

智能包装技术

## 双区型仓库动态拣货策略的设计及路径优化研究

孙军艳, 牛亚儒, 苏宝, 张媛媛  
(陕西科技大学 机电工程学院, 西安 710021)

**摘要:**目的 针对双区型仓库,以拣货时间最短为目标函数构建数学模型,进一步提高拣货效率。方法 提出并设计动态货位调整与人工拣货协同作业的动态拣货策略,分别采用 GA 算法和 GASA 算法进行最优化求解。结果 GASA 算法优于 GA 算法,拣货单为 1 张情况下的拣货时间可减少 4%;与静态拣货策略相比,拣货单为 10 张情况下,采用 GASA 算法时,文中设计动态拣货策略下的拣货时间可减少 6%,且随着拣货单数量的增加,拣货时间节约占比越大。结论 GASA 算法较 GA 算法其求解动态拣货路径优化问题更高效、优化结果更好。文中所提动态拣货策略更方便实施,在静态拣货路径优化基础上,可进一步提高拣货效率,且拣货单越多,效果就越显著。

**关键词:** 双区型仓库; 动态拣货; 路径优化; 混合遗传模拟退火算法

**中图分类号:** F252;TP391.9 **文献标识码:** A **文章编号:** 1001-3563(2018)23-0001-08

**DOI:** 10.19554/j.cnki.1001-3563.2018.23.001

### Dynamic Picking Strategy Design and Path Optimization of Two-block Warehouse

SUN Jun-yan, NIU Ya-ru, SU Bao, ZHANG Yuan-yuan

(College of Mechanical & Electrical Engineering, Shaanxi University of Science & Technology, Xi'an 710021, China)

**ABSTRACT:** The work aims to construct the mathematical model for the two-block warehouse with the shortest picking time as the objective function to further improve the picking efficiency. A dynamic picking strategy for dynamic cargo space adjustment and manual picking was proposed and designed. The GA algorithm and the GASA algorithm were respectively used to optimize the solution. The GASA algorithm was better than the GA algorithm. The picking time could be reduced by 4% in the case of one picking order. Compared with the static picking strategy, in the case of 10 picking orders, when the GASA algorithm was used, the picking time under the dynamic picking strategy designed herein could be reduced by 6%, and as the number of picking orders increased, more picking time would be saved. The GASA algorithm is more efficient and better optimized than the GA algorithm for solving the problem of dynamic picking path optimization. The proposed dynamic picking strategy can be implemented more conveniently. On the basis of static picking path optimization, the picking efficiency can be further improved, and the more the picking orders, the more significant the effect.

**KEY WORDS:** two-block warehouse; dynamic picking; path optimization; hybrid genetic simulated annealing algorithm

拣货是仓库作业的核心环节,对于劳动密集型仓库,与拣货作业直接相关的人力占 50%,拣货时间大约占整个仓库作业时间的 30%~40%,拣货成本占配送中心物流总成本的 60%左右<sup>[1]</sup>,因此优化拣货路径、缩短拣货距离、减少拣货时间对于提高配送中心

的运行效率、降低成本具有重要的意义。

Caron F<sup>[2]</sup>等研究了巷道的数量和不同拣选作业策略对拣选路径的影响,研究表明,配送中心内横向过道的存在能够提高拣选作业的效率,但是横向过道不能太多,否则会起到相反的效果,2~3 条横向过道

收稿日期: 2018-09-11

基金项目: 国家自然科学基金(51275407, 51475363, 11072192); 陕西省工业科技攻关项目(2018GY-026); 陕西科技大学博士科研启动基金(2018BJ-12); 国家级大学生创新创业训练计划(12145)

作者简介: 孙军艳(1978—),女,陕西科技大学副教授,主要研究方向为物流信息技术、供应链管理。

为最佳。Roodbergen 和 Koster<sup>[3]</sup>采用启发式算法求解最优拣货路径,并在不同规模的仓库中比较平均拣货时间,得出双区型仓库的平均拣货时间最短。作为一种通用高效的仓储模式,双区型仓库在配送中心内得到了广泛的应用<sup>[4]</sup>,国内外针对其拣货路径优化已有较多的研究成果。Osman Kulak<sup>[5]</sup>等将订单分批与拣选路径优化相结合,采用基于聚类的禁忌搜索算法求解最佳拣选路径。李建斌、周玮等<sup>[6]</sup>基于 TSP 对双区型仓库的拣货路径进行建模,利用蚁群算法、模拟退火算法和禁忌搜索对模型进行求解,得出在订单量较大时蚁群算法最优。李栋栋<sup>[7]</sup>研究了双区型仓库在车载能力有限的条件下,以最小化拣货车行走路程为目标建立模型,采用贪心遗传算法来求解模型,得出该算法可以有效地减少拣货路径。

目前,关于静态拣货路径优化的研究目前比较普遍,该方法适用于订单不大、拣货量不大的情况<sup>[8-10]</sup>。随着电商发展面临的订单量急剧增大情况,使得拣货作业面临着巨大的压力。在静态拣货方式的基础上,为了进一步提高拣货效率,边进行货位调整边拣货的动态拣货思想应运而生。杜颖<sup>[11]</sup>对订单分拣作业中货位指派和路径优化进行了研究,设计了一种动态货位指派作业方式,基于模拟退火算法对模型进行了求解,结果表明“货位指派与拣货”协同作业与分开作业相比作业时间更短,可明显提高拣货效率。徐翔斌等<sup>[12]</sup>将传统的货位指派分割为多阶段、小规模动态调整过程,设计了前进和返回交替进行的启发式路径策略,基于动态出货量轮盘赌算法来计算候选商品被选择的概率,在拣货过程中进行调整,但该研究设定的前提是在固定路径下进行调整,存在一定的局限性。

目前针对动态拣货策略的设计和实现方式的研究还较少,是拣货路径优化的一个新研究方向。文中针对双区型多巷道仓库,设计一种简单方便、易于操作实施,能够进一步提高拣货效率的拣货作业和货位动态调整协同作业的策略,以拣货时间最短为目标建立模型,采用 GA 算法和 GASA 算法,对比分析静态拣货策略和所提出的动态拣货策略。文中研究拟为进一步优化拣货路径、提高拣货效率提供一种思路。

### 1 双区型仓库简介

双区型仓库是由 3 条平行的横向通道和若干条纵向通道构成,每条通道的宽度相等,每条巷道宽度也相等<sup>[13-14]</sup>,纵向巷道除了第 1 列和最后 1 列的货架是单排货架,其它货架均为背靠背式货架,见图 1。

关于仓库尺寸的符号说明如下所述。

- 1)  $d_{ij}$  表示任意 2 个货位之间的距离;
- 2)  $a$  表示货位所在的巷道编号,  $a=1, 2, 3 \dots A$ ;

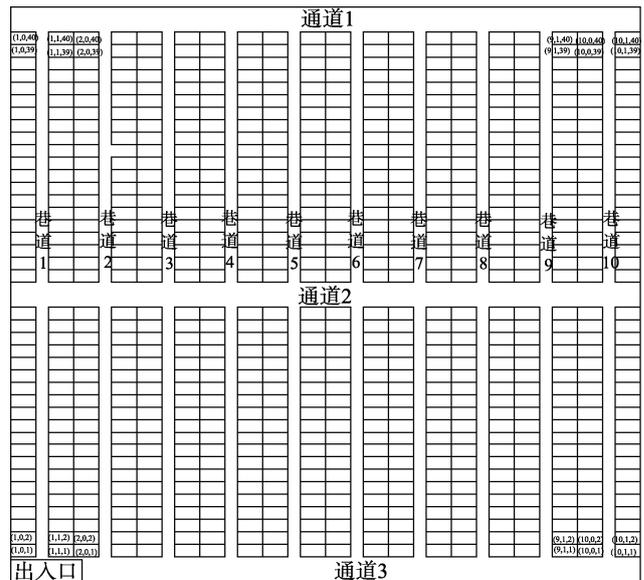


图 1 双区型仓库平面  
Fig.1 Two-block warehouse plan

- 3)  $b$  表示同一巷道中左右两侧的货架,左侧货架即  $b=0$ ,右侧货架即  $b=1$ ;
- 4)  $c$  表示同一排货架中货位的标号,  $c=1, 2, 3 \dots C$ 。

关于拣货单的符号说明如下所述。

- 1)  $R$  表示 1 个周期内的拣货单数量,编号  $r=1, 2, 3 \dots R$ ;
- 2)  $K$  表示 1 个周期内按照拣货单需求拣取的 SKU 的总量,编号  $k=1, 2, 3 \dots K$ ;
- 3)  $n$  表示 1 张拣货单中 SKU 的总量;
- 4)  $H$  表示 1 个周期内需要调整的 SKU 的总量,编号  $h=1, 2, 3 \dots H$ 。

关于拣货员的相关变量如下所述。

- 1)  $v$  表示拣货人员平均行走速度;
- 2)  $t_j$  表示拣货人员平均拣取单个 SKU 所需的时间。

假设有 10 条纵向通道,每条巷道有 40 个货位,每个巷道的宽度和货架的长宽均为 1 m,每个通道的宽度为 2 m。采用混合路径策略,先求出任意 2 个货位之间的最短距离  $d_{ij}$ 。

- 1) 2 个货位点在同一条拣货巷道中,在中间通道的同侧:

$$d_{ij} = |c_i - c_j|; (a_i = a_j, b_i = b_j, 0 \leq c_i, c_j \leq 20 \text{ 或 } 20 < c_i, c_j \leq 40) \tag{1}$$

$$d_{ij} = |c_i - c_j| + 1; (a_i = a_j, b_i \neq b_j, 0 \leq c_i, c_j \leq 20 \text{ 或 } 20 < c_i, c_j \leq 40) \tag{2}$$

- 2) 2 个货位点在同一条拣货巷道中,分散在中间通道两侧:

$$d_{ij} = |c_i - c_j| + 2; (a_i = a_j, b_i = b_j, 0 < c_i \leq 20 < c_j \leq 40 \text{ 或 } 0 < c_j \leq 20 < c_i \leq 40) \tag{3}$$

$$d_{ij} = |c_i - c_j| + 2 + 1; (a_i = a_j, b_i \neq b_j, 0 < c_i \leq 20 < c_j \leq 40 \text{ 或 } 0 < c_j \leq 20 < c_i \leq 40) \quad (4)$$

3) 2 个货位点在不同拣货巷道中, 在中间通道的同侧:

$$d_{ij} = \min(c_i + c_j, 40 - (c_i + c_j)) + |a_i - a_j| \times 3; (a_i \neq a_j, b_i = b_j, 0 < c_i, c_j \leq 20) \quad (5)$$

$$d_{ij} = \min(c_i + c_j, 40 - (c_i + c_j)) + |a_i - a_j| \times 3 + 1; (a_i \neq a_j, b_i \neq b_j, 0 < c_i, c_j \leq 20) \quad (6)$$

$$d_{ij} = \min(c_i + c_j - 40, 80 - (c_i + c_j)) + |a_i - a_j| \times 3; (a_i \neq a_j, b_i = b_j, 20 < c_i, c_j \leq 40) \quad (7)$$

$$d_{ij} = \min(c_i + c_j - 40, 80 - (c_i + c_j)) + |a_i - a_j| \times 3 + 1; (a_i \neq a_j, b_i \neq b_j, 20 < c_i, c_j \leq 40) \quad (8)$$

4) 2 个货位点在不同拣货巷道中, 分散在中间通道两侧:

$$d_{ij} = |c_i - c_j| + 2 + |a_i - a_j| \times 3; (a_i \neq a_j, b_i = b_j, 0 < c_i \leq 20 < c_j \leq 40 \text{ 或 } 0 < c_j \leq 20 < c_i \leq 40) \quad (9)$$

$$d_{ij} = |c_i - c_j| + 2 + |a_i - a_j| \times 3 + 1; (a_i \neq a_j, b_i \neq b_j, 0 < c_i \leq 20 < c_j \leq 40 \text{ 或 } 0 < c_j \leq 20 < c_i \leq 40) \quad (10)$$

## 2 动态拣货策略的设计

对于确定的拣货单, 包括若干订单, 涉及不同货位上的不同货物, 拣货作业是拣货员根据拣货单依次到相应货位上完成拣选的 1 个工作循环, 目标函数是完成拣货的总时间最短。采用静态拣货策略, 1 个周期内, 拣货员依次完成 1 个周期内的拣货单, 各个拣货单之间是独立的, 没有必然的联系<sup>[15]</sup>。动态拣货的基本思想是将拣货作业和动态货位调整这 2 个过程合二为一, 在不增加当前拣货单拣货路径的情况下, 最大可能地减少后续拣货单的拣货路径, 以同一周期内总的作业时间最小为总目标进行优化。值得注意的是, 在不改变现有货架结构和储位的情况下, 要尽量减少调整货位的复杂性, 使调整方案尽可能地简单可行, 以方便拣货员操作实施。

具体思路: 当大量拣货单需要处理时, 对同一周期内当前的拣货单进行拣货的同时, 调整之后拣货单中某货物的货位, 优化后续拣货单的拣选路径, 以此类推, 进而达到优化同一个周期内所有拣货单的拣货路径, 提高拣货效率的目的。由此可见, 所提出的动态拣货策略优化模型的最终目的是依次对同一周期内的拣货单中某些货物的货位进行适当调整, 使得同一周期内所有拣货单的总拣货路径更短, 总的拣货时间更少。

具体做法: 首先在仓库出入口旁边设定 1 个空货

位; 然后以拣货单为单元设计其最优路径的数学模型, 优化出拣货单 1 的拣货路径; 找出拣货单 2 中途经过拣货单 1 的所有货位, 循环判断调整哪个货位可以使拣货单 2 的拣货路径最大化地得到减少, 在进行拣货单 1 的拣选作业时, 将这个货位的货物顺便带至出入口旁边的空货架; 根据重新规划的拣货单 2 路径, 找出拣货单 3 中途经过拣货单 2 的所有货位, 以此类推, 直到完成该周期内所有的拣货单任务。该货位动态调整策略在不增加当前拣货单拣选路径的情况下, 最大可能地优化后续拣货单的路径, 且可以避免二次分拣。同时, 该策略不仅适用于人工拣货, 也适用于电子标签拣选。另外, 由于调整只涉及下一张拣货单的确定数量的货物, 不针对整个货位进行调整, 所以不影响仓库已有的 WMS。由此可见, 所设计的动态拣货策略简单方便, 易于实施, 进一步提高了拣货效率, 其具体调整流程见图 2。

## 3 动态拣货优化模型的建立

### 3.1 模型的假设

- 1) 采用摘果式、按拣货单拣取货物, 其中空闲等待时间忽略不计。
- 2) 拣货员可以沿纵向的巷道和横向的通道 2 个方向行走。
- 3) 拣货区分布: 每个货位的大小相同, 承载力相同, 每个货位只存储 1 种货物。
- 4) 拣货过程中不存在缺货现象, 即补货时间忽略不计。
- 5) 每笔拣货单的大小不超过拣货员的最大拣货量。
- 6) 货位足够大, 保证每个 SKU 都能存储, 即存储时不考虑 SKU 的体积。
- 7) 拣货员每张拣货单拣选只调整 1 个 SKU 的货位, 即每一条拣货路径最多只有 1 个 SKU 位置改变。

### 3.2 模型的构建

静态拣货策略下, 每张拣货单的作业时间包括拣选订单行走时间, 订单进行扫描、检查、确认等的初始化操作时间, 搜索位置, 拣取 SKU, 放置 SKU 等。模型将搜索位置、拣取 SKU、放置 SKU 和其他所花时间合并在一起, 即为每个 SKU 的拣取时间。由此可见, 总时间分 3 部分: 拣货人员按照订单进行拣取的行走时间、初始化操作时间、每个 SKU 的拣取时间。拣货单以总拣货时间最短为目标的函数见式 (11)。在动态拣货策略下, 当前拣货单  $r$  的路径优化模型除了包含式 (11) 中的时间外, 还包含需要调

整货位的拣货时间，见式(12)。重要的是，对于后续拣货单需要调整货位的选择，采用循环判断找出可以使后续拣货单拣货距离减少最多所对应的货位，式(13)为减去拣货单 $r+1$ 被调整货位后的最优路径模型。优化后，1个周期内所有拣货单总的拣货时间见式(14)。

$$Z_r = \frac{\text{Min} \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij}}{v} + t_r + \sum_{k=1}^K \sum_{a=1}^A \sum_{c=1}^C t_u S_{r,k} x_{a,c,k} \quad (11)$$

$$Z'_r = \frac{\text{Min} \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij}}{v} + t_r + \sum_{k=1}^K \sum_{a=1}^A \sum_{c=1}^C t_u S_{r,k} x_{a,c,k} + t_u \quad (12)$$

$$Z'_{r+1} = \frac{\text{Min} \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij}}{v} + t_{r+1} + \sum_{k=1}^K \sum_{a=1}^A \sum_{c=1}^C t_u (S_{r+1,k} - 1) x_{a,c,k} \quad (13)$$

$i, j \neq$  被调整货位

$$U = Z'_1 + \sum_{r=1}^{R-1} (Z'_{r+1} + t_u) \quad (14)$$

模型的约束条件如下所述。

1) 拣货单中的每个货位都要被访问1次， $i$ 和 $j$ 均为被拣货位。

$$x_{ij} = \begin{cases} 1; & \text{货位 } i, j \text{ 之间存在路径} \\ 0; & \text{货位 } i, j \text{ 之间不存在路径} \end{cases} \quad (15)$$

$$\sum_{i=0}^n x_{ij} = 1 (i \neq j) \quad (16)$$

$$\sum_{j=0}^n x_{ij} = 1 (i \neq j) \quad (17)$$

2) 每种SKU只能存放在1个货位：

$$\sum_{a=1}^A \sum_{c=1}^C x_{a,c,k} = 1; \forall k \quad (18)$$

3) 每个货位最多只能存放1种SKU：

$$\sum_{k=1}^K x_{a,c,k} \leq 1; \forall a, c \quad (19)$$

4) 拣货人员最少行走1条巷道：

$$\sum_{a=1}^A G_{r,a} \geq 1; \forall r \quad (20)$$

5) 待调整的SKU的数量最多不超过1个：

$$\sum_{s=1}^{S_{r,k}} e_{r,s} \leq 1; \forall r \quad (21)$$

6) 对拣货人员行走速度、订单初始化操作时间和拣取单件SKU的平均时间进行约束：

$$v > 0, t_r > 0, t_u > 0 \quad (22)$$

7) 货物的品类 $K$ ，巷道数 $A$ ，货位编号 $C$ ，拣货单 $r$ 需要拣取SKU的个数 $S_{r,k}$ 的约束：

$$K > 0, A > 0, C > 0, S_{r,k} > 0 \quad (23)$$

式中： $x_{ack}$ ——货物 $k$ 在第 $a$ 巷道第 $c$ 货位，其值为1，否则为0；

$S_{r,k}$ ——拣货单 $r$ 需要拣取SKU的个数；

$t_r$ ——拣货单 $r$ 进行扫描、检查、确认等初始化操作所需时间；

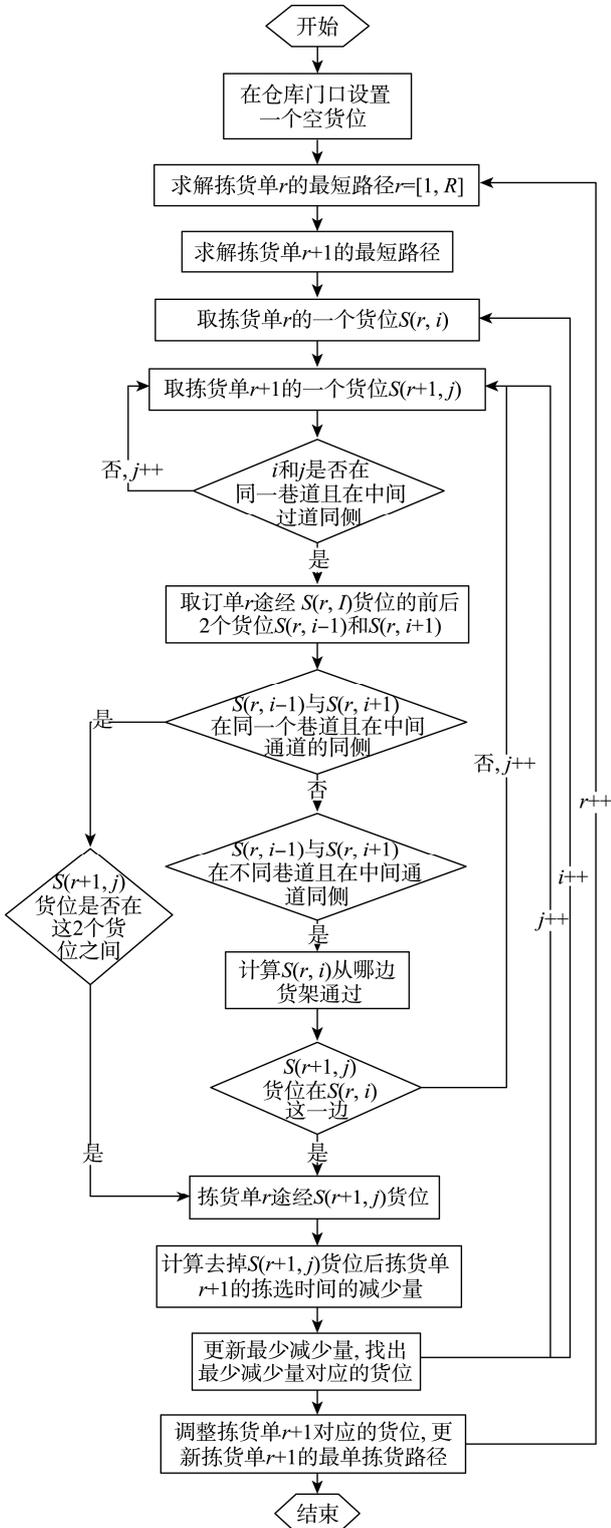


图2 动态货位调整流程

Fig.2 Dynamic cargo space adjustment flow chart

$t_u$ ——拣货人员平均拣取单个 SKU 的时间；

$G_{r,a}$ ——拣货单  $r$  拣取过程中是否访问巷道  $a$ ，若访问为 1，否则为 0；

$e_{r,s}$ ——订单  $r$  拣取过程中拣取被调整的 SKU，

$$e_{r,s} = \begin{cases} 1, & \text{被调整货位 } s \text{ 使拣货单 } r \text{ 的拣货时间减少得最多} \\ 0, & \text{其它货位} \end{cases}$$

### 4 求解算法与实例分析

#### 4.1 求解算法

文中问题属于 NP-hard 问题，通常采用启发式优化算法进行求解。遗传 (GA) 算法通过概率意义下基于“优胜劣汰”思想的群体遗传操作来实现优化。由于其采用适应度值大小来决定个体是否被复制的选择机制，因此容易出现来源于同一种群的个体被大量繁衍的情况，形成近亲繁殖，造成算法存在局部寻优能力差、收敛速度较慢、易陷入局部极值点等缺点，但在把握搜索整体过程方面具有优势。混合遗传模拟退火 (GASA) 算法是在 GA 算法的基础上，吸收模拟退火 (SA) 算法 (具有较强的局部搜索能力) 融合而形成的一种算法。文中分别采用 GA 算法和 GASA 算法进行实例求解与分析。

#### 4.2 实例分析

以某配送中心的拣货单为例，假定拣货人员的平均行走速度为 1.3 m/s，每个巷道的宽度为 1 m，每个横向通道的宽度为 2 m，货架的长和宽均为 1 m，对订单  $r$  进行扫描、检查、确认等初始化操作所需时间为 2 s，拣货人员拣取单个货物所需时间为 1.7 s。

##### 4.2.1 算法适应性验证

1) 基于 GA 算法求解。上述拣货单的 21 种货物，采用 GA 算法，最优路径拣货顺序为：0—5—6—7—8—12—15—16—21—20—19—17—13—14—18—11—10—9—4—3—1—2—0，总拣货距离为 330 m，总拣货时间为 291.55 s，拣货路径示意图见图 3，拣货路径见图 4。

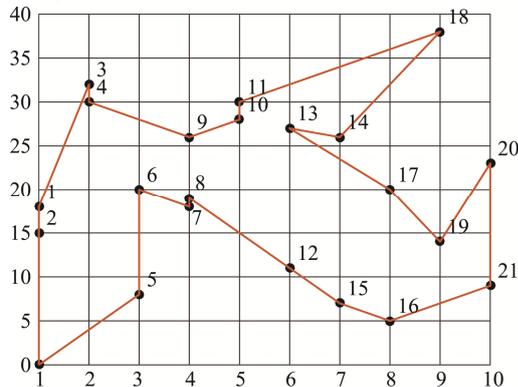


图 3 路径示意  
Fig.3 Path diagram

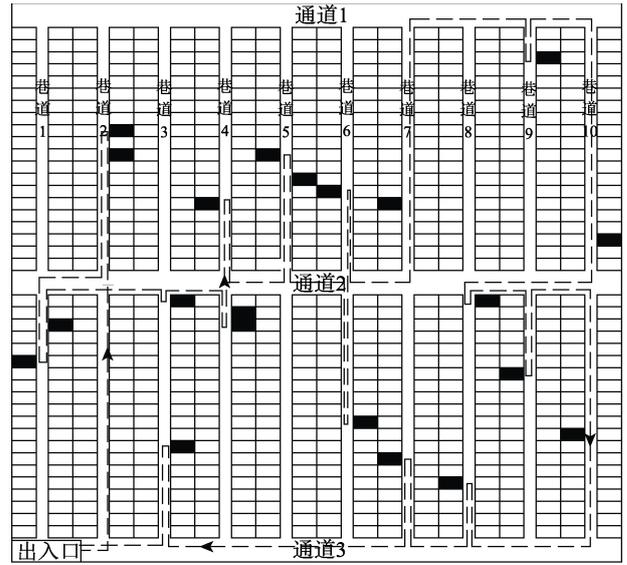


图 4 拣货路径  
Fig.4 Picking path

2) 基于 GASA 算法求解。某拣货单中有 21 种货物需要拣选，采用 GASA 算法，最优路径拣货顺序为：0—5—6—7—8—9—10—11—13—12—15—16—21—20—19—17—14—18—3—4—1—2—0，总拣货距离为 316 m，总拣货时间为 280.78 s。拣货路径见图 5，拣货路径见图 6。

3) 对比结果及分析。由表 1 和图 7 可以看出，GASA 算法比 GA 算法更适合该问题的求解，GASA 算法的拣货路径长度和拣货所需时间均小于 GA 算法得到的结果，用 GA 算法求解时大概在 250 代左右收敛，得到拣货时间为 291.55 s；用 GASA 算法求解时大概在 200 代左右收敛，到 520 代时跳出局部最优，得到拣货时间为 280.78 s，比 GA 算法得到的结果减少了 10.77 s，节约了约 4% 的时间。

##### 4.2.2 动态拣货路径模型的求解及分析

以一个拣货周期的 10 张拣货单所涉及的 139 个货位上的货物为例进行求解分析。

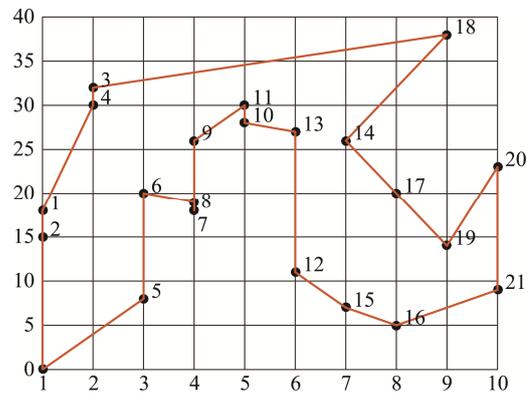


图 5 路径示意  
Fig.5 Path diagram

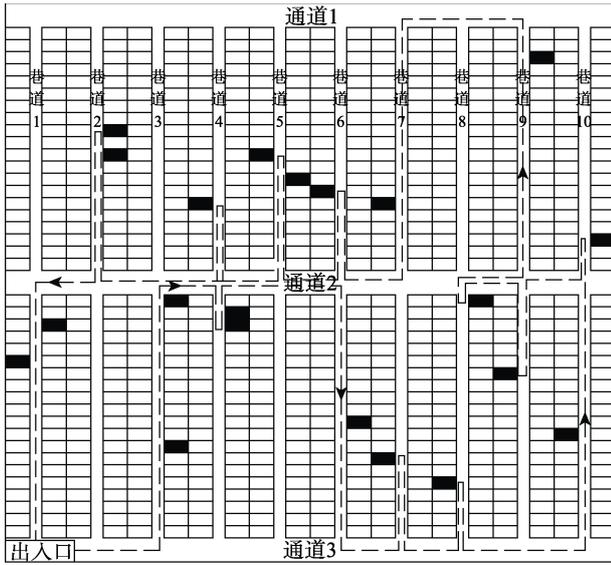


图6 拣货路径  
Fig.6 Picking path

表1 GA 和 GASA 算法对比  
Tab.1 GA and GASA algorithm comparison

算法	CPU 运行 时间/s	拣货路径 长度/m	拣货 时间/s
GA	可以忽略	330	291.55
GASA	可以忽略	316	280.78

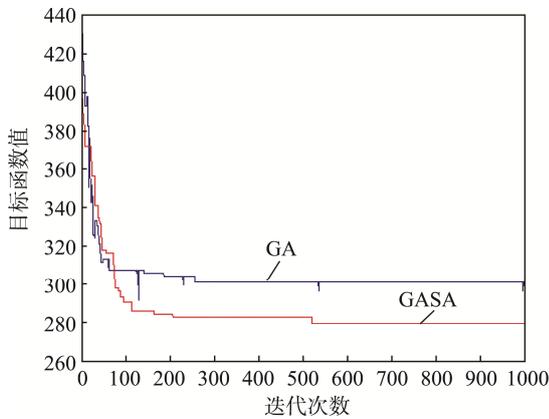


图7 算法对比  
Fig.7 Algorithm comparison

1) GA 算法与 GASA 算法求解对比。运用 GA 算法, 分别对每个拣货单的静态拣货时间、动态拣货时间进行仿真求解, 见表 2。由表 2 可以看出, 10 张拣货单在静态拣货策略下总拣货时间为 2273.3 s; 在动态拣货策略下总拣货时间为 2150.2 s, 节约的拣货时间为 123.1 s, 拣货距离减少了 158 m。运用 GASA 算法, 该周期的 10 张订单采用静态拣货策略下总拣货时间为 2181.1 s, 采用动态拣货策略下总拣货时间为 2051.8 s, 拣货时间节约了 129.3 s, 总拣货距离节约了 168 m, 见表 3。对比 2 种算法, 见表 4, 可以看出, 采用 GASA 算法的动态拣货策略效果更好,

不仅运算快, 而且比 GA 算法和静态策略下的拣货时间更短、距离更短。采用 GASA 算法, 对比静态拣货策略、动态策略下的 10 张拣货单可节约 6% 的时间。

表2 基于 GA 算法的动静态计算结果对比  
Tab.2 Comparison of dynamic and static calculation results based on GA algorithm

订单 编号	静态策略 的拣货时 间/s	动态策略 的拣货时 间/s	减少的 拣货时 间/s	减少的距 离/m	调整的货 号/位
1	245.1	240.9	4.2	8	207
2	176.4	164.5	11.9	14	312
3	228.9	213.8	15.1	17	403
4	185.3	173.5	11.8	15	512
5	241.6	233.1	8.5	11	605
6	255.3	240.1	15.2	21	701
7	232.6	214.9	17.7	23	812
8	264.1	261.2	2.9	3	906
9	201.7	187.6	14.1	19	1011
10	242.3	220.6	21.7	27	0
合计	2273.3	2150.2	123.1	158	

表3 基于 GASA 算法的计算结果对比  
Tab.3 Comparison of calculation results based on GASA algorithm

订单 编号	调整前拣 货时间/s	调整后拣 货时间/s	减少的拣 货时间/s	减少的 距离/m	调整的 货号/位
1	232.6	228.2	4.4	8	207
2	169.8	157.5	12.3	16	312
3	219.5	204.1	15.4	20	403
4	174.1	161.8	12.3	16	512
5	233.8	224.6	9.2	12	617
6	246.8	231.4	15.4	20	701
7	221.2	202.7	18.5	24	812
8	255.5	252.4	3.1	4	906
9	191.5	177.6	13.9	18	1011
10	236.4	211.6	24.8	30	0
合计	2181.1	2051.8	129.3	168	

表4 动态调整结果对比  
Tab.4 Comparison of dynamic adjustment results

算法	CPU 运行 时间/s	静态策 略的拣 货时间/s	动态策略 的拣货 时间/s	减少的 拣货 时间/s	减少的 距离/m
GA	237	2273.3	2150.2	123.1	158
GASA	193	2181.1	2051.8	129.3	168

2) 不同规模拣货单的算法对比。采用 GASA 算法, 分别计算 10, 20, 50, 100 张拣货单的动态拣货时间, 见表 5。由表 5 可以看出, 随着周期内拣货单数量的增加, 平均每张拣货单节约的时间和节约的拣货距离会逐渐增加, 平均每张拣货单节约的时间由

12.93 s 增加到 18.26 s, 平均每张拣货单节约的拣货距离由 16.8 m 增加到 23.69 m, 节约的拣货时间由 5.93% 上升到 8.37%。由此可见, 动态拣货策略随着拣货量的提高, 拣货效率的提高越明显, 即拣货量越大, 节约的拣货时间和拣货距离越多。

表 5 不同规模拣货单的结果对比

Tab.5 Comparison of results of picking orders of different sizes

拣货单/张	静态拣货时间/s	动态拣货时间/s	节约的总距离/m	平均每张拣货单节约距离/m	节约的总时间/s	平均每张拣货单节约时间/s	节约时间所占百分比/%
10	2181.1	2051.8	168	16.8	129.3	12.93	5.93
20	4365.7	4089.3	359	17.95	276.4	13.82	6.33
50	10 907.7	10 149.5	986	19.72	758.2	15.16	6.95
100	21 819.3	19 993.6	2369	23.69	1825.7	18.26	8.37

## 5 结语

文中设计了易于实施的动态货位调整策略, 达到了拣货总时间最短的目标, 构建了拣货路径优化问题的数学模型, 并分别采用 GA 算法和 GASA 算法对模型进行了求解。仿真表明, 相对于 GA 算法求解, GASA 算法更适合文中模型的求解, 算法性能更稳定, 求解得到的拣货时间和拣货距离也更优。相对于静态拣货策略, 通过动态调整拣货单中货物的位置, 可以进一步减少拣货时间。同时说明, 拣货单越多, 则节约的拣货时间和拣货距离就越多, 从而提高了整个物流配送中心的拣货效率和服务水平。文中的模型和数值仿真的结论可为电商企业规划拣货路径提供决策参考。

### 参考文献:

- [1] 冯佳. 一种配送中心订单拣选方式优化的算法[D]. 广州: 暨南大学, 2008.  
FENG Jia. An Algorithm for Optimizing Order Picking Method in Distribution Center[D]. Guangzhou: Jinan University, 2008.
- [2] CARON F, MARCHET Q, PEREGO A. Layout Design in Manual Picking System: a Simulation Approach[J]. Integrated Manufacturing Systems, 2000(11): 94—104.
- [3] ROODBERGEN K J, KOSTER R. Routing Methods for Warehouses with Multiple Cross Aisles[J]. International Journal of Production Research, 2001, 39(9): 1865—1883.
- [4] 王宏, 符卓, 左武. 基于遗传算法的双区型仓库拣货路径优化研究[J]. 计算机工程与应用, 2009, 45(6): 224—228.  
WANG Hong, FU Zhuo, ZUO Wu. Research on Picking Path Optimization of Dual-zone Warehouse Based on Genetic Algorithm[J]. Computer Engineering and Applications, 2009, 45(6): 224—228.
- [5] KULAK O, TANER M E. Joint Order Batching and Picker Routing in Single and Multiple-cross-aisle Warehouses Using Cluster-based Tabu Search Algorithms[J]. Flexible Services & Manufacturing Journal, 2012, 24(1): 52—80.
- [6] 李建斌, 周玮, 陈峰. B2C 电子商务仓库拣货路径优化策略应用研究[J]. 运筹与管理, 2014(1): 7—14.  
LI Jian-bin, ZHOU Wei, CHEN Feng. Research on Application of Picking Path Optimization Strategy in B2C E-Commerce Warehouse[J]. Operational Planning and Management, 2014(1): 7—14.
- [7] 李栋栋. 双区型仓库拣货路径优化研究[D]. 青岛: 青岛大学, 2015.  
LI Dong-dong. Research on Picking Path Optimization of Double-zone Warehouse[D]. Qingdao: Qingdao University, 2015.
- [8] 刘进平. 配送中心拣货路径与储位分配策略仿真分析[J]. 计算机仿真, 2015, 32(4): 310—315.  
LIU Jin-ping. Simulation Analysis of Picking Path and Storage Allocation Strategy in Distribution Center[J]. Computer Simulation, 2015, 32(4): 310—315.
- [9] 李梅娟, 陈雪波. 基于群集智能算法的路径规划问题[J]. 清华大学学报, 2007, 47(S2): 1770—1773.  
LI Mei-juan, CHEN Xue-bo. Path Planning Problem Based on Cluster Intelligent Algorithm[J]. Journal of Tsinghua University, 2007, 47(S2): 1770—1773.
- [10] 李江伟, 许伦辉. 退火算法与神经网络算法结合在路径规划中的研究[J]. 自动化与仪表, 2017, 32(11): 6—9.  
LI Jiang-wei, XU Lun-hui. Research on Annealing Algorithm Combined with Neural Network Algorithm in Path Planning[J]. Journal of Automation & Instrumentation, 2017, 32(11): 6—9.
- [11] 杜颖. 动态货位指派与拣货路径系统优化研究[D]. 南昌: 华东交通大学, 2017.  
DU Ying. Research on Dynamic Location Assignment and Picking Path System Optimization[D]. Nanchang: East China Jiaotong University, 2017.
- [12] 徐翔斌, 李秀. 固定路径下多阶段货位调整研究[J].

- 工业工程与管理, 2017, 22(5): 24—31.
- XU Xiang-bin, LI Xiu. Research on Multistage Cargo Position Adjustment under Fixed Path[J]. Industrial Engineering and Management, 2017, 22(5): 24—31.
- [13] 孙慧, 张柯, 张富金, 等. 基于遗传算法的双区型仓库人工拣货路径优化[J]. 青岛大学学报, 2014, 29(1): 78—82.
- SUN Hui, ZHANG Ke, ZHANG Fu-jin, et al. Optimization of Manual Picking Path for Two-zone Warehouse Based on Genetic Algorithm[J]. Journal of Qingdao University, 2014, 29(1): 78—82.
- [14] 王占磊. 配送中心订单分批及拣选路径优化问题研究[D]. 长春: 吉林大学, 2013.
- WANG Zhan-lei. Research on Order Batching and Picking Path Optimization in Distribution Center[D]. Changchun: Jilin University, 2013.
- [15] CHEN M C, HUANG C L, CHEN K Y, et al. Aggregation of Orders in Distribution Centers Using Data Mining[J]. Expert Systems with Applications, 2005, 28(3): 453—460.