



(12) 发明专利

(10) 授权公告号 CN 109447317 B

(45) 授权公告日 2021. 12. 21

(21) 申请号 201811108726.1

CN 103413209 A, 2013.11.27

(22) 申请日 2018.09.21

孙慧. 配送中心拣货路径优化研究. 《中国优秀硕士学位论文全文数据库 信息科技辑》. 2014, (第12期), 第2, 7-29页.

(65) 同一申请的已公布的文献号

申请公布号 CN 109447317 A

审查员 唐进岭

(43) 申请公布日 2019.03.08

(73) 专利权人 江苏大学

地址 212013 江苏省镇江市京口区学府路301号

(72) 发明人 薛安荣 张宇

(51) Int. Cl.

G06Q 10/04 (2012.01)

G06Q 10/08 (2012.01)

G06N 3/12 (2006.01)

(56) 对比文件

CN 101968860 A, 2011.02.09

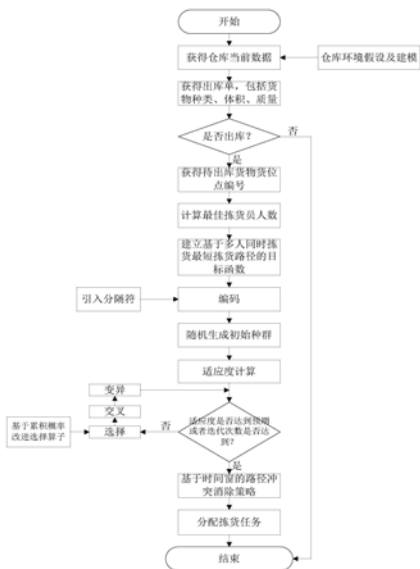
权利要求书4页 说明书10页 附图3页

(54) 发明名称

一种基于多人同时拣货的仓库最短路径优化方法

(57) 摘要

本发明提出了一种基于多人同时拣货的仓库最短路径优化方法。以基于多人同时拣货的最短拣货路径为优化目标,对仓库作业环境进行假设并建立数学模型,依据出库商品信息确定所需拣货人数;在遗传算法的遗传操作中引入分隔符的概念,结合基于累积概率改进的选择算子实现对算法的改进并求解目标函数,获得多人同时拣货的最优拣货路径集合;采用基于时间窗的拣货路径冲突消除策略对获得的路径集合进行规划,采取考虑优先级的等待策略消除路径冲突,获得每条路径的拣货时间;最后比较每条路径的拣货时间,选择用时最短的路径作为优化结果。本发明能提高仓库的拣货效率。



1. 一种基于多人同时拣货的仓库最短路径优化方法,其特征在于,包括:

第一,对仓库作业环境进行假设并建立数学模型;第二,考虑基于多人同时拣货的最短路径优化目标,根据出库单确定所需拣货员人数,同时以总拣货路径最短为衡量标准建立目标函数;第三,在遗传算法的遗传操作中引入分隔符的概念并结合基于累积概率改进的选择算子实现对算法的改进,以此对目标函数求解,获得多人同时拣货的最优拣货路径集合;第四,采用基于时间窗的拣货路径冲突消除策略对获得的路径集合进行规划,当存在路径冲突时采取考虑优先级的等待策略,获得每条路径的拣货时间;第五,对每条路径的拣货时间进行比较,采用用时最短的路径作为优化结果;

建立基于多人同时拣货的最短路径优化目标函数,包括以下步骤:

步骤2.1,根据出库单确定所需拣货员人数,包括以下步骤:

步骤2.1.1,获得出库单,包括货物种类、重量和体积;

步骤2.1.2,依据货物种类、重量和体积信息获得待出库货物所在的货位点编号;

步骤2.1.3,确定所需拣货员人数:令  $K_1 = \left\lceil \sum_{i=1}^N m_i / M \right\rceil$ ,  $K_2 = \left\lceil \sum_{i=1}^N v_i / V \right\rceil$ , 其中  $m_i$ 、 $v_i$  分别

表示每个货位点待拣货物的重量和体积,  $M$ 、 $V$  分别表示每辆小推车的最大载重量和最大承载体积,  $N$  表示待拣货位点的总数量, 符号  $\lceil \cdot \rceil$  表示向上取整, 考虑到拣货工人费用的问题取较少的那一个作为拣货员人数  $K = \min \{K_1, K_2\}$ ;

步骤2.2,考虑所有拣货人员的总拣货路径最短,建立如下目标函数:

$$\min S = \sum_{k=1}^K \left[ \sum_{i=1}^{n_k} D_{r_{k(i-1)}r_{ki}} + D_{r_{kn_k}r_{k0}} \right]$$

其中,

$$D_{r_{k(i-1)}r_{ki}} = l |x_{r_{ki}} - x_{r_{k(i-1)}}| + (W_1 + W_2) |y_{r_{ki}} - y_{r_{k(i-1)}}|$$

$$D_{r_{kn_k}r_{k0}} = l |x_{r_{kn_k}} - x_{r_{k0}}| + (W_1 + W_2) |y_{r_{kn_k}} - y_{r_{k0}}|$$

其中,  $S$  为所有拣货员的总拣货路径,  $K$  为拣货员人数,  $n_k$  为第  $k$  个拣货员所需拣货的货位点数量,  $r_{ki}$  表示货位点在第  $k$  个拣货员的拣货顺序中为第  $i$  个,  $r_{k0}$  表示拣选中心,  $D_{r_{k(i-1)}r_{ki}}$  表示第  $k$  个拣货员所需拣选的第  $i$  个货位和第  $i-1$  个货位之间的距离,  $D_{r_{kn_k}r_{k0}}$  表示第  $k$  个拣货员拣选完  $n_k$  个货位点后返回拣选中心的距离,  $l$  表示货架长度,  $W_1$  表示货架宽度,  $W_2$  表示巷道宽度,  $x_{r_{ki}}$  表示货位点的横坐标,  $y_{r_{ki}}$  表示货位点的纵坐标, 且两者均为正整数, 其中  $(x_{r_{k0}}, y_{r_{k0}}) = (0, 0)$  表示拣选中心;

步骤2.3,约束条件为:

$$\sum_{i=1}^N x_{ij} = 1, (i \neq j, \forall j = 1, 2, \dots, N)$$

$$\sum_{j=1}^N x_{ij} = 1, (i \neq j, \forall i = 1, 2, \dots, N)$$

$$\sum_{i=1}^{n_k} m_i \leq M$$

$$\sum_{k=1}^K n_k \leq N$$

式中,  $x_{ij}$  为决策变量, 且  $x_{ij} = \{0, 1\}$ , 若拣货作业工人在拣货时由货位点  $i$  拣货完成后接着到达下一货位点  $j$ , 则  $x_{ij} = 1$ , 否则,  $x_{ij} = 0$ ,  $\sum_{i=1} m_i$  表示拣货作业进行到货位点  $i$  时, 当前拣货车上已拣取货品的总重量。

2. 根据权利要求1所述的一种基于多人同时拣货的仓库最短路径优化方法, 其特征在于, 所述对仓库作业环境进行假设并建立数学模型, 包括以下步骤:

步骤1.1, 对所述拣货人员及其作业的仓库环境进行分析, 做出以下假设:

1) 所述仓库划分为拣货中心与仓储区, 其中仓储区负责存放货物, 当有出库任务时, 拣货人员首先把货物从仓储区拣选至拣货中心, 在拣货中心进行货物清点后再出库; 当没有出库任务时, 拣货人员位于拣货中心待命;

2) 所述仓储区的货架采用  $x$  排  $y$  列  $z$  层的摆放方式, 货架与货架之间等距, 每个货位大小相同, 且每个货位的承重能力相同;

3) 所述拣货人员在仓储区进行拣货作业时, 默认在各拣货点所需的拣货时间相同, 因此不作考虑;

4) 每辆小推车所承载的货物重量不得超过其最大载重量;

步骤1.2, 对多人同时拣货所处的作业空间进行数学描述如下:

1) 对每个货位进行自然数编号, 编号从  $1, 2, 3, \dots$ , 直到最后一个货位, 其中1号货位表示第1排第1列第1层位置上的货位, 2号货位表示第1排第2列第1层位置上的货位, 以此类推, 拣货中心编号为0;

2) 将仓库投影到平面直角坐标系, 每个货位对应坐标系上的一个坐标点, 拣货员及小推车在二维平面环境中运动。

3. 根据权利要求1所述的一种基于多人同时拣货的仓库最短路径优化方法, 其特征在于, 第三步具体包括以下步骤:

步骤3.1, 编码: 采用实数编码对个体进行编码, 为了使得编码个体符合多人同时拣货的条件, 在编码中引入0作为分隔符, 用来区分不同拣货人员的拣货路径, 同时0也表示拣货区,  $1, 2, 3, \dots, L$  表示各个货位点, 则拣货路径可编码为  $(0, 1, 2, 3, 4, 0, 5, 6, 7, 8, 0, 9, \dots, L, 0)$ , 其中0的数量为  $K+1$  个,  $(0, 1, 2, 3, 4, 0)$  表示第一个拣货员的拣货路径为从拣货区到第1个货位点再到第2个货位点, 再到第3个货位点, 以此类推;

步骤3.2, 初始化种群: 按照种群数目随机生成  $Q$  个个体;

步骤3.3, 适应度函数计算: 选择适应度函数, 对于每个个体计算适应度函数值;

步骤3.4, 选择: 采用基于累积概率改进的选择算子;

步骤3.5, 交叉: 采用常用的两点交叉算子, 在相互配对的两个个体编码串中随机设置两个交叉点, 然后交换两个交叉点之间的元素, 交叉的目的是为了产生新个体, 即产生新的拣选路径, 提高种群的多样性;

步骤3.6, 变异: 采用小范围变异, 依据概率  $p_m$  随机选择除0元素以外的元素进行变异操

作,变异也是为了产生新的货物拣选路径;

步骤3.7,重复步骤3.4~步骤3.6,直到适应度达到预期或者达到迭代次数,并输出最优的种群集合。

4.根据权利要求3所述的一种基于多人同时拣货的仓库最短路径优化方法,其特征在于,采用基于累积概率改进的选择算子的步骤如下:

步骤3.4.1,计算每个个体的适应度值,记为 $T_q$ ;

步骤3.4.2,根据适应度值计算出每个个体被选中的概率;

步骤3.4.3,如果产生的新种群中最优解的适应度值小于上一代的最优解的适应度值,则上一代的最优解替换新的种群中适应度值最差的个体,以此实现选择概率的累积。

5.根据权利要求3所述的一种基于多人同时拣货的仓库最短路径优化方法,其特征在于,交叉和变异操作进行以下的条件约束,以保证产生的新个体的合法性:

如果交叉生成的新个体的各个元素中除了0以外的元素有相同的元素,则从待出库货物货位点集合中随机选择一个数字替换该元素,直到所有除了0以外的元素均是合法的待出库货物的货位点为止;

如果交叉生成的新个体中0元素的数目超过 $K+1$ 个,则在新个体中随机选择除第一个和最后一个以外的0元素与配对生成的个体中的非0元素进行互换,直到两个新个体中的0元素数目均为 $K+1$ 个。

6.根据权利要求1所述的一种基于多人同时拣货的仓库最短路径优化方法,其特征在于,所述第四步的具体步骤如下:

步骤4.1,对于每条拣货路径,首先考察第1个拣货员的拣货路径,将该路径经过的各单位路径时间窗计算出来,初始化各单位路径时间窗集合 $F(e_k)$ ;

步骤4.2,考察当前路径剩余拣货员的拣货路径,获得剩余拣货员要经过的边的集合 $E$ ;

步骤4.3,依照拣货员的优先级顺次考察集合 $E$ 中的项,判断是否存在空闲时间窗,若是,则在该路径上不存在冲突,拣货员可以顺利完成相关货位点的拣货任务,同时更新该边的时间窗集合 $F(e_k)$ ;若否,执行步骤4.4;

步骤4.4,此时拣货员在该路径上存在冲突,采取考虑优先级的等待策略,当拣货员到达冲突路径时,优先级低的拣货员等待优先级高的拣货员先通过,冲突消失后继续通行,同时将优先级较低的拣货员在此边的时间窗后移,更新该边的时间窗集合 $F(e_k)$ ;

步骤4.5,重复步骤4.3~步骤4.4,直到集合 $E$ 中的所有项均考察完为止,输出当前路径的最终拣货时间;

步骤4.6,对于最优拣货路径集合中的每条路径均做上述考察,获得该集合中每条路径的最终拣货时间。

7.根据权利要求1所述的一种基于多人同时拣货的仓库最短路径优化方法,其特征在于,所述第五步包括以下步骤:

步骤5.1,比较最优种群中各条路径的最终拣货时间,采用用时最短的拣货路径作为优化结果;

步骤5.2,对当前路径进行解码,其中第1个0和第2个0之间的数字表示第1个拣货人员的拣货路径,第2个0和第3个0之间的数字表示第2个拣货人员的拣货路径,以此类推,则第 $K$ 个0和第 $K+1$ 个0之间的数字表示第 $K$ 个拣货人员的拣货路径,其中0与0之间的数字表示货位

点,0表示拣货区。

## 一种基于多人同时拣货的仓库最短路径优化方法

### 技术领域

[0001] 本发明涉及最短路径与智能仓储管理技术领域,特别是涉及一种基于多人同时拣货的仓库最短路径优化方法。

### 背景技术

[0002] 近年来,随着计算机技术等先进技术的飞速发展,对于仓库管理系统(Warehouse management system,WMS)的研究也如火如荼,在这个过程中,货物拣选的最短路径优化问题一直是研究重点。传统的对于货物拣选的最短路径优化问题大多采用蚁群算法、遗传算法、粒子群优化算法和模拟退火算法等进行求解,但这些研究通常是集中于单人拣货的最短路径规划问题,不能满足仓库内会存在多人同时拣货的实际情况,具有一定的局限性。随着需要拣选货物的增多,拣货人员的需求也随之增多,当前的算法并不能直接应用于解决多人同时拣货的最短路径优化问题,同时也不能解决多人同时拣货时存在的拣货员的任务分配问题和路径冲突问题。如何确定最佳拣货人数、获得最短拣货路径以及解决多人同时拣货的路径冲突问题是多人同时拣货路径规划的研究重点。

[0003] 因此,需要一种基于多人同时拣货的最短路径优化算法,首先可以满足仓库中需要多个拣货员同时拣货的实际情况,其次需要对传统的遗传算法进行改进,以适用于求解多人同时拣货的最短路径优化目标函数,最后采用基于时间窗的拣货路径冲突消除策略,消除了多人同时拣货可能存在的路径冲突问题,提高了仓库的拣货效率。

### 发明内容

[0004] 本发明的目的在于提出一种基于多人同时拣货的仓库最短路径优化方法,针对现有的算法仅考虑了单人单线路拣货路径规划问题的情况,提出一种适用于多人同时拣货的优化算法,考虑基于多人同时拣货的最短路径优化目标,根据出库单确定所需拣货员人数,并以拣货路径最短为衡量标准建立目标函数;在遗传算法的遗传操作中引入分隔符的概念并结合基于累积概率改进的选择算子实现算法的改进,以解决当前算法并不能直接应用于求解基于多人同时拣货的最短路径优化目标函数的问题;采用基于时间窗的拣货路径冲突消除策略,消除了多人同时拣货路径规划中可能存在的路径冲突,以此获得多人同时拣货用时最短的拣货路径,具体技术方案如下:

[0005] 一种基于多人同时拣货的仓库最短路径优化方法,包括:

[0006] 第一,对仓库作业环境进行假设并建立数学模型;第二,考虑基于多人同时拣货的最短路径优化目标,根据出库单确定所需拣货员人数,同时以总拣货路径最短为衡量标准建立目标函数;第三,在遗传算法的遗传操作中引入分隔符的概念并结合基于累积概率改进的选择算子实现对算法的改进,以此对目标函数求解,获得多人同时拣货的最优拣货路径集合;第四,采用基于时间窗的拣货路径冲突消除策略对获得的路径集合进行规划,当存在路径冲突时采取考虑优先级的等待策略,获得每条路径的拣货时间;第五,对每条路径的拣货时间进行比较,采用用时最短的路径作为优化结果。

[0007] 进一步地,所述对仓库作业环境进行假设并建立数学模型,包括以下步骤:

[0008] 步骤1,对所述拣货人员及其作业的仓库环境进行分析,做出以下假设:

[0009] 1)所述仓库划分为拣货中心与仓储区,其中仓储区负责存放货物,当有出库任务时,拣货人员首先把货物从仓储区拣选至拣货中心,在拣货中心进行货物清点后再出库;当没有出库任务时,拣货人员位于拣货中心待命;

[0010] 2)所述仓储区的货架采用x排y列z层的摆放方式,货架与货架之间等距,每个货位大小相同,且每个货位的承重能力相同;

[0011] 3)所述拣货人员在仓储区进行拣货作业时,默认在各拣货点所需的拣货时间相同,因此不作考虑;

[0012] 4)每辆小推车所承载的货物重量不得超过其最大载重量。

[0013] 步骤2,对多人同时拣货所处的作业空间进行数学描述如下:

[0014] 1)对每个货位进行自然数编号,编号从1,2,3,...,直到最后一个货位,其中1号货位表示第1排第1列第1层位置上的货位,2号货位表示第1排第2列第1层位置上的货位,以此类推,其中拣货中心编号为0;

[0015] 2)将仓库投影到平面直角坐标系,每个货位对应坐标系上的一个坐标点,拣货员及小推车在二维平面环境中运动。

[0016] 进一步地,所述建立基于多人同时拣货的最短路径目标优化函数,包括以下步骤:

[0017] 步骤1,根据出库单确定所需拣货员人数,包括以下步骤:

[0018] 步骤1.1,获得出库单,包括货物种类、重量和体积;

[0019] 步骤1.2,依据货物种类、重量及体积获得待出库货物所在的货位点编号;

[0020] 步骤1.3,确定所需拣货员人数:令  $K_1 = \left\lceil \sum_{i=1}^N m_i / M \right\rceil$ ,  $K_2 = \left\lceil \sum_{i=1}^N v_i / V \right\rceil$ , 其中  $m_i$ 、 $v_i$  分

别表示每个货位点待拣货物的重量和体积,  $M$ 、 $V$  分别表示每辆小推车的最大载重量和最大承载体积,  $N$  表示待拣货位点的总数量, 符号  $\lceil \cdot \rceil$  表示向上取整。考虑到拣货工人费用的问题取较少的那一个作为拣货员人数  $K = \min \{K_1, K_2\}$ 。

[0021] 步骤2,考虑所有拣货人员的总拣货路径最短,建立如下目标函数:

$$[0022] \quad \min S = \sum_{k=1}^K \left[ \sum_{i=1}^{n_k} D_{r_{k(i-1)}r_{ki}} + D_{r_{k0}r_{k0}} \right] \quad (1)$$

[0023] 其中,

$$[0024] \quad D_{r_{k(i-1)}r_{ki}} = l |x_{r_{ki}} - x_{r_{k(i-1)}}| + (W_1 + W_2) |y_{r_{ki}} - y_{r_{k(i-1)}}| \quad (2)$$

$$[0025] \quad D_{r_{k0}r_{k0}} = l |x_{r_{k0}} - x_{r_{k0}}| + (W_1 + W_2) |y_{r_{k0}} - y_{r_{k0}}| \quad (3)$$

[0026] 其中,  $S$  为所有拣货员的总拣货路径,  $K$  为拣货员人数,  $n_k$  为第  $k$  个拣货员所需拣货的货位点数量,  $r_{ki}$  表示货位点在第  $k$  个拣货员的拣货顺序中为第  $i$  个,  $r_{k0}$  表示拣选中心,



$D_{r_{k(i-1)}r_{ki}}$  表示第k个拣货员所需拣选的第i个货位和第i-1个货位之间的距离,  $D_{r_{knk}r_{k0}}$  表示第k个拣货员拣选完 $n_k$ 个货位点后返回拣选中心的距离,  $l$ 表示货架长度,  $w_1$ 表示货架宽度,  $w_2$ 表示巷道宽度,  $x_{r_{ki}}$ 表示货位点的横坐标,  $y_{r_{ki}}$ 表示货位点的纵坐标, 且两者均为正整数, 其中  $(x_{r_{k0}}, y_{r_{k0}}) = (0, 0)$  表示拣选中心;

[0027] 步骤3, 约束条件为:

$$[0028] \quad \sum_{i=1}^N x_{ij} = 1, (i \neq j, \forall j = 1, 2, \dots, N) \quad (4)$$

$$[0029] \quad \sum_{j=1}^N x_{ij} = 1, (i \neq j, \forall i = 1, 2, \dots, N) \quad (5)$$

$$[0030] \quad \sum_{i=1}^{n_k} m_i \leq M \quad (6)$$

$$[0031] \quad \sum_{k=1}^K n_k \leq N \quad (7)$$

[0032] 式中,  $x_{ij}$  为决策变量, 且  $x_{ij} = \{0, 1\}$ 。若拣货作业工人在拣货时由货位点i拣货完成后接着到达下一货位点j, 则  $x_{ij} = 1$ , 否则,  $x_{ij} = 0$ 。 $\sum_{i=1} m_i$  表示拣货作业进行到货位点i时, 当前拣货车上已拣取货品的总重量。式(4)、(5)对拣货作业共同起作用, 表示从任一货位点i到达另一货位点j距离最短的路线有且只有一条, 拣货时只能走这条最短路线而不得走其它路线。式(6)限制拣货车上已拣货品的总重量不得超过拣货车的核定车载量。式(7)表示拣货员的拣货数量不得超过需要拣货的总拣货数量。

[0033] 进一步地, 所述遗传算法及其改进操作包括以下步骤:

[0034] 步骤1, 编码: 采用实数编码对个体进行编码, 为了使得编码个体符合多人同时拣货的条件, 在编码中引入0作为分隔符, 用来区分不同拣货人员的拣货路径, 同时0也表示拣货区,  $1, 2, 3, \dots, L$  表示各个货位点, 则拣货路径可编码为  $(0, 1, 2, 3, 4, 0, 5, 6, 7, 8, 0, 9, \dots, L, 0)$ , 其中0的数量为K+1个,  $(0, 1, 2, 3, 4, 0)$  表示第一个拣货员的拣货路径为从拣货区到第1个货位点再到第2个货位点, 再到第3个货位点, 以此类推;

[0035] 步骤2, 初始化种群: 按照种群数目随机生成Q个个体;

[0036] 步骤3, 适应度函数计算: 选择公式(1)作为适应度函数, 对于每个个体计算适应度函数值;

[0037] 步骤4, 选择: 采用基于累积概率改进的选择算子, 步骤如下:

[0038] 步骤4.1, 计算每个个体的适应度值, 记为  $T_q$ ;

[0039] 步骤4.2, 根据适应度值计算出每个个体被选中的概率, 计算公式如下

$$[0040] \quad P_{select}(q) = \frac{1}{f(T_q)} = \frac{1}{T_q} \quad (8)$$

$$\sum_{q=1}^Q T_q$$

[0041] 其中  $P_{select}(q)$  表示个体q被选择的概率,  $f(T_q)$  表示每个个体的适应度值与所有个



体总体适应度值的比,  $P_{\text{select}}(q)$  越大, 则个体被选中进入下一代的概率就越大, 根据概率随机选择进入下一代的个体;

[0042] 步骤4.3, 如果产生的新种群的最优解的适应度值小于上一代的最优解的适应度值, 则上一代的最优解替换新的种群中适应度值最差的个体, 以此来实现选择概率的累积, 这一步保证了优良个体均能被选择到新的种群中, 即用时较短的拣选路径更容易被选择到新种群中。

[0043] 步骤5, 交叉: 采用常用的两点交叉算子, 在相互配对的两个个体编码串中随机设置两个交叉点, 然后交换两个交叉点之间的元素, 交叉的目的是为了产生新个体, 即产生新的拣选路径, 提高种群的多样性;

[0044] 步骤6, 变异: 采用小范围变异, 依据概率  $p_m$  随机选择除0元素以外的元素进行变异操作, 变异也是为了产生新的货物拣选路径, 对路径中的不理想路段进行变异可以提高算法的效率, 使得遗传算法可以保持种群的多样性, 防止出现早熟现象;

[0045] 进一步地, 由于个体编码的特殊性, 本实施例对算法中的交叉和变异操作进行以下的条件约束, 以保证产生的新个体的合法性:

[0046] 1) 如果交叉生成的新个体的各个元素中除了0以外的元素有相同的元素, 则从待出库货物货位点集合中随机选择一个数字替换该元素, 直到所有除了0以外的元素均是合法的待出库货物的货位点为止;

[0047] 2) 如果交叉生成的新个体中0元素的数目超过  $K+1$  个, 则在新个体中随机选择除第一个和最后一个以外的0元素与配对生成的个体中的非0元素进行互换, 直到两个新个体中的0元素数目均为  $K+1$  个。

[0048] 步骤7, 重复步骤4~步骤6, 直到适应度达到预期或者达到迭代次数, 并输出最优的种群集合。

[0049] 进一步地, 当拣货员的拣货路径发生冲突时, 采用基于时间窗的拣货路径冲突消除策略, 具体步骤如下:

[0050] 步骤1, 对于每条拣货路径, 首先考察第1个拣货员的拣货路径, 将该路径经过的各单位路径时间窗计算出来, 初始化各单位路径时间窗集合  $F(e_k)$ ;

[0051] 步骤2, 考察当前路径剩余拣货员的拣货路径, 获得剩余拣货员要经过的边的集合  $E$ ;

[0052] 步骤3, 依照拣货员的优先级顺次考察集合  $E$  中的项, 判断是否存在空闲时间窗, 若是, 则在该路径上不存在冲突, 拣货员可以顺利完成相关货位点的拣货任务, 同时更新该边的时间窗集合  $F(e_k)$ ; 若否, 执行步骤4;

[0053] 进一步地, 按照拣货任务分配的先后顺序确定拣货员的优先级, 最先分配任务的拣货员优先级最高, 以此类推, 最后获得拣货任务的拣货员优先级最低。

[0054] 步骤4, 此时拣货员在该路径上存在冲突, 采取考虑优先级的等待策略, 当拣货员到达冲突路径时, 优先级低的拣货员等待优先级高的拣货员先通过, 冲突消失后继续通行, 同时将优先级较低的拣货员在此边的时间窗后移, 更新该边的时间窗集合  $F(e_k)$ ;

[0055] 步骤5, 重复步骤3~步骤4, 直到集合  $E$  中的所有项均考察完为止, 输出当前路径的最终拣货时间;

[0056] 步骤6, 对于最优拣货路径集合中的每条路径均做上述考察, 获得该集合中每条路

径的最终拣货时间。

[0057] 进一步地,比较路径集合中每条路径的拣货时间,采用用时最短的拣货路径作为优化结果,包括以下步骤:

[0058] 步骤1,比较最优种群中各条路径的最终拣货时间,采用用时最短的拣货路径作为优化结果;

[0059] 步骤2,对当前路径进行解码,其中第1个0和第2个0之间的数字表示第1个拣货人员的拣货路径,第2个0和第3个0之间的数字表示第2个拣货人员的拣货路径,以此类推,则第K个0和第K+1个0之间的数字表示第K个拣货人员的拣货路径,其中0与0之间的数字表示货位点,0表示拣货区。

[0060] 和现有技术相比,本发明的有益效果为:

[0061] (1) 本发明实施例提供的最短路径优化算法可以解决仓库内需要多个拣货员同时拣货的要求,更符合仓库管理的实际情况,且在一定程度上可以缩短货物拣选路径。

[0062] (2) 本发明采用改进的遗传算法来求解最短路径优化问题,对遗传算法的编码操作进行改进,在编码中引入0作为分隔符,以适应于求解多人同时拣货的最短路径优化目标函数,同时采用基于累积概率改进的选择算子,用父代种群中的最优解替换子代种群中适应度值最差的个体,以此来实现选择概率的累积,保证了后代个体的优良性。

[0063] (3) 算法中采用基于时间窗的拣货路径冲突消除策略,消除了多人同时拣货可能存在的路径冲突问题,在一定程度上提高了仓库的拣货效率。

## 附图说明

[0064] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例描述所需要使用的附图作简单地介绍。

[0065] 图1为本发明实施例提供的基于多人同时拣货的仓库最短路径优化算法流程图。

[0066] 图2为本发明实施例所描述的仓库模型在平面上的投影示意图。

[0067] 图3为时间窗优化前各拣货路径所用的时间。

[0068] 图4为采用基于时间窗的路径冲突消除策略后各拣货路径所用的时间。

## 具体实施方式

[0069] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述。

[0070] 以某仓库的某次出库任务为例,该仓库仓储区为10排10列3层,出库任务中包含12个货位点,12个货位点的坐标及每个货位点的出货量如表1所示,其中拣货区0也作为一个货位点:

[0071] 表1

[0072]

货位点i	$x_i$	$y_i$	$z_i$	出货量	货位点i	$x_i$	$y_i$	$z_i$	出货量
0	0	0	0	0	198	7	8	2	92kg
2	1	2	1	22kg	208	7	8	3	5kg
12	2	9	1	32kg	219	8	9	1	165kg
95	4	5	1	12kg	229	8	9	2	16kg

97	4	7	1	5kg	239	8	9	3	41kg
157	6	7	1	45kg	241	9	1	1	11kg
188	7	8	1	145kg					

[0073] 请参阅图1所示,本发明提供了一种基于多人同时拣货的仓库最短路径优化算法,包括五个步骤:第一,对仓库作业环境进行假设并建立数学模型;第二,考虑基于多人同时拣货的最短路径优化目标,根据出库单确定所需拣货员人数,同时以总拣货路径最短为衡量标准建立目标函数;第三,在遗传算法的遗传操作中引入分隔符的概念并结合基于累积概率改进的选择算子实现对算法的改进,以此对目标函数求解,获得多人同时拣货的最优拣货路径集合;第四,采用基于时间窗的拣货路径冲突消除策略对获得的路径集合进行规划,当存在路径冲突时采取考虑优先级的等待策略,获得每条路径的拣货时间;第五,对每条路径的拣货时间进行比较,采用用时最短的路径作为优化结果。

[0074] 一、仓库环境的假设及建模

[0075] 步骤1,对所述拣货人员及其作业的仓库环境进行分析,做出以下假设:

[0076] 1) 所述仓库划分为拣货中心与仓储区,其中仓储区负责存放货物,当有出库任务时,拣货人员首先把货物从仓储区拣选至拣货中心,在拣货中心进行货物清点后再出库;当没有出库任务时,拣货人员位于拣货中心待命;

[0077] 2) 所述仓储区的货架采用x排y列z层的摆放方式,货架与货架之间等距,每个货位大小相同,且每个货位的承重能力相同,该仓库仓储区为10排10列3层;

[0078] 3) 所述拣货人员在仓储区进行拣货作业时,默认在各拣货点所需的拣货时间相同,因此不作考虑;

[0079] 4) 每辆小推车所承载的货物重量不得超过其最大载重量,体积不得超过最大承载体积,在这里每辆小推车最大载重量为200kg。

[0080] 步骤2,对多人同时拣货所处的作业空间进行数学描述如下:

[0081] 1) 对每个货位进行自然数编号,编号从1,2,3,...,直到最后一个货位,其中1号货位表示第1排第1列第1层位置上的货位,2号货位表示第1排第2列第1层位置上的货位,以此类推,其中拣货中心编号为0;

[0082] 2) 将仓库投影到平面直角坐标系,每个货位对应坐标系上的一个坐标点,拣货员及小推车在二维平面环境中运动。

[0083] 二、建立基于多人同时拣货的最短路径目标优化函数,包括以下步骤:

[0084] 步骤1,根据出库单确定所需拣货员人数,包括以下步骤:

[0085] 步骤1.1,获得出库单,包括货物种类、重量及体积;

[0086] 步骤1.2,依据货物种类、重量及体积获得待出库货物所在的货位点编号;

[0087] 步骤1.3,确定所需拣货员人数:令  $K_1 = \left\lceil \sum_{i=1}^N m_i / M \right\rceil$ ,  $K_2 = \left\lceil \sum_{i=1}^N v_i / V \right\rceil$ , 其中  $m_i$ 、 $v_i$  分

别表示每个货位点待拣货物的重量和体积,  $M$ 、 $V$  分别表示每辆小推车的最大载重量和最大承载体积,  $N$  表示待拣货位点的总数量, 符号  $\lceil \cdot \rceil$  表示向上取整。考虑到拣货工人费用的问题取较少的那一个作为拣货员人数  $K = \min \{K_1, K_2\}$ 。在这里可以计算得到所需拣货员人数

$$K = \left\lceil \sum_{i=1}^N m_i / M \right\rceil = \left\lceil \frac{591}{200} \right\rceil = 3。$$

[0088] 步骤2,考虑所有拣货人员的总拣货路径最短,建立如下目标函数:

$$[0089] \quad \min S = \sum_{k=1}^K \left[ \sum_{i=1}^{n_k} D_{r_{k(i-1)}r_{ki}} + D_{r_{kn_k}r_{k0}} \right] \quad (9)$$

[0090] 其中,

$$[0091] \quad D_{r_{k(i-1)}r_{ki}} = l |x_{r_{ki}} - x_{r_{k(i-1)}}| + (W_1 + W_2) |y_{r_{ki}} - y_{r_{k(i-1)}}| \quad (10)$$

$$[0092] \quad D_{r_{kn_k}r_{k0}} = l |x_{r_{kn_k}} - x_{r_{k0}}| + (W_1 + W_2) |y_{r_{kn_k}} - y_{r_{k0}}| \quad (11)$$

[0093] 其中,S为所有拣货员的总拣货路径,K为拣货员人数, $n_k$ 为第k个拣货员所需拣货的货位点数量, $r_{ki}$ 表示货位点在第k个拣货员的拣货顺序中为第i个, $r_{k0}$ 表示拣选中心, $D_{r_{k(i-1)}r_{ki}}$ 表示第k个拣货员所需拣选的第i个货位和第i-1个货位之间的距离, $D_{r_{kn_k}r_{k0}}$ 表示第k个拣货员拣选完 $n_k$ 个货位点后返回拣选中心的距离,l表示货架长度, $W_1$ 表示货架宽度, $W_2$ 表示巷道宽度, $x_{r_{ki}}$ 表示货位点的横坐标, $y_{r_{ki}}$ 表示货位点的纵坐标,且两者均为正整数,其中 $(x_{r_{k0}}, y_{r_{k0}}) = (0,0)$ 表示拣选中心;

[0094] 步骤3,约束条件为:

$$[0095] \quad \sum_{i=1}^N x_{ij} = 1, (i \neq j, \forall j = 1, 2, \dots, N) \quad (12)$$

$$[0096] \quad \sum_{j=1}^N x_{ij} = 1, (i \neq j, \forall i = 1, 2, \dots, N) \quad (13)$$

$$[0097] \quad \sum_{i=1}^{n_k} m_i \leq M \quad (14)$$

$$[0098] \quad \sum_{k=1}^K n_k \leq N \quad (15)$$

[0099] 式中, $x_{ij}$ 为决策变量,且 $x_{ij} = \{0, 1\}$ 。若拣货作业工人在拣货时由货位点i拣货完成后接着到达下一货位点j,则 $x_{ij} = 1$ ,否则, $x_{ij} = 0$ 。 $\sum_{i=1} m_i$ 表示拣货作业进行到货位点i时,当前拣货车上已拣取货品的总重量。式(12)、(13)对拣货作业共同起作用,表示从任一货位点i到达另一货位点j距离最短的路线有且只有一条,拣货时只能走这条最短路线而不得走其它路线。式(14)限制拣货车上已拣货品的总重量不得超过拣货车的核定车载量。式(15)表示拣货员的拣货数量不得超过需要拣货的总拣货数量。

[0100] 三、遗传算法及其改进操作包括以下步骤:

[0101] 步骤1,编码:采用浮点数编码方式对个体进行编码,为了使得编码个体符合多人同时拣货的条件,在编码中引入0作为分隔符,用来区分不同拣货人员的拣货路径,同时0也

表示拣货区, 1, 2, 3, ..., L表示各个货位点, 则总体拣货路径可编码为 (0, 2, 12, 95, 97, 0, 157, 188, 198, 208, 0, 219, 229, 239, 241, 0), 其中0的数量为K+1个, 其中K为拣货员人数, 在这里K=3。(0, 2, 12, 95, 97, 0)表示第一个拣货员的拣货路径为从拣货区到第2个货位点再到第12个货位点, 再到第95个货位点, 以此类推;

[0102] 步骤2, 初始化种群: 按照种群数目随机生成Q个个体, 这里取Q=50;

[0103] 步骤3, 适应度函数计算: 选择公式 (9) 作为适应度函数, 对于每个个体计算适应度函数值;

[0104] 步骤4, 选择: 采用基于累积概率改进的选择算子, 步骤如下:

[0105] 步骤4.1, 计算每个个体的适应度值, 记为 $T_q$ ;

[0106] 步骤4.2, 根据适应度值计算出每个个体被选中的概率, 计算公式如下

$$P_{select}(q) = \frac{1}{f(T_q)} = \frac{1}{\sum_{q=1}^Q T_q} \quad (16)$$

[0107] 其中 $P_{select}(q)$ 表示个体q被选择的概率,  $f(T_q)$ 表示每个个体的适应度值与所有个体总体适应度值的比,  $P_{select}(q)$ 越大, 则个体被选中进入下一代的概率就越大, 根据概率随机选择进入下一代的个体;

[0109] 步骤4.3, 如果产生的新种群的最优解的适应度值小于上一代的最优解的适应度值, 则上一代的最优解替换新的种群中适应度值最差的个体, 以此来实现选择概率的累积, 这一步保证了优良个体均能被选择到新的种群中, 即用较短的拣选路径更容易在新种群中出现。

[0110] 步骤5, 交叉: 采用常用的两点交叉算子, 在相互配对的两个个体编码串中随机设置两个交叉点, 然后交换两个交叉点之间的元素, 交叉的目的是为了产生新个体, 即产生新的拣选路径, 以提高种群的多样性。假设有如下两条染色体, 则交叉操作及结果如下所示:

[0111] 
$$\begin{array}{l} 02, 12, 95, 97, 0, 157, 188, 198, 208, 0, 219, 229, 239, 241, 0 \\ 097, 95, 12, 2, 0, 208, 198, 188, 157, 0, 241, 239, 229, 219, 0 \end{array} \rightarrow \begin{array}{l} 0, 97, 12, 95, 97, 0, 157, 188, 198, 157, 0, 219, 229, 239, 241, 0 \\ 0, 2, 95, 12, 2, 0, 208, 198, 188, 208, 0, 241, 239, 229, 219, 0 \end{array}$$

[0112] 其中, 随机选择两个个体中的第2位和第10位作为两个交叉点进行交叉。

[0113] 步骤6, 变异: 采用小范围变异的方式对个体实施变异, 依据概率 $p_m$ 随机选择除0元素以外的元素进行变异操作,  $p_m$ 一般取值0.01~0.2, 变异的目的是与交叉一致, 也是为了产生新的货物拣选路径, 对路径中的不理想路段进行变异可以提高算法的效率, 使得遗传算法可以保持种群的多样性, 以防止出现早熟现象。 $p_m$ 一般取值0.01~0.2, 本实施例中取 $p_m=0.05$ 。对上述交叉后的个体进行变异操作, 结果如下:

[0114] 
$$\begin{array}{l} 0, 97, 12, 95, 97, 0, 157, 188, 157, 0, 219, 229, 239, 241, 0 \\ 0, 2, 95, 12, 2, 0, 208, 198, 188, 208, 0, 241, 239, 229, 219, 0 \end{array} \rightarrow \begin{array}{l} 0, 97, 12, 95, 97, 0, 157, 0, 198, 157, 0, 219, 229, 239, 241, 0 \\ 0, 2, 95, 12, 2, 0, 208, 198, 188, 208, 0, 241, 239, 229, 219, 0 \end{array}$$

[0115] 其中, 依概率选择个体1中的第8位和个体2中的第5位进行变异操作。

[0116] 进一步地, 由于个体编码的特殊性, 本实施例对算法中的交叉和变异操作进行以下的条件约束, 以保证产生的新个体的合法性:

[0117] 1) 如果交叉生成的新个体的各个元素中除了0以外的元素有相同的元素, 则从待出库货物货位点集合中随机选择一个数字替换该元素, 直到所有除了0以外的元素均是合

法的待出库货物的货位点为止；

[0118] 2) 如果交叉生成的新个体中0元素的数目超过K+1个,则在新个体中随机选择除第一个和最后一个以外的0元素与配对生成的个体中的非0元素进行互换,直到两个新个体中的0元素数目均为K+1个。

[0119] 依照约束条件对产生的新个体中的元素进行调整,结果如下:

$$\begin{array}{ccc} 0,97,12,95,97,0,157,0,198,157,0,219,229,239,241,0 & & 0,188,12,95,97,0,157,2,198,208,0,219,229,239,241,0 \\ 0,2,95,12,157,0,208,198,188,208,0,241,239,229,219,0 & \rightarrow & 0,2,95,12,157,0,97,198,188,208,0,241,239,229,219,0 \end{array}$$

[0120] 步骤7,重复步骤4~步骤6,直到适应度达到预期或者达到迭代次数,并输出最优的种群集合。

[0121] 三、对上一步获得的拣货路径集合采用基于时间窗的拣货路径冲突消除策略,计算每条拣货路径的最终拣货时间,具体步骤如下:

[0122] 步骤1,对于每条拣货路径,首先考察第1个拣货员的拣货路径,将该路径经过的各单位路径时间窗计算出来,初始化各单位路径时间窗集合 $F(e_k)$ ;

[0123] 步骤2,考察剩余拣货员的拣货路径,得到剩余拣货员要经过的边的集合E;

[0124] 进一步地,考察拣货路径(0,95,97,208,219,241,0,2,157,198,239,0,12,188,229,0),可以得到各拣货员要经过的边分别为 $(e_{12}, e_{29}, e_{89}, e_{78}, e_{67})$ 、 $(e_{29}, e_{89}, e_{78})$ 和 $(e_{45}, e_{47}, e_{78}, e_{89}, e_{90})$ ,该路径的各条边的时间窗如图3所示;

[0125] 步骤3,依照拣货员的优先级顺次考察集合E中的项,判断是否存在空闲时间窗,若是,则在该路径上不存在冲突,拣货员可以顺利完成相关货位点的拣货任务,同时更新该边的时间窗集合 $F(e_k)$ ;若否,执行步骤4;

[0126] 进一步地,按照拣货任务分配的先后顺序确定拣货员的优先级,最先分配任务的拣货员优先级最高,以此类推,最后获得拣货任务的拣货员优先级最低;

[0127] 进一步地,拣货员之间的冲突包括追击类冲突、节点冲突和相向冲突。对于追击类冲突,由于拣货员的速度相同,所以可以不予考虑。对于节点冲突,首先查询各拣货员当前所在边的时间窗,若时间窗结束时间不同,则不存在节点冲突;若结束时间相同,判断二者下一节点是否为同一节点,若非同一节点,同样不存在节点冲突;若二者结束时间相同且下一节点为相同节点,则可以判定存在节点冲突。对于该类冲突的解决,需要比较拣货员的优先级,优先级高的先行通过。对于相向冲突,只需检测拣货员在下一路段和本路段是否为时间窗交互占用,即可确定是否存在相向冲突。对于该类冲突采用时间窗口后移策略,即对于优先级较低的任务,顺次后移其各路径的时间窗,直到消除所有相向冲突为止;

[0128] 步骤4,此时拣货员在该路径上存在冲突,采取考虑优先级的等待策略,当拣货员到达冲突路径时,优先级低的拣货员等待优先级高的拣货员先通过,冲突消失后继续通行,同时将优先级较低的拣货员在此边的时间窗后移,更新该边的时间窗集合 $F(e_k)$ ;

[0129] 步骤5,重复步骤3~步骤4,直到集合E中的所有项均考察完为止,输出当前路径的最终拣货时间;

[0130] 进一步地,对上述路径规划后的各单位路径的时间窗如图4所示。可以看出,采取该策略后,在拣货员2的拣货用时延长的前提下,拣货员1与拣货员3拣货用时减少,总的拣货用时减少。

[0131] 步骤6,对于最优拣货路径集合中的每条路径均做上述考察,获得该集合中每条路

径的最终拣货时间。

[0132] 五、选择当前种群中耗时最小的路径作为优化结果,包括以下步骤:

[0133] 步骤1,比较最优种群中各条路径的最终拣货时间,采用用时最短的拣货路径作为优化结果;

[0134] 步骤2,对当前路径进行解码,其中第1个0和第2个0之间的数字表示第1个拣货人员的拣货路径,第2个0和第3个0之间的数字表示第2个拣货人员的拣货路径,以此类推,则第K个0和第K+1个0之间的数字表示第K个拣货人员的拣货路径,其中0与0之间的数字表示货位点,0表示拣货区。

[0135] 上述实例的最终优化结果为(0,2,157,198,239,0,12,188,229,0,95,97,208,219,241,0),各拣货人员的拣货路径以及所用时间如表2所示,完成该拣货任务的最终用时为20min:

[0136] 表2

[0137]

拣货员	拣货路径	拣货时间
1	0-2-157-198-239-0	20min
2	0-12-188-229-0	20min
3	0-95-97-208-219-241-0	20min

[0138] 综上,本发明提出了一种基于多人同时拣货的仓库最短路径优化算法。以基于多人同时拣货的最短拣货路径为优化目标,对仓库作业环境进行假设并建立数学模型,依据出库商品信息确定所需拣货人数;在遗传算法的遗传操作中引入分隔符的概念,结合基于累积概率改进的选择算子实现对算法的改进并求解目标函数,获得多人同时拣货的最优拣货路径集合;采用基于时间窗的拣货路径冲突消除策略对获得的路径集合进行规划,采取考虑优先级的等待策略消除路径冲突,获得每条路径的拣货时间;最后比较每条路径的拣货时间,选择用时最短的路径作为优化结果。本发明充分考虑了现有仓库拣货路径优化算法中仅考虑单人拣货路径的局限性,针对仓库拣货中会出现多人同时拣货的情况,以基于多人同时拣货的最短路径作为优化目标,并改进遗传算法求解目标函数;考虑多人同时拣货可能会存在的路径冲突问题,采用基于时间窗的冲突消除策略,获得用时最短的拣货路径。本发明能提高仓库的拣货效率。



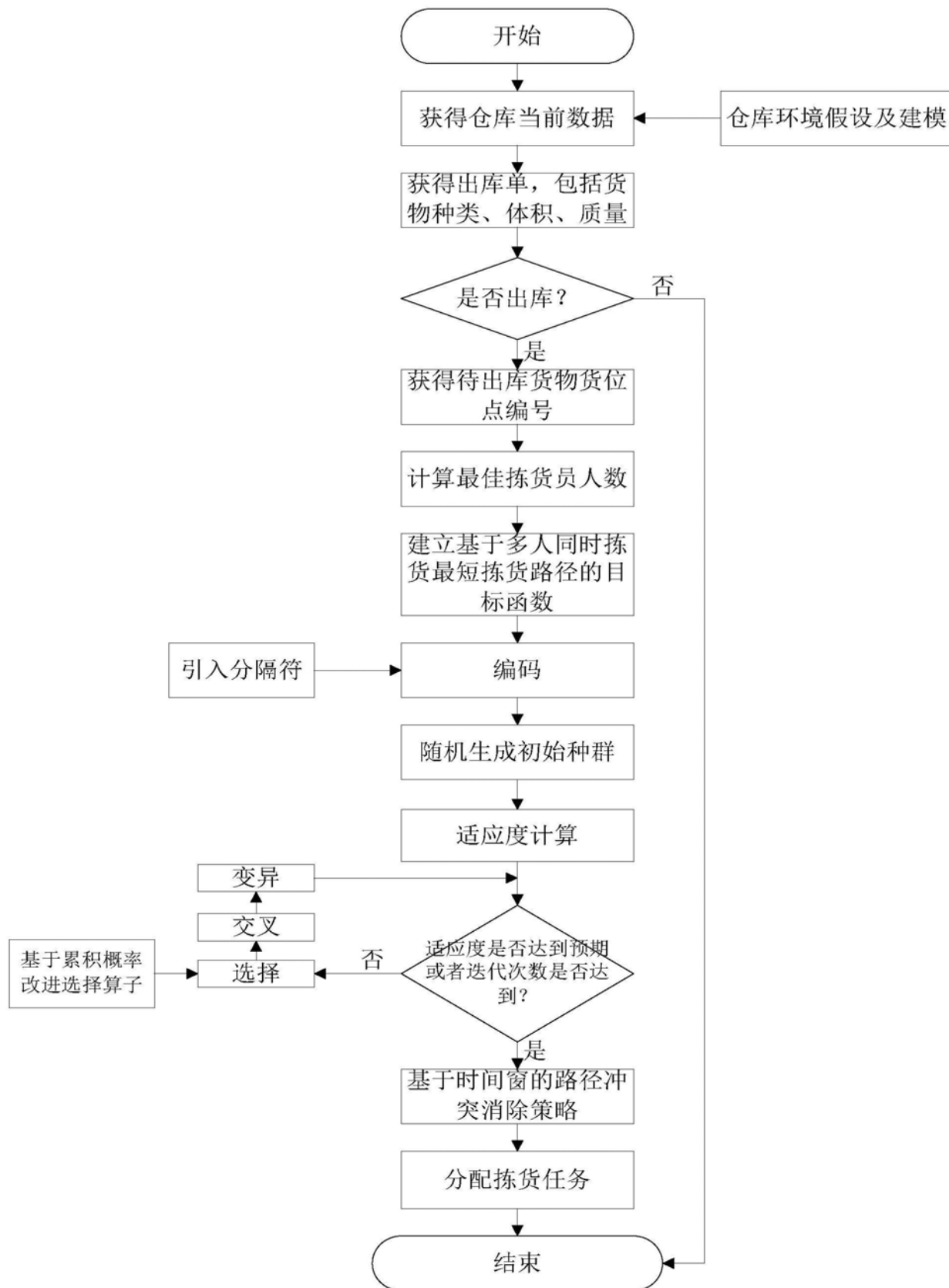


图1

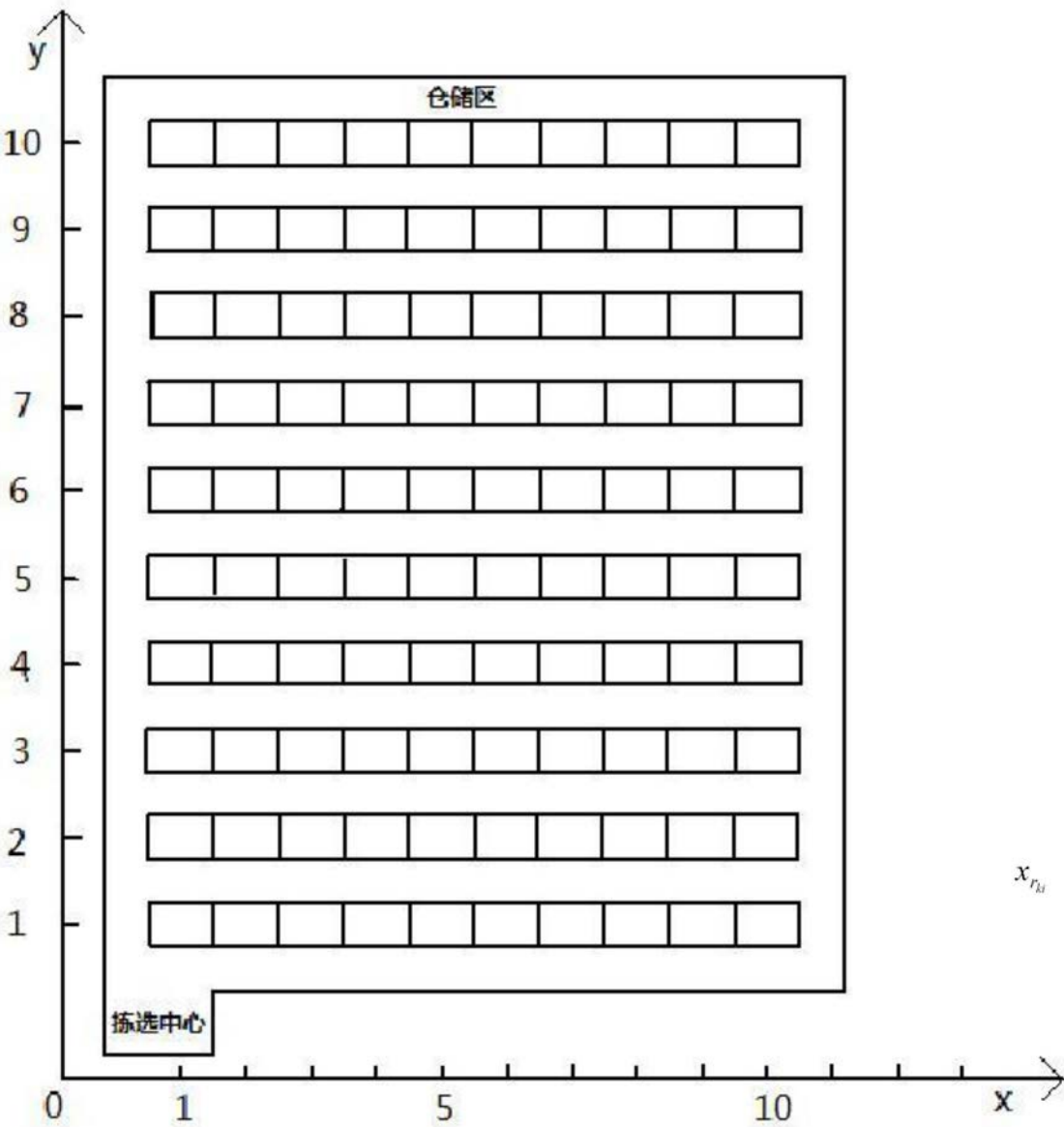


图2

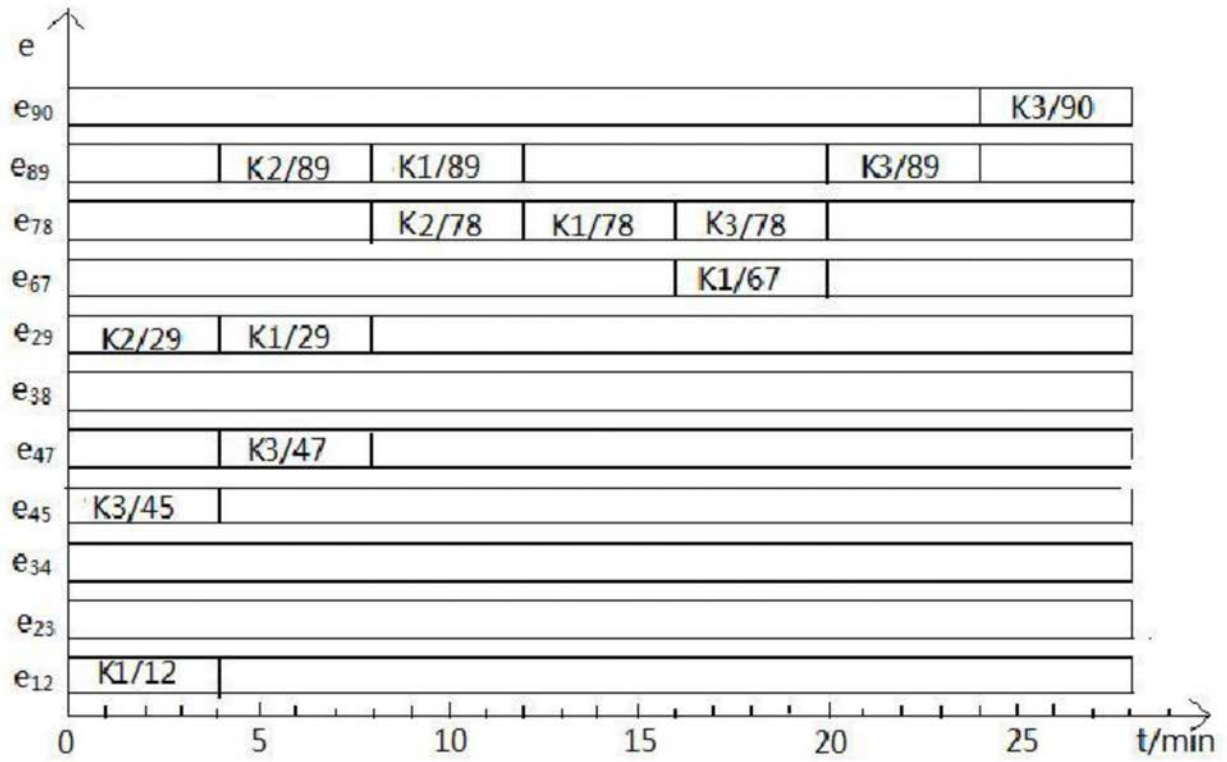


图3

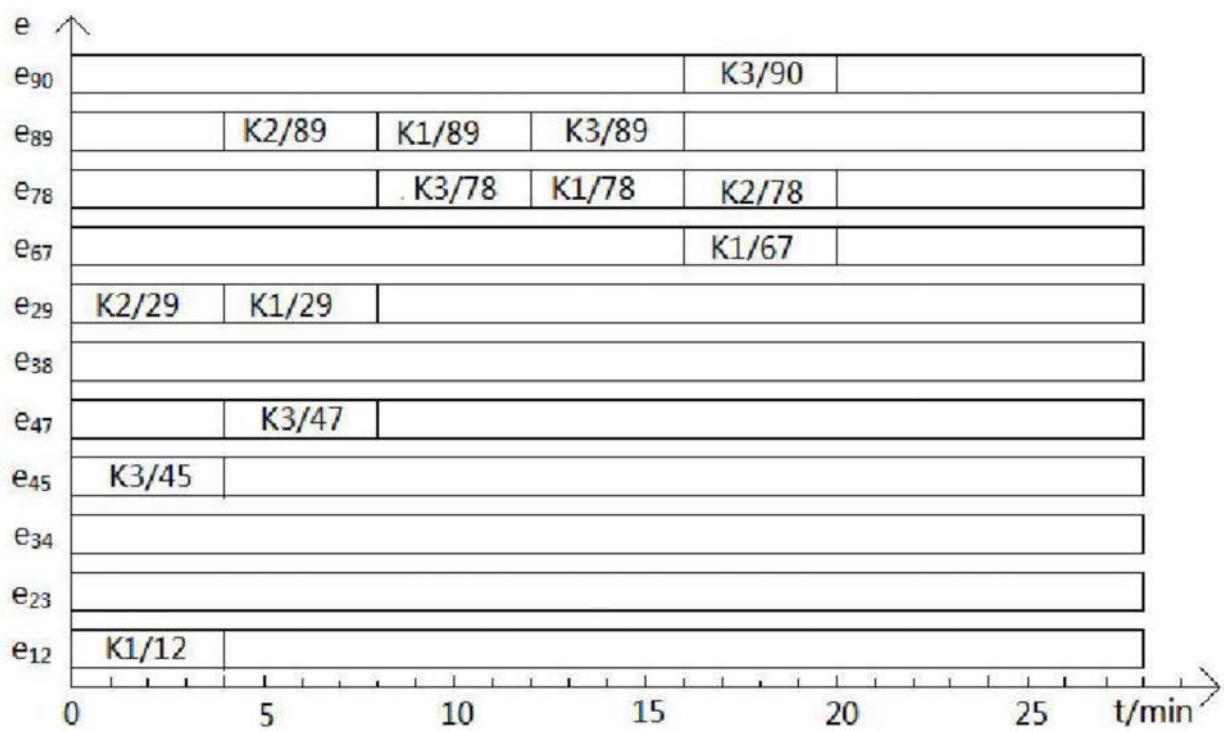


图4