

基于改进离散粒子群算法的危化品仓库垛位布局优化研究

戴波^{*1}, 林双双², 张岩¹, 刘学君¹

(1. 北京石油化工学院 信息工程学院, 北京 102617;

2. 北京化工大学 信息科学与技术学院, 北京 100029)

摘要: 堆垛是危化品仓储的重要方式之一,其布局优化是带有特殊约束的非确定性多项式难题(NP-hard). 为此建立了以仓储利用率为目标函数,危化品仓储安全距离为约束条件的仓储堆垛布局优化数学模型. 针对此问题的非二进制离散特性,提出了符合危化品垛位布局优化问题的离散粒子群算法,该算法重新定义了速度与位置更新公式,设计了最高水平线分层排放策略,实现了危化品仓库安全约束条件下适应度函数的计算,优化了垛位与通道位置的布局. 实验表明:该算法在满足危化品仓储安全的条件下,可有效提高货物堆垛仓储的利用率.

关键词: 危化品仓库;布局优化;NP-hard;离散粒子群

中图分类号: TQ086; TP301.6 **文献标识码:** A **doi:** 10.7511/dllgxb202003009

0 引言

在《常用危险化学品贮存通则》^[1]中,国家明确规定了危险化学品(后面简称危化品)仓库的存储要求,对于以堆垛码放的危化品仓库,其码放方案需满足存储安全距离的要求. 在传统危化品仓库管理中大多采用人工随机堆垛方式,非常容易造成货物堆垛距离不符合国家安全规定,或仓库空间浪费的问题,开展危化品仓库堆垛位置的布局优化研究有重要的现实意义. 因此本文建立危化品仓库垛位布局模型,并设计最高水平线分层排放策略,将该布局问题转化为带有一定约束条件的非确定性多项式难题的组合优化问题^[2-3],属于 NP-hard 难题. 类似的这类典型问题主要有 TSP 问题^[4]、背包问题^[5]、调度问题^[6]、排样问题^[7]等.

目前主要的优化求解算法有数学规划法、启发式算法、智能算法等. 20 世纪七八十年代以来,随着智能优化算法的不断出现,如遗传算法^[8]、禁忌搜索^[9]、蚁群算法^[10]、粒子群优化算法^[11]、模拟退火算法^[12]等,这些智能算法具有全局搜索能

力,可以很好地解决 NP-hard 的“组合爆炸”问题,在解决组合优化问题上显示出一定的优越性. 粒子群优化(particle swarm optimization, PSO)算法是由 Eberhart 和 Kennedy 于 1995 年提出的一种群智能优化算法^[13],同其他算法相比较,PSO 算法更容易实现且具有更强的全局优化能力. 但 PSO 算法中速度、位置等都是连续的量,大多应用于连续空间的优化. 然而,本文研究的危化品仓库堆垛位置布局的优化问题是离散的,基本的粒子群算法无法解决本文的问题.

近年来,关于粒子群优化算法用于求解离散优化问题的研究,也日益引起了人们的注意,出现了一系列的离散 PSO 算法,并在实际工程问题中得到了较好的应用^[14]. Kennedy 和 Eberhart 提出了针对 0-1 问题的二进制粒子群优化(BPSO)算法,该算法沿用了基本粒子群优化的速度更新公式,即速度仍作用于连续空间,而位置则利用 Sigmoid 函数将其离散化^[15]. Ashoka Varthanan 等以二进制粒子群算法来解决轴承制造业的生产分配问题^[16],钱真坤采用重新定义了的粒子群更新公式来解决商旅问题^[17]. 但传统 BPSO 算法编

收稿日期: 2019-12-23; 修回日期: 2020-01-06.

基金项目: 国家重点研发计划资助项目(2016YFC0801502);北京市教育委员会科技能力提升计划资助项目(TSJH20161001701).

作者简介: 戴波^{*}(1962-),男,教授, E-mail: daibo@bipt.edu.cn;林双双(1995-),女,硕士, E-mail: 1278514944@qq.com.

码方式和粒子群速度与位置更新公式无法计算本文的危化品仓库垛位布局优化问题. 本文根据危化品仓库垛位布局的实际问题重新定义粒子群算法中的更新公式, 来实现粒子群算法的完全离散化, 进而优化求得最佳布局方案.

1 模型建立

仓库作为危化品的存储地点具有存储较集中, 一旦发生危险危害性大的特点, 所以其内部在进行危化品仓库布局方案设计时, 要考虑布局方案的安全性. 本文在设计布局方案时以一个布局空间为基准, 将仓库利用率作为优化目标函数, 将安全距离以及仓库便利需求作为约束条件的方式来建立优化模型.

1.1 目标函数

本文以仓库俯视图的角度将危化品仓库垛位布局问题转化为二维矩形布局问题. 其具体描述如下:

现有一长度为 L , 宽度为 W 的危化品仓库, 拟设置 N 个长 $\times$ 宽为 $l_i \times w_i (i=1, \dots, N)$ 的堆垛位置, 垛距为 a , 墙距为 b , 主通道宽度为 R , 辅通道宽度为 r (所有长度单位都为 m). 以仓库利用率 f 作为目标函数:

$$f = \frac{\sum_{i=1}^N (l_i \times w_i)}{L \times H} \quad (1)$$

其中 $\sum_{i=1}^N (l_i \times w_i)$ 为所有待布局堆垛底面积, 由于以仓库俯视图角度进行布局, 可以忽略垛高; H 为布局结果所占仓库有效宽度.

1.2 约束条件

布局时考虑安全距离问题, 以及仓库实际便利性问题, 可以表述成以下约束条件:

(1) 所有堆垛在仓库内且同一行相邻两个堆垛距离为 a ;

(2) 仓库内不同行的同列堆垛距离应大于等于 a , 且堆垛不应出现插空现象;

(3) 靠墙的堆垛与墙之间的距离应为 b .

将以上约束表述为数学描述, 建立一个二维坐标系如图 1 所示, 将去除墙距的实际待布区域的左下角坐标设为 $(0, 0)$, (x_i, y_j) 表示布局后第 i 行 j 列个堆垛在仓库中位置的左下角坐标.

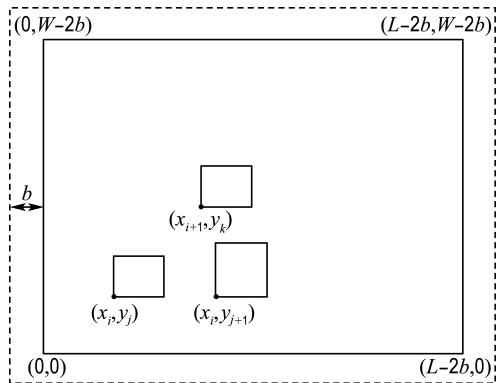


图 1 仓库示意图

Fig. 1 Warehouse diagram

其数学描述为

$$\begin{aligned} \text{s. t. } & 0 < x_i < L - 2b - l_{ij} \\ & 0 < y_j < W - 2b - w_{ij} \\ & |(x_i, y_j) - (x_i, y_{j+1})| = l_{ij} + a \\ & |(x_{i+1}, y_j) - (x_i, y_j)| \geq \\ & \quad \sqrt{(x_{i+1} - x_i)^2 + (w_{ij} + a)^2} \end{aligned} \quad (2)$$

式中: l_{ij} 表示第 i 行 j 列个堆垛的长, w_{ij} 表示第 i 行 j 列个堆垛的宽.

2 改进离散粒子群算法设计

2.1 基本 PSO 和 BPSO 算法

基本粒子群优化 (PSO) 算法是基于种群的迭代搜索算法, 算法随机初始化一组解, 在迭代过程中通过跟踪粒子本身最优解 (即个体极值 p_{best}) 和种群最优解 (全局极值 g_{best}) 来寻找最优解. 粒子通过如下两个公式来更新速度和位置:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (p_{\text{best}_i} - x_i^k) + c_2 r_2 (g_{\text{best}} - x_i^k) \quad (3)$$

$$x_i^{k+1} = x_i^k + v_i^k \quad (4)$$

式中: ω 是惯性权重, x_i 和 v_i 分别为第 i 个粒子的位置和速度, c_1 和 c_2 为加速因子, p_{best_i} 是第 i 个粒子的最优位置, g_{best} 为全局最优位置, k 为迭代次数.

Kennedy 和 Eberhart 于 1997 年提出了一种针对 0-1 规划问题的二进制 PSO (BPSO) 算法. 具体公式如下:

$$v_{id}^{k+1} = v_{id}^k + \varphi (p_{id}^k - x_{id}^k) + \varphi (p_{gd}^k - x_{id}^k) \quad (5)$$

$$x_{id}^{k+1} = \begin{cases} 1; & \text{rand}() < \text{sig}(v_{id}) \\ 0; & \text{rand}() \geq \text{sig}(v_{id}) \end{cases} \quad (6)$$

$$\text{sig}(v_{id}) = \frac{1}{1 + \exp(-v_{id})} \quad (7)$$

式中: $rand()$ 表示分布在 $[0,1]$ 的随机数; $sig(v_{id})$ 代表由速度 v_{id} 决定的一个范围在 $[0,1]$ 的概率函数。

2.2 改进离散粒子群算法设计

2.2.1 算法具体定义操作

(1) 粒子位置与速度定义

对于危化品仓库的垛位,其堆放顺序与横竖摆放(通常以较长的边为堆垛长时为横放,否则为竖放)影响着其排放策略最终的利用率,因此将粒子群中粒子的位置定义为 $\mathbf{x}=(r_1x_1 \ r_2x_2 \ \cdots \ r_Nx_N)$, $1 \leq i \leq N$, $1 \leq x_i \leq N$; $r_i=1$ 或 -1 ,其中 N 为待布垛位个数, r_i 代表堆垛的横竖摆放, r_i 等于1时代表横放, r_i 等于 -1 时代表竖放; x_i 代表该堆垛的序号,其位置代表摆放顺序。同样将粒子的速度定义为一个向量 $\mathbf{v}=(t_1v_1 \ t_2v_2 \ \cdots \ t_Nv_N)$, $1 \leq i \leq N$, $1 \leq v_i \leq N$; $t_i=1$ 或 -1 ,这里 t_i 等于1代表垛位不旋转, t_i 等于 -1 代表垛位旋转 90° 。

(2) 粒子位置与速度加法运算法则

基本 PSO 算法粒子的位置更新是通过粒子位置与速度相加,即速度与位置相加会得到一个新的位置,即 $\mathbf{x}'=\mathbf{x} \oplus \mathbf{v}$,基于本文危险化学品仓库垛位布局问题,对粒子位置与速度相加运算法则做出如下定义:

$$\mathbf{x}'=\mathbf{x} \oplus \mathbf{v}=\begin{cases} r'_i=r_i, x'_{v_i}=x_i; & t_i=1 \\ r'_i=-r_i, x'_{v_i}=x_i; & t_i=-1 \end{cases} \quad (8)$$

(3) 粒子位置与位置减法运算法则

粒子位置与速度相加会得到一个新的位置,因此粒子位置与位置相减会得到粒子的速度,即 $\mathbf{v}=\mathbf{x}'-\mathbf{x}$,其具体运算法则如下:

$$\mathbf{v}=\mathbf{x}'-\mathbf{x}=\begin{cases} v_i=k; & x_i=x'_k \\ t_i=1; & r_i=r'_k \\ t_i=-1; & r_i \neq r'_k \end{cases} \quad (9)$$

上述定义的粒子位置与速度加法运算和粒子位置与位置减法运算可以相互推导,符合等式的运算法则,例如: $\mathbf{x}=(3 \ -2 \ 1 \ -5 \ 4)$, $\mathbf{v}=(2 \ 3 \ -1 \ 4 \ 5)$,则

$$\mathbf{x}'=\mathbf{x} \oplus \mathbf{v}=(-1 \ 3 \ -2 \ -5 \ 4)$$

$$\mathbf{v}=\mathbf{x}'-\mathbf{x}=(2 \ 3 \ -1 \ 4 \ 5)$$

(4) 粒子速度与速度加法运算法则

粒子两个速度相加会得到一个新的速度,即 $\mathbf{v}=\mathbf{v}_1 \oplus \mathbf{v}_2$,具体运算法则如下:

$$\mathbf{v}=\mathbf{v}_1 \oplus \mathbf{v}_2=\begin{cases} t_i=t_{1i}, v_{v_{2i}}=v_{1i}; & t_{2i}=1 \\ t_i=-t_{1i}, v_{v_{2i}}=v_{1i}; & t_{2i}=-1 \end{cases} \quad (10)$$

(5) 粒子速度数乘运算法则

$$c \otimes \mathbf{v}=\begin{cases} \mathbf{v}; & c \times rand() \leq 1 \\ 0; & c \times rand() > 1 \end{cases} \quad (11)$$

该式含义为速度以 $1/c$ 的概率保留,通过设置参数 c 可以控制速度的保留概率。

(6) 粒子运动方程

根据上述定义的运算规则,可以将粒子运动方程定义为

$$\mathbf{v}^{k+1}=\omega \otimes \mathbf{v}^k \oplus c_1 \otimes (\mathbf{p}_{\text{best}}-\mathbf{x}^k) \oplus c_2 \otimes (\mathbf{g}_{\text{best}}-\mathbf{x}^k) \quad (12)$$

$$\mathbf{x}^{k+1}=\mathbf{x}^k \oplus \mathbf{v}^k \quad (13)$$

其中惯性权重 ω 体现的是粒子继承先前速度的能力,在该改进离散粒子群中引入惯性权重指数递减概念^[18],即:

$$\omega(k)=\omega_{\text{end}}\left(\frac{\omega_{\text{start}}}{\omega_{\text{end}}}\right)^{1-k/T_{\text{max}}} \quad (14)$$

式中: T_{max} 表示最大迭代次数,通常取 $\omega_{\text{start}}=0.9$, $\omega_{\text{end}}=0.4$ 。

2.2.2 最高水平线的分层排放策略 在布局中需要排放策略来实现目标函数也就是算法中适应度值的计算,即解码过程。目前在排样问题和装箱问题中用到一些排放方法。在 Bottom Left (BL) 算法中,箱子从右上角开始向下并尽量向左移动,该算法一个主要的缺点是无法填充先前安排的箱子导致的可用空隙;而 BL-Fill (BLF) 算法能够填充这些空隙^[19]。Thomas 等^[20]提出一个或多个矩形的宽度组合可以使水平线表面最大化来实现矩形定位。马康等提出了在传统最低水平线法的基础上引入搜索策略的最低水平搜索算法来解决传统的最低水平线法没有能利用最高轮廓线以下已排矩形件间的空隙问题^[21]。上述算法无法满足本文危化品仓库布局的全部约束条件,故本文提出了基于最高水平线的分层排放方法来计算目标函数利用率,即粒子群算法中的适应度值。具体方法步骤如下:

(1) 以去除四周墙边距的实际待布区域开始放置堆垛,设置起始底边轮廓线 Line 1 长度为实际待布区域的长 $L-2b$;

(2) 每当放置一个 $l_i \times w_i$ ($i=1, \cdots, N$) 堆垛时,对比底边轮廓线 Line i 和 l_i 如若:

a. Line i 大于等于 l_i , 记录此时水平线高度 high i , 并更新底边轮廓线 Line $i+1$ 为 Line i 长度减掉 l_i+a ;

b. Line i 小于 l_i , 以当前水平线高度最高的 high i 位置分层为辅通道位置, Line i 重新等于起始值 $L-2b$, 并重复步骤 a.

(3) 重复步骤(2)直至放置完所有矩形堆垛.

根据该方法排完待布局垛位后再对每层堆垛按照水平高度线 high i 由小至大重新排列, high i 相同时底边小的在左边. 按照上述方法实现垛位安全约束下的布局同时实现辅通道位置的布局, 如图 2 所示. 再根据式(1)计算该布局区域的利用率, 即每个可行解的适应度值.

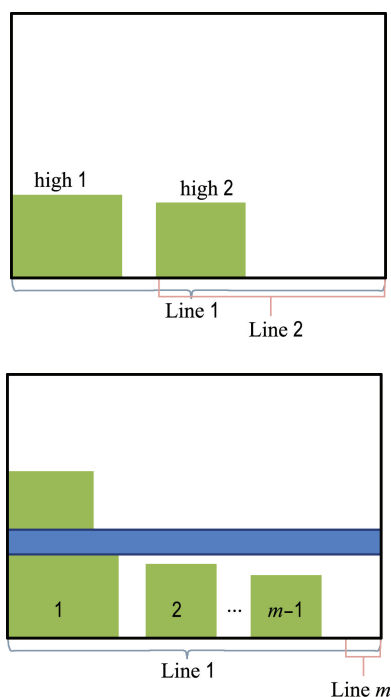


图 2 堆垛定位算法示意图

Fig. 2 Schematic diagram of stacking positioning algorithm

2.3 改进离散粒子群算法实现

(1) 初始化粒子种群

将 N 个堆垛标号为 1 到 N , 随机产生 M 个形式相同的序列, 如 $1 \leq i \leq N, \mathbf{x} = (r_1 x_1 \quad r_2 x_2 \quad \cdots \quad r_i x_i \quad \cdots \quad r_N x_N), 1 \leq x_i \leq N, r_i = 1$ 或 -1 的序列, 记为种群的初始位置. 同样随机产生 M 个 $\mathbf{v} = (t_1 v_1 \quad t_2 v_2 \quad \cdots \quad t_i v_i \quad \cdots \quad t_N v_N), 1 \leq i \leq N, 1 \leq v_i \leq N, t_i = 1$ 或 -1 , 记为种群初始速度. 其中 M 即为种群中粒子个数, N 为一个粒子的长度即

待布局垛位个数. 并通过多次实验设置加速因子 c_1 和 c_2 .

(2) 计算适应度值

设置最大迭代次数 $T_{\max}$, 以目标函数利用率作为适应度函数, 根据每个粒子的编码情况按照上述的最高水平线分层排放方法进行排放, 计算每个粒子对应的排放后仓库利用率, 即为每个粒子的适应度值.

(3) 确定个体极值以及全局极值

在每次迭代过程中, 选取每个粒子迭代中适应度值最大的粒子作为个体极值 p_{best} , 选取所有粒子中在所有当前迭代中适应度值最大的粒子作为全局极值 g_{best} .

(4) 更新粒子

根据式(14)计算惯性权重值, 并根据式(12)、(13)更新粒子速度与位置.

(5) 判断是否终止算法

判断是否满足迭代终止条件(最大迭代次数), 是则算法结束, 否则跳转至(2). 算法的流程图如图 3 所示.

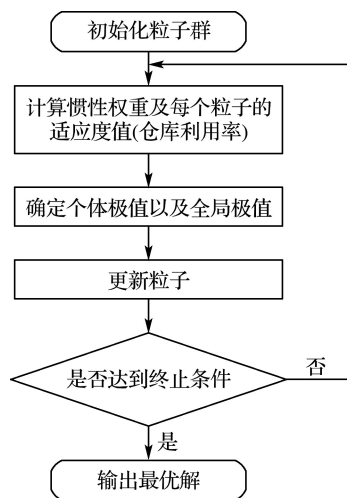


图 3 改进离散粒子群算法流程图

Fig. 3 Flow chart of improved discrete particle swarm optimization algorithm

3 实验结果

3.1 固体包装箱类危化品存储仓库布局

为了验证本文垛位布局优化算法的有效性, 现有一拟存储固体包装箱类危化品长为 28 m、宽为 15 m 的危化品仓库, 该仓库大门在距仓库左

边长为 13.5 m 的位置,大门所对位置为主通道位置,其宽度为 1.5 m,将其分为两个待布区域. 现该仓库右区域拟放置 10 个 1.5 m×1.3 m, 10 个 1.5 m×1.5 m, 10 个 2 m×2 m 的堆垛. 左区域拟放置 10 个 2 m×2 m, 10 个 2 m×1.5 m, 10 个 1.5 m×1.5 m 的堆垛,墙距为 0.5 m,垛距为 0.5 m. 在危化品包装箱来货后,在仓库外将危化品包装箱以堆垛的方式码放在垫板上,最后用叉车将其运送至仓库内. 一般情况下仓库管理者会按照叉车叉送的顺序从某个角落进行摆放,可能造成不符合安全距离要求及空间浪费的问题.

现分别将左右区域待布局的堆垛编号为 1~30. 根据本文最高水平线分层排放策略分别在两个待布局区域按顺序排放堆垛,仓库两个区域均不能排放下所有的堆垛,因此只能减少堆垛个数进行布局,最终结果如图 4 所示. 相比于人工随机码放堆垛该排放方案可以满足安全距离要求,但从实际布局结果可以看出该方案也会造成一定空间的浪费.

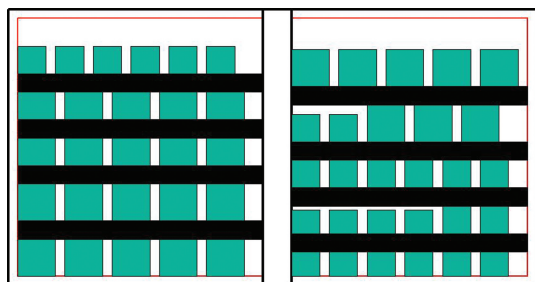


图4 最高水平线分层排放实际布局结果

Fig. 4 The actual layout result of the highest level stratified arrangement

根据本文的改进离散粒子群算法对上述待布局区域的垛位进行优化.

左区域布局优化结果为 $x = (25 \quad -27 \quad 18 \quad 19 \quad -15 \quad -9 \quad -21 \quad 28 \quad 23 \quad 3 \quad -1 \quad 4 \quad -22 \quad -26 \quad 14 \quad 20 \quad -11 \quad -2 \quad 29 \quad 16 \quad -12 \quad -13 \quad 10 \quad 7 \quad -24 \quad 30 \quad -17 \quad -6 \quad -8 \quad 5)$; 其意义为编号为 25 的堆垛第一个排放, 编号为 27 的堆垛第二个排放并旋转 90° , 依此类推, 其辅通道位置为距底边墙 2.5、5.5、8.5、11.5 m 处, 宽度为 1 m.

右区域布局优化结果为 $x = (10 \quad -4 \quad 12$

$19 \quad 30 \quad -21 \quad 6 \quad 8 \quad -18 \quad 16 \quad 22 \quad -26 \quad 1 \quad -9 \quad -13 \quad 17 \quad -23 \quad 25 \quad 3 \quad 7 \quad 5 \quad 14 \quad 28 \quad -29 \quad -2 \quad 11 \quad 20 \quad -15 \quad -27 \quad -24)$; 辅通道位置为距底边墙 2.5、5.5、8.5、11.5 m 处, 宽度为 1 m, 整个仓库布局结果如图 5 所示.

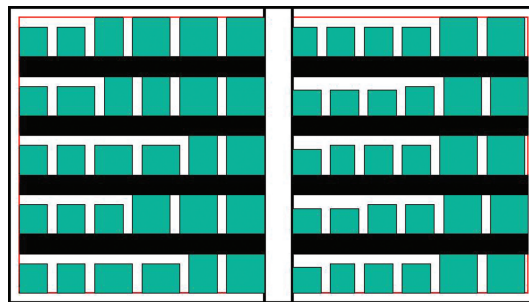


图5 优化后布局结果

Fig. 5 Layout results after optimization

采用最高水平线分层排放堆垛和采用遗传算法优化以及采用本文改进离散粒子群算法优化后仓库左右两区域仓库面积利用率对比如表 1 所示, 证明了本文算法的优越性.

表1 固体类危化品垛位布局优化前后仓库面积利用率对比

Tab. 1 Comparison of solid hazardous chemicals warehouse area utilization before and after optimization of stacking position layout

区域	仓库面积利用率/%		
	最高水平线 分层方法	遗传算法 优化后	改进离散粒子 群优化后
左区域	41.23	42.34	45.68
右区域	37.94	40.00	42.05

3.2 压缩气体和液化气体类

压缩气体和液化气体类危化品需用钢瓶盛装, 在存储以及装卸过程中不得摔扔、撞击和在地面滚动. 故在该类仓库内以将钢瓶装入集装格的方式码垛. 现有一长为 32.5 m、宽为 23.0 m 的仓库, 该仓库大门在仓库距左边长为 16.5 m 的位置, 大门所对位置为主通道位置, 其宽度为 1.5 m. 仓库现有一批液化 CH_4 钢瓶装入集装格待放入仓库的左区域, 其中有 11 个集装格的底边长×宽为 1.4 m×1.4 m, 10 个 1.8 m×1.4 m, 11 个 1.8 m×1.8 m 待入库, 其安全墙距为 0.5 m, 垛距为 0.5 m. 将上述集装格编号为 1~32, 经过

本文布局优化算法得出的方案为 $x=(-6 \quad -12 \quad -17 \quad -15 \quad -19 \quad -26 \quad 22 \quad -30 \quad 3 \quad 5 \quad -20 \quad -18 \quad -13 \quad 31 \quad -24 \quad -29 \quad -1 \quad -11 \quad -4 \quad 10 \quad 9 \quad 2 \quad 23 \quad 25 \quad -16 \quad -14 \quad -21 \quad -8 \quad -7 \quad 27 \quad 28 \quad -32)$; 在距底边墙 2.3、5.1、7.9 m 处设置辅通道排放完所有置物架后未将整个仓库左区域占满, 可以在距底边墙 10.7 m 设置一宽为 1.5 m 的主通道, 用来隔离剩余区域将要放置的可同库存储的货物, 具体布局优化结果如图 6 所示。

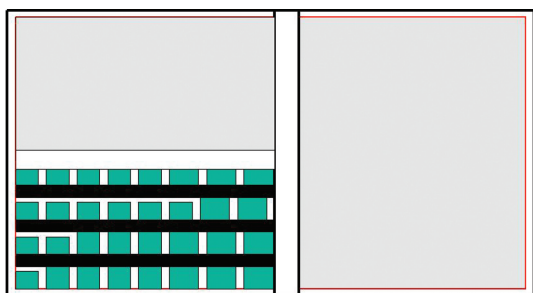


图 6 液体危化品优化后布局结果

Fig. 6 Layout results after optimization for liquid hazardous chemicals

以最高水平线分层排放方法布局但未优化的布局结果如图 7 所示。

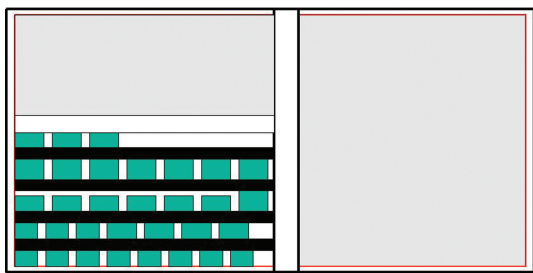


图 7 布局优化前结果

Fig. 7 Layout results before optimization

布局方案优化前后以及与遗传算法优化后的利用率对比如表 2 所示。

由上述实验结果可以得出本文布局优化算法可以保证安全约束条件, 并在同等安全约束条件下有效提高仓库利用率。在实际仓库布局时仓库管理者可以将待布堆垛或置物架进行编号, 根据优化布局方案先划定通道位置, 再按照优化方案从左下方入库。

表 2 液体类危化品垛位布局优化前后仓库面积利用率对比

Tab. 2 Comparison of liquid hazardous chemicals warehouse area utilization before and after optimization of stacking position layout

优化方法	仓库面积利用率/%
最高水平线分层方法	39.32
遗传算法优化后	43.00
改进离散粒子群优化后	46.67

4 结 论

(1) 针对保证安全距离条件下, 在仓库同一区域放置不同大小的危化品堆垛问题, 建立了垛位布局问题的数学模型, 即以安全距离为约束条件和利用率为优化目标的目标函数。

(2) 设计了符合危化品仓库垛位布局的编码方式, 重新定义了粒子群算法中粒子的速度与位置的加法、位置与位置的减法、速度与速度的加法、速度数乘的运算规则, 实现了粒子群算法的离散化。

(3) 设计了基于最高水平线分层的堆垛定位排放策略, 来计算目标函数利用率, 即改进离散粒子群的适应度值的计算, 最终实现了在安全约束条件下提高仓库利用率的危化品仓库垛位布局。

目前未对仓库分区情况进行考虑, 在后续课题中将研究动态分区时的危化品垛位布局问题。

参考文献:

- [1] 国家技术监督局. 常用危险化学品贮存通则: GB 15603—1995 [S]. 北京: 国家技术监督局, 1995.
State Bureau of Technical Supervision. Rule for Storage of Chemical Dangers: GB 15603-1995 [S]. Beijing: State Bureau of Technical Supervision, 1995. (in Chinese)
- [2] 戴 波, 李雁飞, 陈增强, 等. 危险化学品仓库堆垛安全布局建模及优化研究 [J]. 中国安全生产科学技术, 2018, 14(4): 147-154.
DAI Bo, LI Yanfei, CHEN Zengqiang, *et al.* Study on modeling and optimization for safe layout of stacking in dangerous chemical warehouse [J]. *Journal of Safety Science and Technology*, 2018, 14(4): 147-154. (in Chinese)

- [3] DAI Bo, LI Yanfei, REN Haisheng, *et al.* Research on optimization for safe layout of hazardous chemicals warehouse based on genetic algorithm [J]. **IFAC-Papers OnLine**, 2018, **51**(18): 245-250.
- [4] LUO Wei, LIN Dong, FENG Xinxin. An improved ant colony optimization and its application on TSP problem [C] // **Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data, iThings-GreenCom-CPSCoM-Smart Data 2016**. Chengdu: IEEE, 2016.
- [5] 万晓琼, 张惠珍. 求解 0-1 背包问题的混合蝙蝠算法 [J]. 计算机应用研究, 2019, **36**(9): 2579-2583.
WAN Xiaoqiong, ZHANG Huizhen. Hybrid bat algorithm for solving 0-1 knapsack problem [J]. **Application Research of Computers**, 2019, **36**(9): 2579-2583. (in Chinese)
- [6] ZHOU Wei, BU Yanping, ZHOU Yeqing. An improved genetic algorithm for solving flexible job shop scheduling problem [C] // **2013 25th Chinese Control and Decision Conference, CCDC 2013**. Guiyang: IEEE Computer Society, 2013: 4553-4558.
- [7] 朱强, 薛峰, 郑仕勇, 等. 约束二维排样问题的一种求解算法 [J]. 锻压技术, 2016, **41**(9): 148-152.
ZHU Qiang, XUE Feng, ZHENG Shiyong, *et al.* An algorithm of the constrained two-dimensional nesting [J]. **Forging & Stamping Technology**, 2016, **41**(9): 148-152. (in Chinese)
- [8] PANCHAL G, PANCHAL D. Solving NP hard problems using genetic algorithm [J]. **International Journal of Computer Science and Information Technologies**, 2015, **6**(2): 1824-1827.
- [9] 王运龙, 吴张盼, 李楷, 等. 基于禁忌搜索算法的船舶舱室智能布局设计 [J]. 华中科技大学学报 (自然科学版), 2018, **46**(6): 49-53.
WANG Yunlong, WU Zhangpan, LI Kai, *et al.* Ship cabin intelligent layout design based on tabu search algorithm [J]. **Journal of Huazhong University of Science and Technology (Nature Science Edition)**, 2018, **46**(6): 49-53. (in Chinese)
- [10] MAHI M, BAYKAN Ö K, KODAZ H. A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem [J]. **Applied Soft Computing Journal**, 2015, **30**: 484-490.
- [11] DU K L, SWAMY M N S. Particle swarm optimization [M] // **Search and Optimization by Metaheuristics**. Berlin: Springer International Publishing, 2016.
- [12] 何庆, 吴意乐, 徐同伟. 改进遗传模拟退火算法在 TSP 优化中的应用 [J]. 控制与决策, 2018, **33**(2): 219-225.
HE Qing, WU Yile, XU Tongwei. Application of improved genetic simulated annealing algorithm in TSP optimization [J]. **Control and Decision**, 2018, **33**(2): 219-225. (in Chinese)
- [13] HE Jie, GUO Hui. A modified particle swarm optimization algorithm [J]. **Telkomnika - Indonesian Journal of Electrical Engineering**, 2013, **11**(10): 6209-6215.
- [14] 程毕芸, 鲁海燕, 徐向平, 等. 求解旅行商问题的改进局部搜索混沌离散粒子群优化算法 [J]. 计算机应用, 2016, **36**(1): 138-142, 149.
CHENG Biyun, LU Haiyan, XU Xiangping, *et al.* Improved local-search-based chaotic discrete particle swarm optimization algorithm for solving traveling salesman problem [J]. **Journal of Computer Applications**, 2016, **36**(1): 138-142, 149. (in Chinese)
- [15] NGUYEN B H, XUE B, ANDREA P. A novel binary particle swarm optimization algorithm and its applications on knapsack and feature selection problems [M] // **Intelligent and Evolutionary Systems**. Berlin: Springer International Publishing, 2017.
- [16] ASHOKA VARTHANAN P, MURUGAN N, MOHAN KUMAR G M. An AHP based heuristic DPSO algorithm for generating multi criteria production - distribution plan [J]. **Journal of Manufacturing Systems**, 2013, **32**(4): 632-647.
- [17] 钱真坤. 基于离散粒子群优化算法的含权旅行商问题新解法 [J]. 计算机应用与软件, 2019, **36**(1): 127-131.
QIAN Zhenkun. A new method of weighted traveling salesman problem based on discrete particle swarm optimization [J]. **Computer**

- Applications and Software**, 2019, **36**(1): 127-131.
(in Chinese)
- [18] 封京梅,刘三阳. 基于惯性权重指数递减的粒子群优化算法求解绝对值方程 [J]. 吉林大学学报(理学版), 2016, **54**(6): 1265-1269.
FENG Jingmei, LIU Sanyang. Particle swarm optimization algorithm based on inertia weight exponentially decreasing for solving absolute value equations [J]. **Journal of Jilin University (Science Edition)**, 2016, **54**(6): 1265-1269. (in Chinese)
- [19] DAODEN K, THAIUPATHUMP T. Applying shuffled frog leaping algorithm and bottom left fill algorithm in rectangular packing problem [C] // **2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)**. Macau: IEEE, 2017: 136-139.
- [20] THOMAS J, CHAUDHARIN S. Design of efficient packing system using genetic algorithm based on hyper heuristic approach [J]. **Advances in Engineering Software**, 2014, **73**: 45-52.
- [21] 马 康,高 尚. 分布估计算法求解矩形件排样优化问题 [J]. 电子设计工程, 2017, **25**(2): 49-54.
MA Kang, GAO Shang. Solution to optimize cutting pattern in rectangular packing problem based on estimation of distribution algorithm [J]. **Electronic Design Engineering**, 2017, **25**(2): 49-54. (in Chinese)

Research on optimization of hazardous chemicals warehouse stacking position layout based on improved discrete particle swarm optimization algorithm

DAI Bo^{*1}, LIN Shuangshuang², ZHANG Yan¹, LIU Xuejun¹

(1.College of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing 102617, China;

2.College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: Stacking is one of the important ways of hazardous chemicals warehousing storage. The layout optimization is a non-deterministic polynomial problem with special constraints (NP-hard). To solve this problem, the mathematical model of storage stacking optimization based on the utilization rate of storage capacity as the objective function and the safety distance of hazardous chemicals storage as the constraints condition is established. To solve the non-binary discrete characteristics of this problem, the discrete particle swarm optimization algorithm for the optimization of the layout of hazardous chemicals warehouse is proposed, which redefines the speed and position update formula, designs the highest level stratified arrangement strategy, realizes the calculation of the fitness function under the safety constraints of hazardous chemicals warehouse, and optimizes the layout of the position and channel position. Experiments show that the algorithm can effectively improve the utilization rate of cargo storage under the condition of meeting the safety of hazardous chemicals storage.

Key words: hazardous chemicals warehouse; layout optimization; NP-hard; discrete particle swarm