收敛速度比 MPSO 更快一些,并且再结合上述优化结果则表明在求解该问题时,AGA 的整体优化性能比 MPSO 更强一些。

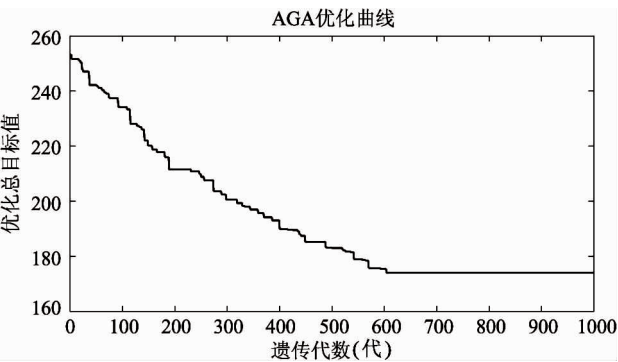


图 3 AGA 优化总目标值变化曲线

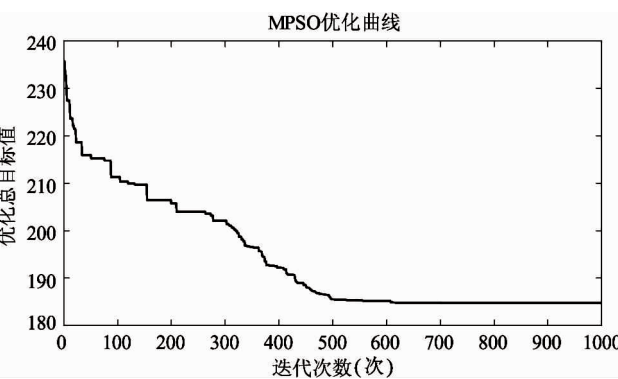


图 4 MPSO 优化总目标值曲线

4.3 对比分析

为了进一步验证本文提出的 2 种算法通用性以及对比算法的优化性能,本文在 4 组不同规模的货位分配优化问题下,对 4 种算法基本 GA (simple genetic algorithm, SGA)、AGA、基本 PSO (simple particle swarm optimization, SPSO)、MPSO 再次进行仿真实验,其中,SGA 与 SPSO 为无改进的基本算法,其他参数设置与 AGA 和 MPSO 设置相同。由于归一化加权后的总目标函数 F 值并不能直接反映优化效果,因此本文选取更为直观的出入库效率目标函数值 f_1 和货架稳定性目标函数值 f_2 以及收敛代数 G_{conve} 等 3 个指标,在相同环境下分别运行 10 次,计算上述 3 个指标的均值,得到如表 3 所示结果。

根据表 3,从收敛代数来看,在问题规模较小情况下,基本算法与本文提出的改进算法收敛代数也大致相同,同样 AGA 与 MPSO 的收敛代数也相当,这是因为在问题规模不大时,最优解相对来说比较容易求得。但随着问题规模的扩大,本文设计的 2 种改进算法相比基本算法在求解效率方面的优势开始显现,这一结果也验证了改进策略可提升算法末期的局部寻优能力,加速收敛。而 AGA 和 MPSO 单独对比分析可知,随着问题规模扩大,AGA 收敛速度波动较小,算法鲁棒性更强。

结合表 4,在优化效果方面,相比基本算法,在不同规模问题下 AGA 与 MPSO 的优化效果均高于相应的基本算法,说明改进策略也提升了算法的全局寻优能力,得到的货位分配方案改善了出入库效率,增强了货架的稳定性,而从 AGA 与 MPSO 二者单独的对比则再次验证了 AGA 整体优化性能比 MPSO 更强。当然,随着问题规模的扩大,所有算法的优化效果值都在下降,这是因为随规模扩大初始方案的目标函数值呈指数级扩大,因此,优化效果下降也是合理的。

表 3 4 种算法对不同规模问题的求解结果

货品数 货位数	SGA			AGA			SPSO			MPSO		
	f_1	f_2	G_{conve}	f_1	f_2	G_{conve}	f_1	f_2	G_{conve}	f_1	f_2	G_{conve}
$I=25, J=120$	99.60	4892.2	518.9	99.51	4892.2	494.3	112.34	5246	484.6	102.43	5204.2	476.2
$I=50, J=203$	93.44	9388.9	623.4	91.35	9373.4	593.6	95.44	12003.5	648.2	93.73	11892.4	622.6
$I=75, J=547$	212.43	38177.4	719.7	208.99	33885.5	636.7	220.33	38165.8	685.9	224.84	35128.2	688.1
$I=100, J=810$	247.56	61345.2	782.4	232.50	57158.7	711.1	241.09	62108.3	879.3	239.37	59009.9	826.3

表 4 4 种算法对不同规模问题的优化效果

货品数/货位数	SGA		AGA		SPSO		MPSO	
	$E_{opt1}(\%)$	$E_{opt2}(\%)$	$E_{opt1}(\%)$	$E_{opt2}(\%)$	$E_{opt1}(\%)$	$E_{opt2}(\%)$	$E_{opt1}(\%)$	$E_{opt2}(\%)$
$I=25, J=120$	31.27	47.25	31.33	47.25	22.48	43.43	29.32	43.88
$I=50, J=203$	20.82	41.84	22.59	41.94	19.13	25.65	20.57	26.34
$I=75, J=547$	17.03	5.83	18.37	16.42	13.94	5.86	12.18	13.35
$I=100, J=810$	6.42	6.71	12.11	13.08	8.87	5.55	9.52	10.26

5 结 论

本文以实际问题为背景,分析了非传统布局仓储及货位分配优化问题的特征,构建了以货物出入库效率最高和货架稳定性最优为目标的多目标货位分配优化模型。然后,针对优化模型分别设计了适用于解决此类问题的自适应遗传算法(AGA)和改进的粒子群优化算法(MPSO),并采用 Mtalab 实现了问题的仿真以及算法求解。结果表明,本文提出的 2 种方法均能更有效地提高货物出入库的效率,降低货物存放重心,提高货架的稳定性。此外,4 种算法性能对比的结果则说明本文设计的改进算法相比基本算法对于优化货位分配效果更加明显,同时也表明 AGA 整体优化性能比 MP-SO 更优,鲁棒性更强。但如何提高算法求解大规模以及超大规模问题的求解速度与优化精度仍然有待进一步研究。

参考文献

[1] Shah B, Khanzode V. A comprehensive review of warehouse operational issues [J]. *International Journal of Logistics Systems & Management*. 2017,26(3):346-378

[2] Rouwenhorst B, Reuter B, Stockrahm V, et al. Warehouse design and control: framework and literature review [J]. *European Journal of Operational Research*, 2000,122(3):515-533

[3] Gue K R, Meller R D. Aisle configurations for unit-load-warehouses [J]. *Iie Transactions*, 2009,41(3):171-182

[4] Pohl L M, Meller R D, Gue K R. An analysis of dual-command operations in common warehouse designs [J]. *Transportation Research Part E*, 2009,45(3):367-379

[5] Pohl L M, Meller R D, Gue K R. Turnover based storage in non-traditional unit-load warehouse designs [J]. *Iie Transactions*, 2011,43(10):703-720

[6] Öztürkçülu Ö, Gue K R, Meller R D. Optimal unit load warehouse designs for single command operations [J]. *Iie Transactions*, 2012, 44(6):459-475

[7] Martínez H J. Analytical study of the Fishbone Warehouse layout [J]. *International Journal of Logistics Research & Applications*, 2012,15(6):365-388

[8] Clark K A, Meller R D. Incorporating vertical travel into non-traditional cross aisles for unit load warehouse designs [J]. *Iie Transactions*, 2013,45(12):1322-1331

[9] Rardona L F, Soto D F, Rivera L, et al. Detailed design of fishbone warehouse layouts with vertical travel [J]. *International Journal of Production Economics*, 2015(4),170:825-837

[10] 周丽, 郭健, 朱杰. 配送中心存储区域的鱼骨布局设计与分析 [J]. *管理世界*, 2014(5):184-185

[11] 蒋美仙, 冯定忠, 赵晏林, 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):2920-2929

[12] 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(3):201-204

[13] Fontana M E, Nepomuceno V S. Multi criteria approach for products classification and their storage location assignment [J]. *International Journal of Advanced Manufacturing Technology*, 2016,88(9-12):1-12

[14] 张贵军, 姚俊, 周晓根, 等. 基于精英多策略的货位分配优化方法 [J]. *计算机科学*, 2018,45(1):273-279

[15] 李鹏飞, 马航. 基于病毒协同遗传算法的自动化立体仓库货位优化模型 [J]. *中国管理科学*, 2017,25(5):70-77

[16] 陈月婷, 何芳. 基于改进粒子群算法的立体仓库货位分配优化 [J]. *计算机工程与应用*, 2008,44(11):229-231

[17] 刘超, 贺利军, 朱光宇. 基于熵和隶属度函数的高维多目标优化问题求解 [J]. *计算机工程*, 2016,42(6):185-190

[18] 邹进. 自适应逐次逼近遗传算法及其在水库群长期调度中的应用 [J]. *系统工程理论与实践*, 2013,(1):267-272

[19] 黄勤, 赵海茹, 陈玲. 基于改进 PSO 的组合预测模型研究 [J]. *计算机工程与应用*, 2015,51(14):258-263

[20] 田军, 马文正, 汪应洛, 等. 应急物资配送动态调度的粒子群算法 [J]. *系统工程理论与实践*, 2011,31(5):898-906

[21] Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3):240-255

Storage location assignment optimization in non-traditional warehouse base on AGA and MPSO

Hu Yingcong^{*}, Liu Jiansheng^{*}, Zhang Yougong^{**}

(^{*} School of Economic and Management, Nanchang University, Nanchang 330031)

(^{**} School of Mechanical and Electrical Engineering, Nanchang University, Nanchang 330031)

Abstract

Non-traditional layout warehouse is a new hot spot in the current storage industry. In view of the storage location assignment optimization with Fishbone layout, taking inventory efficiency and shelf stability as the optimization objective based on its characters, an optimization model for the storage location assignment is formulated. Then, the adaptive genetic algorithm (AGA) and modified particle swarm optimization (MPSO) are developed to solve the problem. A dynamic adaptive strategy is exploited in AGA to improve selection, crossover and mutation operators, which can overcome premature, improve the capability of local search ,and enhance the robustness. Considering the complex nonlinear characteristics of PSO searching process, the inertial weights and the time-varying accelerated learning factors are introduced to improve the early global search ability, the late convergence and optimize the overall performance. These algorithms are compiled by Matlab, and the effectiveness and versatility are verified through a case. The experiment results show that AGA is superior to other algorithms in dealing with storage location assignment optimization.

Key words: non-traditional layout warehouse, storage-location assignment, adaptive genetic algorithm (AGA), modified particle swarm optimization (MPSO)

附表 1 货物信息

货品编号	货品质量(kg)	存取频率(%)	所占货位(个)	货品编号	货品质量(kg)	存取频率(%)	所占货位(个)
1	53	3	6	26	17	3	6
2	42	1	5	27	58	1	6
3	28	3	6	28	58	2	2
4	36	2	3	29	39	1	4
5	31	3	4	30	13	2	2
6	14	2	1	31	22	1	4
7	22	3	4	32	28	3	6
8	17	3	5	33	52	3	4
9	20	2	5	34	11	1	5
10	22	3	4	35	13	1	4
11	31	3	3	36	19	3	4
12	13	2	4	37	43	3	3
13	56	2	3	38	47	1	2
14	58	2	4	39	43	1	6
15	35	2	2	40	33	1	5
16	35	2	2	41	38	2	5
17	27	1	3	42	25	1	4
18	56	1	5	43	48	3	2
19	29	1	6	44	20	1	4
20	16	2	6	45	45	1	6
21	50	1	1	46	20	2	4
22	30	2	5	47	29	3	2
23	23	3	2	48	42	3	3
24	31	3	6	49	50	3	6
25	15	1	5	50	15	1	4

附表 2 初始货位分配方案										
编号	1	2	3	4	5	6	7	8	9	10
货位	1,4,7,2	1,1,3,4	3,1,2,4	4,5,5,4	3,3,5,3	4,3,8,2	2,1,1,1	1,3,4,4	1,3,4,2	4,1,12,1
	3,5,5,2	1,2,11,2	3,5,4,2	1,4,5,3	2,1,2,4		4,4,2,3	3,3,2,2	2,4,5,2	4,6,3,4
	1,1,4,1	2,7,1,1	3,2,6,4	2,1,10,3	3,5,7,2		4,6,5,4	1,3,10,2	4,3,1,4	3,6,1,2
	3,7,2,2	3,6,3,2	1,1,8,1		4,4,7,1		4,5,7,1	2,7,2,2	2,6,4,3	3,3,6,3
	1,6,4,3	4,3,3,3	3,2,3,2					1,3,6,3	2,1,6,1	
	3,2,1,2		3,3,2,3							
编号	11	12	13	14	15	16	17	18	19	20
货位	1,4,6,2	3,6,4,1	2,2,4,1	2,5,1,2	4,3,1,3	3,1,10,1	2,4,3,2	4,1,8,4	2,3,6,3	4,3,4,1
	2,2,11,4	2,3,6,1	2,3,8,1	4,3,2,1	1,5,2,3	2,3,2,2	1,1,8,2	4,3,7,2	1,2,8,2	4,2,1,3
	1,4,1,1	3,7,3,4	1,3,3,3	2,3,8,4			1,4,8,4	3,3,1,2	2,6,1,4	4,2,7,3
		4,2,6,2		4,5,5,2				3,4,7,3	4,1,6,2	1,2,7,3
								1,4,7,4	2,2,4,4	3,6,4,2
									1,1,2,2	3,2,6,3
编号	21	22	23	24	25	26	27	28	29	30
货位	1,6,5,3	4,5,2,4	4,6,4,3	1,5,6,3	2,2,4,3	3,4,6,3	2,1,9,2	2,1,7,4	3,1,3,2	3,3,4,1
		2,2,3,1	2,8,1,2	2,2,5,2	4,1,11,4	2,1,11,2	1,2,10,2	3,7,1,4	3,1,6,3	2,2,9,4
		1,1,8,4		2,3,7,1	1,2,8,1	1,5,6,4	4,6,3,3		3,5,3,1	
		1,3,6,4		3,3,2,4	4,4,5,4	1,4,5,1	4,1,7,2		3,2,2,2	
		1,8,1,2		4,4,6,2	4,2,3,1	4,4,3,2	3,3,9,3			
				1,2,10,4		4,2,10,1	4,7,4,2			
编号	31	32	33	34	35	36	37	38	39	40
货位	4,1,9,3	3,4,5,1	2,4,6,1	3,2,4,3	4,2,1,1	1,1,12,4	4,6,2,2	1,2,5,1	2,6,3,3	1,6,3,2
	2,7,4,3	4,1,12,3	1,5,3,1	3,1,8,3	4,7,4,4	2,4,1,4	2,8,2,1	2,2,3,3	2,4,7,3	3,7,4,3
	1,1,6,2	2,2,10,3	3,1,4,1	2,7,2,3	2,3,9,2	4,3,1,1	3,3,10,1		1,3,8,4	2,1,11,3
	1,2,7,2	2,3,4,2	2,2,8,3	1,1,4,2	3,5,6,4	2,4,2,1			2,1,6,2	2,1,2,1
		1,1,7,1		3,2,11,1					2,4,3,3	4,1,4,3
		2,3,9,3							1,3,3,1	
编号	41	42	43	44	45	46	47	48	49	50
货位	2,2,1,4	2,9,1,4	4,1,2,2	4,3,9,2	3,6,2,1	3,4,4,2	4,2,4,4	2,3,1,2	3,1,12,2	3,4,6,4
	4,6,2,3	3,2,10,4	1,1,9,2	1,1,2,4	2,2,9,3	1,4,1,2	2,5,4,4	2,5,5,1	4,7,1,2	1,5,3,3
	2,4,8,2	3,2,1,4		2,1,5,3	1,4,3,1	3,1,6,4		1,1,5,3	4,2,8,2	4,3,4,2
	1,3,5,4	1,6,2,4		3,3,5,1	1,5,4,2	1,4,2,4			4,5,7,2	4,6,1,3
	3,1,9,4				3,5,7,4				2,2,5,1	
					2,1,5,4				4,1,10,4	

附表3 AGA 优化后货位分配方案

编号	1	2	3	4	5	6	7	8	9	10
货位	2,4,2,1	4,1,13,1	4,3,8,1	2,6,3,2	4,2,4,1	2,7,1,1	2,2,4,2	4,2,1,2	2,5,7,1	2,2,7,3
	4,3,5,1	4,2,3,3	2,7,1,1	1,4,2,1	1,1,5,1		3,1,13,2	4,2,2,3	3,3,3,2	4,1,9,1
	4,4,2,1	2,6,4,1	1,4,7,1	3,4,7,3	3,2,6,2		1,7,4,1	1,3,8,1	2,1,3,1	3,1,10,4
	2,3,7,1	3,4,4,1	4,4,4,1		4,5,2,1		4,1,6,1	2,8,1,1	2,2,1,1	4,2,1,1
	2,2,11,2	1,1,3,4	1,1,1,2					3,7,1,1	2,5,1,1	
	3,6,3,1		3,2,4,1							
编号	11	12	13	14	15	16	17	18	19	20
货位	1,1,1,1	2,5,1,1	1,2,6,1	1,2,3,1	3,3,2,2	1,1,10,1	2,1,4,1	3,4,2,1	4,4,5,2	3,4,3,1
	4,1,1,1	4,2,9,1	3,5,2,1	3,2,7,1	1,5,3,1	2,2,2,1	2,5,5,1	3,8,2,1	2,5,2,1	4,2,9,3
	3,2,10,1	3,2,4,1	1,2,1,1	3,1,5,1			2,2,5,2	3,4,4,1	3,1,11,2	3,3,3,1
		2,7,3,1		3,9,1,1				4,3,3,1	1,2,4,2	1,6,3,1
								4,1,1,1	3,3,2,4	1,6,4,1
									3,4,2,1	1,5,5,1
编号	21	22	23	24	25	26	27	28	29	30
货位	2,4,1,1	1,1,2,2	1,1,9,1	3,1,6,2	2,1,4,3	1,3,6,3	2,5,3,1	3,4,7,1	4,3,9,1	3,1,6,3
		3,4,3,1	1,2,11,2	2,1,1,4	4,6,1,1	4,3,9,2	1,4,7,1	3,4,8,2	3,1,9,1	4,4,5,1
		2,1,1,1		3,1,4,1	1,3,4,1	3,3,4,1	3,4,7,1		4,2,6,1	
		1,8,1,2		1,3,8,1	3,1,12,2	1,4,2,2	2,2,10,1		1,3,9,1	
		2,2,9,2		4,5,5,1	1,3,1,2	2,3,7,1	2,2,7,1			
				1,1,2,1		3,5,5,1	2,5,4,1			
编号	31	32	33	34	35	36	37	38	39	40
货位	4,2,2,2	3,1,10,1	3,3,2,1	2,4,6,3	2,1,8,1	2,3,3,1	1,1,3,2	3,5,2,1	2,6,2,1	3,1,3,1
	1,2,10,1	3,1,1,1	2,3,1,1	4,3,1,3	2,4,3,1	1,4,2,2	2,1,10,1	4,1,11,1	4,4,2,1	4,2,9,1
	2,4,4,1	3,1,1,2	2,2,7,1	3,6,2,1	2,2,4,1	3,1,5,2	4,4,3,2		3,1,8,1	4,6,4,1
	1,5,1,3	1,5,5,4	2,5,6,1	4,8,1,2	3,4,1,2	1,3,2,1			2,4,7,2	2,6,1,2
		3,2,1,1		1,1,1,3					4,7,1,1	2,1,9,2
		4,3,1,1							1,3,1,2	
编号	41	42	43	44	45	46	47	48	49	50
货位	3,6,4,1	3,3,3,1	4,1,3,1	1,2,3,2	1,1,11,1	3,6,1,1	4,1,5,1	2,2,4,2	1,1,3,2	4,1,2,2
	1,1,11,1	2,3,8,1	4,4,1,1	1,1,10,3	4,1,8,1	3,2,1,1	3,1,9,1	3,3,3,1	2,3,6,1	2,1,3,1
	2,4,5,1	1,5,2,2		3,2,6,4	4,6,1,1	2,4,4,3		4,2,6,2	1,3,3,1	4,3,7,2
	4,3,2,1	1,3,5,1		4,5,1,2	2,1,6,1	3,1,3,1			2,2,7,1	3,7,2,1
	2,3,4,2				1,1,1,1				1,2,2,2	
					3,4,1,1				2,2,7,2	

附表 4 MPSO 优化后货位分配方案										
编号	1	2	3	4	5	6	7	8	9	10
货位	1,1,1,1	4,2,7,2	3,5,2,1	2,2,1,1	4,7,2,1	4,2,2,3	1,1,2,2	3,1,10,2	3,1,3,4	2,3,1,1
	1,5,1,1	3,1,2,1	4,2,1,1	2,4,2,1	3,6,1,1		2,1,6,1	2,1,6,2	4,4,4,3	2,1,1,1
	1,2,10,1	2,4,6,1	4,1,1,2	1,5,6,2	3,1,3,1		1,3,1,1	2,1,8,1	1,1,4,2	3,2,3,1
	3,4,2,1	4,1,1,1	1,2,4,2		4,5,6,4		4,2,6,1	3,2,2,1	1,1,5,1	4,3,5,2
	1,1,3,1	4,1,8,1	2,1,2,1					1,1,6,1	2,2,3,3	
	3,2,1,1		3,1,4,4							
编号	11	12	13	14	15	16	17	18	19	20
货位	1,2,3,1	1,2,9,3	2,1,4,1	4,3,3,1	3,4,5,1	4,1,11,1	3,3,7,1	2,6,5,2	3,1,5,2	2,1,1,2
	1,7,2,4	3,2,1,2	1,3,5,1	4,2,3,1	3,1,1,1	4,1,6,2	1,6,2,3	3,6,4,1	1,3,5,3	4,2,2,1
	1,2,1,3	3,6,3,3	4,8,1,3	3,1,5,1			2,2,2,1	3,3,3,1	2,6,2,1	3,2,6,2
		4,1,4,2		1,7,3,1				2,7,4,2	3,3,2,4	2,4,2,2
								3,5,6,1	1,4,5,2	4,6,5,1
									2,1,3,1	1,4,4,1
编号	21	22	23	24	25	26	27	28	29	30
货位	1,4,4,2	3,1,4,1	3,5,3,2	4,4,2,3	3,3,6,4	2,5,1,1	3,4,8,2	4,2,7,1	4,3,4,1	4,2,3,3
		2,4,2,3	4,7,2,2	4,4,2,1	1,6,3,2	1,4,4,3	3,1,7,3	2,5,5,1	4,2,4,1	3,3,8,2
		2,3,3,1		3,6,1,2	4,6,3,3	3,4,3,1	2,1,4,3		4,5,4,1	
		3,2,4,1		2,2,8,1	2,5,1,3	4,1,8,3	1,5,2,2		3,3,3,3	
		1,2,3,2		1,1,4,1	3,2,11,3	3,4,1,1	1,2,4,3			
				3,2,5,2		2,3,3,2	1,2,1,1			
编号	31	32	33	34	35	36	37	38	39	40
货位	3,3,6,1	4,2,8,2	1,2,7,1	3,2,5,3	3,4,3,3	4,4,4,2	2,2,5,1	4,2,6,3	3,4,5,2	1,3,3,1
	3,3,4,1	2,3,4,1	2,2,9,2	4,6,2,2	2,4,7,1	2,2,6,1	2,3,2,2	1,1,9,4	1,7,2,1	3,1,8,2
	2,2,4,1	4,6,3,2	2,6,4,1	1,4,3,2	3,3,4,4	1,4,2,1	3,3,2,1		1,3,9,2	3,2,3,2
	1,3,2,2	3,4,7,1	2,3,7,1	2,2,4,4	2,2,6,4	3,1,3,2			3,2,8,1	4,4,6,1
		2,3,2,1		3,3,6,3					1,2,6,2	3,3,3,2
		2,3,8,3							3,4,7,3	
编号	41	42	43	44	45	46	47	48	49	50
货位	2,2,1,2	2,5,4,3	4,4,3,1	3,1,9,1	1,3,7,2	2,3,2,3	2,2,4,2	2,2,7,2	2,1,5,3	2,4,7,3
	3,5,5,3	3,5,1,1	2,1,10,2	2,1,5,2	3,5,3,3	3,5,4,1	2,3,5,2	3,2,3,3	1,6,1,2	2,5,5,2
	2,5,3,1	3,4,4,1		2,4,5,1	3,3,2,3	3,5,4,3		2,6,3,2	4,4,6,3	3,4,2,2
	3,7,1,4	3,2,4,2		4,5,4,2	3,2,6,3	2,3,4,4			3,3,6,2	3,2,7,1
	2,2,10,3				2,4,6,2				2,2,2,2	
					1,2,8,2				2,4,4,2	