

## 智能存储过程优化设计

### 一．问题阐述

某智能仓库，有两个巷道，4排货架，中间两排货架并排在一起，每排货架有200货格，其中长20个，高10个，每个货格尺寸为300cm\*300cm, 堆垛机行进速度与升降为0~1.6米/秒，行进过程可以升降，堆垛机存取货物时只有一个出口，智能仓库存放10类物品，这10类物品需求量不同，假设不同需求，设计这10类物品如何存放，占多少货格，取货速度能达到最优。

### 二．货位优化方案

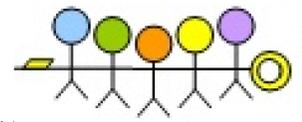
自动化立体仓库的高效率是建立在高效的货位管理基础之上的。对自动化立体仓库的货位进行管理就是要合理地分配和使用货位，既考虑如何提高货位的利用率，又要保证出库效率。

#### 2.1 存储管理的存储策略

良好的存储策略可以减少出入库移动的距离、缩短作业时间，甚至能够充分利用存储空间。一般自动化立体仓库常见的存储策略有：定位存放、随机存放、分类存放、分类随机存放与共享存放。

##### 1. 定位存放

在这种存放策略下，每个存储货物都有固定的存储位置，货物之间不能存储。因此，需要规定每项货位的存储容量必须大于其可能的最大在库量。定位存放的优点是货物的存储位置可以按照周转率大小或出货率来安排，以缩短出入库的搬运距离。可针对各种货物的特性做储位的安排调整，将不同货品特性之间的相互影响降到最小。缺点



是储位必须按各项货品的最大在库量来设计。 总之,定位存放易管理、需时少,但浪费空间。所以,该方案适用于厂房空间大或者多种少量商品的存放。

## 2. 随机存放

这种存储方式,每个货品被指派存放的为止都是经由随机过程产生,而且可以随机改变。即任何物品可以被存放在任何可以利用的位置。优点是由于储位可共享,因此只需要按所有库存货品最大在库量设计即可,存储空间的使用效率高。缺点是货品的出入库管理以及盘点工作进行会比较困难。

## 3. 分类存放

所有的储放货品按照一定的特性加以分类,每一类货品都有固定存放的位置,而同属一类的不同货品又可以按一定的法则来分配货位。分类存储通常按产品相关性、流动性、产品尺寸、重量或产品特性来分类。优点:便于畅销品的存取,具有定位储放的各项优点。各分类的储存区域可以根据货品特性再作设计,有助于货品的储放管理。缺点:储位必须按各项货品最大在库量设计,一次存储空间平均使用率较低。该类存储策略适用于产品相关性大、周转率差别大、产品尺寸相差大的情况。

## 4. 分类随机储放

每一类货品都有固定的存放位置,但在各类的储区内,每个货物的分配是随机的。分类随机储放优点:具有分类储放的部分优点,又可以节省储位数量,提高储区利用率。缺点:货品出入库管理以及盘

点工作的难度较高。分类随机储放兼具分类储放及随机储放的特点，需要的存储空间介于两者之间。

## 5. 共享储放

在确定知道各货品的进出库时刻，不同的货品可共享相同储位的方式称为共享储放。共享储放在管理上虽然较复杂，所需的存储空间及搬运时间却更为经济。本文在研究自动化立体仓库的货位优化时，采用了分类储存策略。所有的储存货品按照产品相关性、流动性、产品尺寸、重量或产品特性加以分类，每类货品都有固定存放的位置，而同属一类的不同货品又按一定的法则来分配货位。

### 2.2 货位分配原则

通常需要将适当的存储策略配合货位分配原则来确定货位的初始位置。对自动化立体仓库的货位进行管理，是要合理的分配和使用货位。即考虑如何提高货位的利用率，又要保证出库效率。货位分配包含两方面的含义，一是为出入基于遗传算法的自动化立体仓库的货位优化研究库的物料分配最佳货位，即入库货位分配；二是要选择待出库物料的货位，即出库货位分配。

货位分配考虑的原则有很多，针对于自动化立体仓库，其货位分配原则有以下几种：

(1) 货架承重均匀，上轻下重。分散存放，物料分散存放在仓库的不同位置，避免因集中存放，而使得货架受力不均匀。较重的物品存放在下面的货位，较轻的物品存放在高处的货位，使货架承载稳定；

(2) 加快周转，先入先出。同种物料出库时，先入先出，以加快

物料周转，避免因物料长期积压产生锈蚀、变形、变质及其他损坏造成的损失；

(3) 提高可靠性，分巷道存放。自动化立体仓库有多个巷道时，同种物品分散在不同的巷道进行存放，以防止因某巷道堵塞影响某种物料的出库，造成生产中断；

(4) 提高效率，就近入/出库。在线自动仓库，为保证快速响应出库请求，一般将物料就近放置在出库台附近；

(5) 产品相关性。商品相关性大者在订购时经常被同时订购，所以应尽可能存放在相邻位置。

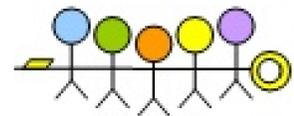
### 三. 建立优化数学模型

在数学模型建立的开始，我们需要对变量进行定义。假设， $m$ 代表托盘货物的基于遗传算法的自动化立体仓库的货位优化研究重量； $f$ 代表货物的存取频率； $n$ 代表货位上货品的数量； $v_x$ 代表传送带的运算速度； $v_y$ 代表堆垛机的纵向运行速度； $v_z$ 代表堆垛机的垂直上升速度(注： $v_x : v_y : v_z = 1:5:1$ )

#### (1) 模型建立

货位分配的策略有多种，针对本文的货位优化研究我们只考虑两个原则“货架承重均匀，上轻下重”与“加快周转，先入先出”。在模型的建立过程中需要同时考虑这两个原则，即在使得仓库出入库效率最高的前提下同时要满足货架的承载均匀条件。可见，模型是一个多目标优化模型，需要同时满足两个目标函数。

得货架的重心最小——“货架承重均匀，上轻下重”



$$\min f_1 = \sum_{i=1}^p \sum_{j=1}^q (G_{ij} * n_{ij} * j)$$

$G_{max}$ 为货架上每个货位所能承受的最大重量，其中  $G < G_{max}$

$G$  为第  $i$  列第  $j$  层货位单位上货物的重量，

$n_{ij}$  为第  $i$  列第  $j$  层货位单位上货物的数量。

目标函数的意义：为了使货架的承载能力均匀，不至于出现倾覆性倒塌，需要遵循“货架承重均匀，上轻下重”的原则。货架的稳定性可以等价为使得货架的重心最低，也就是说使得每排货架上所放货物的总的重心最低。

货物的配送量最小——“提高效率，就近入出库

$$\min f_2 = \sum_{k=1}^l \sum_{i=1}^p \sum_{j=1}^q (t_{kij} / f_{kij})$$

$$t_{ijk} = \frac{iL_0}{v_x} + \frac{jL_0}{v_y} + \frac{kL_0}{v_z} = (i + \frac{j}{5} + k) \frac{L_0}{v_x}$$

$t_{kij}$  为将  $k$  排  $i$  列  $j$  层货位上的物品搬运到巷道口所用时间， $f_{kij}$  为将  $k$  排  $i$  列  $j$  层货位上的单位物品的存取频率。

目标函数的意义：货位优化的目的在于提高生产率，将无用运动减低到最小。根据货品的物流流动性使货物的配送量最小。

(2) 约束条件

$$s.t. \begin{cases} 1 \leq i \leq l \\ 1 \leq j \leq p \\ 1 \leq k \leq q \end{cases} \text{ 且 } x, y, z \text{ 为整数}$$

注：前面假设本文货位优化的区域为  $l$  排  $p$  列  $q$  层的货架区域。将距离巷道口的最近排记为第 1 排，最近的列记为第 1 列，最低层记为

第1层。i 表示货位处于第 i 排；J 表示货位处于第 j 列；k 表示货位处于第 k 层。本题中  $l=2, p=20, q=10$ ，即研究 2 排 20 列 10 层的仓库存储优化问题。为研究方便，设  $L=H=4$ ，即货格长和高分别为 4，且  $v_x=v_z$ ，即堆垛机的走行速度和升降速度相同。而且本题是一个多目标优化问题的求解，而求解难度较大，故将多目标问题的求解转化为大把目标的求解，目前，将多目标优化问题转换为单目标或一系列单目标优化问题的方法有效用函数法、权重和法、妥协法等。本文采用权重法进行解答。

现有六名专家对上述两个目标函数权重进行打分，得到如下表

	专家 1	专家二	专家三	专家四	专家五	专家六	加权
	0.12	0.18	0.2	0.16	0.12	0.22	
1	1	0	1	1	1	0	0.6
2	0	1	0	0	0	1	0.4

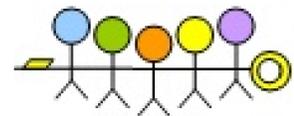
那么货位分配的目标函数可表示为

$$\min f(x) = 0.6 * f_1(x) + 0.4 * f_2(x), \text{ 其中 } X \text{ 表示货品在货架上的位置}$$

考虑最小化问题。由于后悔值越小个体越好，因此需要将后悔值转换为适应值从而确保优秀个体具有较大的适应值。设  $r(x)$  表示个体  $x$  的后悔值， $r_{\max}$  表示当前代中的最大后悔值， $r_{\min}$  表示当前代中的最小后悔值。变换方式如下

$$\text{eval}(x) = \frac{r_{\max} - r(x) + \gamma}{r_{\max} - r_{\min} + \gamma}$$

其中， $\gamma$  是正实数，通常被限制在开区间  $(0, 1)$  中。该系数有两个作用，一个是为了避免式 (4.14) 产生被零除错误；另一个是可以



将选择方式从适应值比例选择调整到纯粹随机选择。

## 四．算法设计与实现

### 4.1 确定染色体编码方法

本文所研究的货位优化问题除了是多目标优化问题外，还应该是一种组合优化问题，它需要组合两个约束目标来寻找满足约束项目的最佳组合，所以本文决定采用整数排列编码方式对染色体进行编码。采用整数排列编码方式，我们将货物与货位的从属关系进行编码如下。

(1) 整个一条染色体代表一种货位分配方案

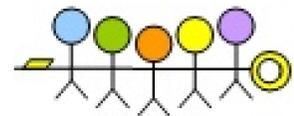
(2) 染色体中基因的个数表示在优化区域内货物的个数（将货位所存放的货物打包视为一个货物），将每条染色体分为  $m$  段， $m$  就表示在该优化区域内货物的总数。

(3) 基因的值表示货物存储的货位，基因的位置表示存放在该货位上的货物的编号。每段基因用整数表示要存储的货位坐标（排，列，行）。假设在 2 排 4 列 3 层的优化区域内共有 5 个物品，其中 1 号货物存放在坐标 (1, 1, 2) 的货位上；2 号货物存放在坐标 (1, 2, 1) 的货位上；3 号货物存放在坐标 (2, 3, 2) 的货物上；4 号货物存放在坐标 (2, 4, 3) 的货位上；5 号货物存放在坐标 (1, 3, 3) 的货位上。那么染色体 “112121 232 243 133” 就形成了一个解，表示一种货位分配方案。如图下表所示

112	121	232	243	133	,
-----	-----	-----	-----	-----	---

### 4.2 适应度函数的确定

本文研究的是目标函数的最小化问题，孤儿后悔值函数只要去目标函数的倒数就可以变为适应度函数，并通过适应度大小来评价



个体优劣程度，为防止溢出，我在目标函数整体上加 1，从而得到本文目标函数所对应的后悔值函数如下：

$$\min r(x) = \frac{1}{0.6 f_1(x) + 0.4 f_2(x) + 1}$$

适应度函数如下：

$$\text{eval}(x) = \frac{r_{\max} - r(x) + \gamma}{r_{\max} - r_{\min} + \gamma}$$

其中  $r$  是正实数，通常被限制在开区间  $(0, 1)$  中，在这里我们令  $r=0.08$

### 4.3 选择、交叉和变异

遗传算法是一种群体型操作，以群体中所有个体为对象。选择、交叉和变异是遗传算法的 3 个主要操作算子，由这三个操作算子构成了遗传操作，使得遗传算法具有了其他传统算法没有的特点。

#### (1) 遗传算子

选择运算使用比例选择算子

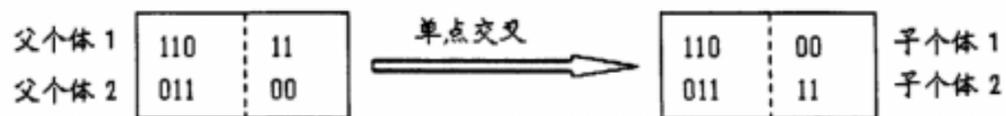
比例选择因子是利用比例于各个个体适应度的概率决定其子孙的遗留可能性。若设种群数为  $M$ ，个体  $f$  的适应度为  $f$ ，则个体  $i$  被选择的概率为

$$p_i = \frac{f_i}{\sum_{k=1}^M f_k}$$

当个体选择的概率给定后，产生  $[0, 1]$  之间的均匀随机数来决定哪个个体参加交配。若个体的选择概率大，则能被多次选中，它的遗传基因就会在种群中扩大；若个体的选择概率小，则被淘汰。

叉运算使用单点交叉算子

只有一个交叉点位置，任意挑选经过选择操作后种群中两个个体作为交叉对象，随机产生一个交叉点位置，两个个体在交叉点位置，两个个体在交叉点位置互换部分基因码，形成两个子个体，如图 5.2 所示。



变异运算使用基本位变异算子或均匀变异算子。

为了避免问题过早收敛，对于二进制的基因码组成的个体种群，实现基因码的小概率翻转，即 0 变为 1，而 1 变为 0，如图 5.3 所示。



#### 4.4 遗传算法操作过程

第一步，初始化。采用实数编码方案进行染色体编码，确定种群规模为  $N$ ，应用 MATLAB 遗传算法工具箱函数 `crtbp` 产生种群规模为  $N$  的初始种群。初始种群的产生意味着产生了  $N$  个初始货位的分配方案，将种群中坐标重复的或者超过优化区域的染色体去掉，并补充新的染色体，使之满足种群的规模为  $N$ 。

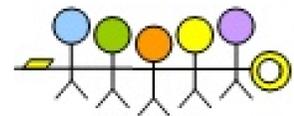
第二步，种群适应度评价。评价初始种群的目标函数值和适应值。

第三步，通过选择、交叉、变异操作，产生新一代群体。

第四步，对产生的新种群中的个体进行评价，计算目标函数值和适应值。

第五步，记录本带群体中的最优个体，保留最优解。

第六步，完成预先设定的进化代数后算法停止。



## 4.5 系统仿真

考虑到现在所学知识的限制，现在只研究单排货架，并且假设堆垛机一直匀速行驶，且其行驶速度和升降速度  $V_x=V_y=1\text{m/s}$ ，现在对这类货物的出入库频率重量和所占货格数假设如下

货品编号	出入库频率	重量	所占货格数
1	0.1	100	50
2	0.2	50	30
3	0.16	40	20
4	0.24	80	40
5	0.1	90	15
6	0.05	100	5
7	0.05	60	10
8	0.04	80	10
9	0.03	70	6
10	0.03	60	14

该计算中燃热提供有 200 个基因位，每个基因长度为，每个基因长度为 4，染色体长度为  $200*4$ ，在计算中假设染色体种群代数为  $M=400$ ，进化代数为  $T=400$ ，交叉概率  $P_c=0.8$ ，变异概率  $P_m=0.2$

选择：我们应用 MATLAB 遗传算法工具箱中的 `crtbase` 函数与 `crtbp` 函数产生初始种群格式如下示：

`NIND=400;`

`BaseV=crtbase(800, 4);`

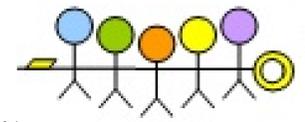
`Chrom=crtbp(Nnm3, BaseV)+ones(NrYD, 800);`

将目标函数转换为可是别的形式如下：

$$\text{ObjV} = 0.6 f_1(\text{chrom}) + 0.4 f_2(\text{chrom})$$

$$\text{FitnV} = \text{eavl}(\text{chrom}) = \frac{r_{\max} + r(\text{chrom}) + 0.08}{r_{\max} - r_{\min} + 0.08}$$

根据目标函数的适应度值进行选择与基因重组工作。 本文采用



MATLAB遗传算法工具箱函数中的轮盘选择 (rws) 函数进行选择操作；

采用单点交叉 (xovsp) 函数进行基因的重组。格式如下所示：

```
SelCh=select('rws', Chrom, FitnV) ;
```

```
SelCh=recombine('Xovsp', SelCh, 0.8) ;
```

变异

```
f=rep([1 : 20], [1, 10]) ;
```

```
SelCh=mutbga(SelCh, f) ;
```

```
SelCh=fix(SelCh) ;
```

计算子代目标函数如下：

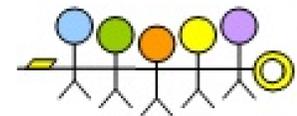
$$\text{ObjVSel} = 0.6 * f_1(\text{SelCh}) + 0.4 * f_2(\text{SelCh})$$

#### 4.6 仿真优化结果

货位进行优化前，布局不合理，仓库的综合性能比较差。优化后

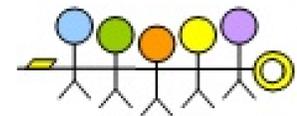
的货位分配及存放的货物重量与出入库频率分别如下表所示：

列层	1	2	3	4	5	6	7	8	9	10
1	80	80	80	80	100	50	50	50	40	80
2	80	80	80	80	100	50	50	50	40	80
3	80	80	80	80	100	50	50	50	40	80
4	80	80	80	80	100	50	50	50	40	80
5	80	80	80	80	100	50	50	40	40	80
6	80	80	80	80	100	50	50	40	40	80
7	80	80	80	80	100	50	50	40	40	80
8	80	80	80	80	100	50	50	40	60	80
9	80	80	80	80	100	50	50	40	60	80
10	80	80	80	80	100	50	50	40	60	80
11	100	100	100	100	100	50	50	40	60	60



12	100	100	100	100	100	50	50	40	60	60
13	100	100	100	100	100	50	50	40	60	60
14	100	100	100	100	100	40	40	40	60	60
15	100	100	100	100	100	40	90	60	60	60
16	100	100	100	100	90	90	90	70	60	60
17	100	100	100	100	90	90	90	70	60	60
18	100	100	100	100	90	90	90	70	60	60
19	100	100	100	100	90	90	90	70	60	60
20	100	100	100	100	90	90	90	70	60	60

列层	1	2	3	4	5	6	7	8	9	10
1	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.2	0.16	0.04
2	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.2	0.16	0.04
3	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.2	0.16	0.04
4	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.2	0.16	0.04
5	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.16	0.16	0.04
6	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.16	0.16	0.04
7	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.16	0.16	0.04
8	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.16	0.05	0.04
9	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.16	0.05	0.04
列层	1	2	3	4	5	6	7	8	9	10
10	0.24	0.24	0.24	0.24	0.1	0.2	0.2	0.16	0.05	0.04
11	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.16	0.05	0.05
12	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.16	0.05	0.05
13	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.16	0.05	0.03
14	0.1	0.1	0.1	0.1	0.1	0.16	0.16	0.16	0.05	0.03
15	0.1	0.1	0.1	0.1	0.1	0.16	0.1	0.05	0.03	0.03
16	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.03	0.03	0.03
17	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.03	0.03	0.03



18	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.03	0.03	0.03
19	0.1	0.1	0.05	0.05	0.1	0.1	0.1	0.03	0.03	0.03
20	0.05	0.05	0.05	0.05	0.1	0.1	0.03	0.03	0.03	0.03

优化后的货位在满足提高出入库效率的基础上同时也使得货架的重心比较低，符合货架稳定性的要求。由此可见，在具体货位分配优化应用过程中，可能由于仓库规模很大导致需要处理的货物比较多，如果使用传统的求解方法进行货位优化求解时可能导致计算时间成级数增长，从而无法应用到实际问题中。远过上面所示的优化结果，我们可以得到这样一个结论，即应用遗传算法进行货位优化时可以有效地对问题进行求解，且其算法非常简单，虽然无比保证获得最优解，但是由于收敛速度快，则可以在较短的时间段内获得一个令人满意的近似解，所以应用遗传算法对货位优化进行求解具有较大的应用价值。