

基于“货到人”拣选模式的储位分配问题研究

李珍萍¹, 范欣然¹, 吴凌云^{2,3}

(1. 北京物资学院 信息学院, 北京 101149; 2. 中国科学院 数学与系统科学研究院应用数学研究所管理、决策与信息系统重点实验室, 北京 100190; 3. 中国科学院大学 数学科学学院, 北京 100049)

摘 要:研究了“货到人”拣选模式下的储位分配问题,以订单拣选过程中搬运货架总时间最短为目标建立了整数非线性规划模型,并证明其为 NP-hard 问题,分别设计了求解模型的贪婪算法和单亲进化遗传算法。首先根据订单和物品的关联关系对物品进行聚类,基于聚类结果设计了求解模型的贪婪算法。然后设计了直接求解模型的单亲进化遗传算法,遗传算法中采用了 0-1 矩阵编码、多点基因倒位算子、单点基因突变算子和精英保留等策略,通过合理选取参数,能够很快求解出问题的近似最优解。最后利用模拟算例和一个具体实例进行计算,并对贪婪算法和遗传算法的求解时间和求解效果进行了比较分析。结果显示,对于小规模问题,两种算法均能在较短的时间内以很高的概率得到问题的全局最优解,对于中等规模的实际问题,利用两种算法得到的储位分配方案均优于企业目前采取的基于出库频率的储位分配方案,遗传算法得到的储位分配方案对应的货架搬运次数、货架搬运总时间等均优于贪婪算法。本文设计的遗传算法可以作为智能仓库管理信息系统的核心算法。

关键词:货到人;储位分配;整数非线性规划;聚类;贪婪算法;单亲进化遗传算法
中图分类号:O221.4;F253.4 **文章标识码:**A **文章编号:**1007-3221(2020)02-0001-11 **doi:**10.12005/orms.2020.0028

Study on the Storage Allocation Problem Under Cargo to Person Picking Mode

LI Zhen-ping¹, FAN Xin-ran¹, WU Ling-yun^{2,3}

(1. School of Information, Beijing Wuzi University, Beijing, 101149 China; 2. key Laboratorg of Management, Decision and Information Systems Academy of Mathematics and System Science, Chinese Academy of Sciences, Beijing 100190, China; 3. School of mathematical sciences, University Chinese Academy of Sciences, Beijing 100049, China)

Abstract:The storage allocation problem under cargo to person picking mode is studied. An integer nonlinear programming model is established to minimize the total time of moving shelves in the process of orders picking, the problem is proved to be NP-hard, and then two different approaches are designed to solve the model. In the greedy approach, the items are first clustered based on the relationship of orders and items, and then a greedy algorithm is applied on the clustering results. In another approach, the partheno evolution genetic algorithm is designed to solve the model directly. By adopting 0-1 matrix encoding rule, multipoint gene inversion operator, single point gene mutation operator and elitism scheme, with reasonable selection of parameters, the approximate optimal solution of the problem can be quickly found. Simulation is done on several simulation examples and a real case. The running time and effect of the greedy algorithm and the genetic algorithm are compared and analyzed. The results show that for small scale problems, both algorithms can get the global optimal solution with high probability in a relatively short time, and for the medium scale real case, the storage allocation schemes obtained by both algorithms are better than the enterprise's current storage allocation scheme. The storage allocation scheme obtained by genetic algorithm is better than that of the greedy algorithm in terms of the number of shelf movements and the total moving time. The genetic algorithm designed in this paper can be used as the core algorithm of intelligent warehouse management information system.

Key words: cargo to person; storage allocation; integer nonlinear programming; cluster; greedy algorithm; partheno evolution genetic algorithm

收稿日期:2017-09-09
基金项目:国家自然科学基金项目(71771028,71540028);北京市自然科学基金资助项目(Z180005);北京市属高校高水平科研创新团队建设项目(IDHT20180510);北京市科技创新服务能力建设—高精尖学科建设项目,北京市智能物流协同创新中心开放课题(BILSCIC-2019KF-18);北京物资学院重大项目(2019XJZD09);北京高等学校高水平人才交叉培养“实培计划”项目毕业设计(科研类)
作者简介:李珍萍(1966-),女,山东平度人,教授,博士,研究方向:优化理论与方法,智能算法;吴凌云(1975-),男,福建人,研究员,博士,研究方向:组合优化,智能算法,生物信息学。

0 引言

近年来,电子商务的快速发展改变了消费者的购物习惯和企业的商业运作模式,为了快速处理多品种、小批量、多批次的碎片化订单,“货到人”拣选模式受到越来越多国内外电商企业的关注,早在2012年,亚马逊以7.75亿美元收购了Kiva系统并建立了基于AGV的智能仓库,将订单拣选效率提高了一倍。在货到人拣选模式下,物品按照一定的规则摆放在货架上,拣选人员站在固定的拣选台前,当有订单需要拣选时,搬运机器人(AGV)会将指定的货架搬到拣选工作台供拣选人员从货架上拣取物品^[1-3]。由于在货到人拣选模式下,搬运机器人(AGV)需要不停地搬运货架到拣选口,因此,储位分配规则将直接影响订单拣选过程中的货架搬运次数和搬运时间(成本),从而影响订单拣选效率。研究货到人拣选模式下的储位分配问题,对于提高订单拣选效率,降低拣选成本具有重要意义^[2]。

虽然关于储位分配问题已经有大量的研究成果,但这些研究大部分都是针对传统人到货拣选模式的^[4-15]。由于货到人拣选模式与人到货拣选模式具有本质的区别,因此已有的关于人到货拣选模式下的储位分配方法并不能直接用于解决货到人拣选模式下的储位分配问题。近年来,部分学者针对货到人拣选模式下的储位分配问题开展了一些初步的研究工作。周方圆等以总拣选成本最小化为目标建立了基于AGV的货到人拣选模式下的储位分配问题数学模型,在一种物品可以同时存放在多个储位的前提下设计了求解模型的启发式算法^[16]。Li等研究了基于AGV的货到人拣选系统中物品储位已知的情况下对于给定的待拣选订单如何确定拟搬运货架的问题,以搬运货架总时间极小化为目标建立了混合整数规划模型,并设计了求解模型的启发式算法^[17]。Yuan等^[18,190]通过模拟实验方法研究了货到人拣选模式下基于工作量均衡的分区同步拣货储位分配问题,在一种物品只能存放到一个分区的假设下,设计了按照物品的出库速率确定物品储位所在分区的入库策略,但该研究并没有考虑物品之间的相关性。

除了基于AGV的货到人拣选模式以外,还有一些学者从不同角度研究了基于堆垛机的自动化立体仓库中的储位分配问题^[20-22]。虽然基于堆垛机的自动化立体仓库采用的也是“货到人”拣选模式,但自动化立体仓库中的堆垛机工作原理与

AGV搬运货架的工作原理不同,自动化立体仓库中每个托盘上只存放一种物品,堆垛机每次只能移动一个托盘,相当于每次只能拣选一种物品;基于AGV的货到人拣选模式中,每个货架上有多个储位,每个储位可以存放一种或一组绑定的物品组合(如亚马逊智能仓库就是把一组绑定的物品组合存放在一个储位上^[23]),因此一个货架上可以同时存放多种物品。AGV每次将一个货架搬运到拣选口,工作人员可以同时拣选一个货架上的多种物品。可见,基于AGV的货到人拣选模式下的储位分配问题比基于堆垛机的自动化立体仓库中的储位分配问题更加复杂。

本文拟对基于AGV的货到人拣选模式下的储位分配问题开展深入研究。以订单拣选过程中搬运货架的总行驶时间(成本)最小化为目标建立静态储位分配问题的优化模型,并设计求解模型的算法。

1 问题描述与分析

假设某智能仓库采用基于AGV的货到人拣选模式,该仓库拟将 m 种物品存储于 n 个可移动货架上。每个货架上有 h 个储位,每个储位最多可以存放一种物品,因此每个货架最多可以存放 h 种物品(其中 $m \leq nh$)。已知每个货架的位置及AGV搬运每个货架到拣选口的来回行走成本(时间)。根据历史订单信息可以统计出每种物品的历史出库频率,已知未来一段时间内需要拣选的 R 个订单中包含的物品种类等信息。假设每种物品至少存放在一个货架的某个储位上,拣选每个订单都需要把包含订单中某种物品的至少一个货架搬运至拣选口。问题是如何将 m 种物品摆放到 n 个货架上,才能使未来一段时间内拣选 R 个订单的过程中AGV搬运货架的总行驶成本(时间)最小。

由于货到人智能仓库主要存放的是多品种、小批量的小件物品^[1-3],顾客对这些物品的需求特点是:每个订单中包含一种或多种物品,但每一种物品的需求量较少。因此可以假设仓库中每种物品在一个货位上的存放数量能够满足一段时间内的订单需求,不需要考虑缺货的情况。为了提高顾客的满意度,大部分电商企业面对源源不断的在线订单都会采取实时拣选策略,实时拣选策略有两种方式:一种是按单个订单进行拣选,即每收到一个订单就组织一次拣选,拣选完一个订单再拣选下一个订单;另一种是分批拣选,即以很小的时间段

(比如10分钟)进行截单,对一个时间段内收到的所有订单同时进行拣选。如果把按时间段截单得到的所有订单看成一个合并的订单组合(或拣选单),则按时间分段的拣选也可以看成按照单个订单进行拣选。本文主要考虑按订单拣选方式。在按订单拣选方式下,如果不考虑AGV空载行驶成本(时间),则拣选每个订单的成本与AGV搬运货架的总行驶成本(时间)有关。当拣选一个订单需要搬运多个货架的时候,虽然多个AGV协同工作可以有效减少完成订单拣选的时间(由于多个AGV同时搬运货架),但并不能减少所有AGV搬运多个货架的总时间。由于AGV搬运货架的成本与搬运货架的总时间有关,因此,本文不考虑多个AGV协同工作的情况。

基于以上分析,为了简化问题,本文做出如下假设:

(1)每个货架的每个储位最多存放一种物品;且每个储位上存放的物品数量足以满足未来一段时间拣选 R 个订单的需求;

(2)每种物品至少存放在一个货架的某个储位上^[1-3];

(3)拣选一个订单中的某种物品时,只需要把包含该物品的一个货架搬运到拣选口即可^[17];

(4)订单拣选过程中,不考虑AGV空载行驶成本(时间)和排队等待拣选的成本(时间),只考虑AGV顶着货架来回行驶的成本(时间);

(5)每个货架在仓库中的位置固定,即AGV每次从仓库中搬运货架到拣选口,完成一个订单的拣选以后,仍然将货架放回仓库中原来的货架位置。

(6)不考虑订单合并拣选的情况,即拣选完一个订单再拣选下一个订单。

(7)不考虑多个AGV协同工作的情况。

2 储位分配问题的整数非线性规划模型

为了建立储位分配问题的数学模型,首先定义如下符号:

m :物品种类; n :货架个数; h :每个货架上的储位个数; t_j :AGV搬运货架 j 到拣选台的来回行驶成本(时间); R :待拣选的订单个数;

$$a_{ki} = \begin{cases} 1 & \text{第 } k \text{ 个订单中包含物品 } i \\ 0 & \text{否则} \end{cases}$$

定义决策变量:

$$x_{ij} = \begin{cases} 1 & \text{物品 } i \text{ 存放在货架 } j \text{ 上} \\ 0 & \text{否则} \end{cases}$$

$$y_{kj} = \begin{cases} 1 & \text{拣选第 } k \text{ 个订单需要搬运货架 } j \\ 0 & \text{否则} \end{cases}$$

以拣选订单过程中AGV搬运货架总行驶成本(时间)极小化为目标函数的储位分配问题可以表示成如下整数非线性规划模型:

$$\min z = \sum_{k=1}^R \sum_{j=1}^n t_j y_{kj} \quad (1)$$

$$\text{s. t.} \quad \begin{cases} \sum_{i=1}^m x_{ij} \leq h, j = 1, 2, \dots, n \end{cases} \quad (2)$$

$$\begin{cases} \sum_{j=1}^n x_{ij} \geq 1, i = 1, \dots, m \end{cases} \quad (3)$$

$$\begin{cases} \sum_{j=1}^n x_{ij} y_{kj} \geq a_{ki}, k = 1, \dots, R; i = 1, \dots, m \end{cases} \quad (4)$$

$$\begin{cases} x_{ij} = 0, 1; i = 1, \dots, m; j = 1, \dots, n \end{cases} \quad (5)$$

$$\begin{cases} y_{kj} = 0, 1; i = 1, \dots, R; j = 1, \dots, n \end{cases} \quad (6)$$

目标函数(1)表示极小化完成 R 个订单拣选需要AGV搬运货架的总行驶成本(时间);约束条件(2)表示每个货架最多可以存放 h 种物品,该约束等价于每个货位最多存放一种物品;约束条件(3)表示每种物品至少存放在一个货架上;约束条件(4)表示如果第 k 个订单中包含物品 i ,则拣选第 k 个订单中物品 i 时至少要将一个包含物品 i 的货架搬运至拣选口;约束条件(5)(6)为变量取值约束。

货到人拣选模式下的储位分配问题可以表示成整数非线性规划模型(1)~(6),可以证明当每个货架上的货位数 h 大于2的时候,该问题属于NP-hard问题。

定理1 当 $h > 2$ 时,模型(1)~(6)对应的货到人拣选模式下的储位分配问题是NP-hard问题。

证明 模型(1)~(6)对应的问题可以看成由储位分配和货架选择两个子问题构成的联合优化问题。其中储位分配子问题的决策变量为 x_{ij} ,约束条件包括(1)(2)(3)(5);货架选择子问题的决策变量是 y_{kj} ,约束条件包括(4)(6),目标函数是(1)。

在储位分配结果已知(即 x_{ij} 取值确定)的情况下,由式(1)(4)(6)构成的货架选择子问题等价于若干个独立的赋权集合覆盖问题^[23]的和,其中每一个待拣选订单相当于一个待覆盖的全集,每一个货架相当于一个子集,子集对应的权重就是搬运货架的时间。由于当子集中包含的元素个数大于2

(即一个货架上的储位个数大于2)的时候,赋权集合覆盖问题是 NP-hard 问题^[23]。因此当 $h > 2$ 时,货架选择子问题是 NP-难问题。

综上可以推出,由模型(1)~(6)定义的货到人拣选模式下的储位分配问题是 NP-hard 问题。

3 算法设计

由于货到人拣选模式下的储位分配问题是 NP-hard 问题,虽然该问题可以表示为整数非线性规划模型,理论上当问题规模较小时,可以直接求解整数非线性规划模型得到精确最优解。但实际中的储位分配问题涉及到的物品、货架、订单等数量巨大,其对应模型中的决策变量和约束条件都很多,难以在短时间内通过求解整数非线性规划模型得到精确最优解,因此需要设计求解模型的快速有效算法。

本文将根据物品的历史出库频率、待拣选订单中包含的物品信息等,分别设计两种求解模型的算法:基于聚类的贪婪算法和单亲进化遗传算法。

3.1 基于聚类的贪婪算法

算法的基本思想:首先根据历史订单(或者待拣选订单)中包含的物品种类等信息对物品进行聚类;再根据每个货架上的储位数量、每种物品的出库频率和各个货架对应的搬运时间等信息,利用贪婪思想对聚类以后的物品进行分组,并确定每一组物品对应的货架。

算法的步骤:

第1步 利用 K-means 方法对物品进行聚类(其中类别数目由用户自行设定)。

根据订单物品之间的关联关系和用户设定的类别数目,利用欧氏距离公式,采用 K-means 算法对物品进行聚类。由于聚类过程中考虑了订单与物品的关联性,因此聚类后的同一类中的物品之间的关联度较大,同一类物品被同时订购的可能性也较大。

第2步 对聚类后的物品进行二次分组。

根据聚类以后每类中包含的物品种类和一个货架上的储位数 h ,确定每类物品需要占用的货架数量并将每类物品分成一组或多组,使得每组物品刚好可以存放在一个货架上。

具体方法是:对于聚类后的第 i 类物品,假设其包含的物品种类为 m_i ,一个货架上的储位数为 h 。如果 $m_i \leq h$,则可以直接把第 i 类物品存放在一个货架上,否则数据 $m_i > h$,则第 i 类物品需要存放在

$\lceil \frac{n_i}{h} \rceil$ 个货架上。

当某一类中的物品需要存放在多个货架上时,可以按照该类中各种物品的历史出库频次由大到小排序,根据排序对物品进行二次分组,使得每组中的物品种类均不超过 h ,即每组物品都可以存放在一个货架上。

第3步 计算二次分组后每一组物品对应的货架搬运总次数,确定每一组物品对应的货架位置。

计算二次分组后每一组物品对应的出库频次(即每组物品对应的货架在订单拣选过程中需要搬运的总次数),按照每组物品对应货架的搬运总次数确定每一组物品应该存放的具体货架。尽可能将出库频次较高的分组放到搬运时间较短的货架上。

注 实际使用该算法时有可能出现某些分组包含物品种类较少的情况,为了提高储位利用率,可以在第2步中将物品种类较少的分组合并。

3.2 单亲进化遗传算法

在基于聚类的贪婪算法中,如果不同类别的物品不允许摆放在同一个货架上,则物品需要占用较多的货架。而且当同一类中物品种类较多时,直接按照物品的出库频次进行二次分组,并不能很好的体现物品之间的关联关系。为了提高储位利用率并降低拣选成本,本节将设计基于精英保留的单亲进化遗传算法直接求解储位分配模型。

3.2.1 单亲进化遗传算法操作规则

(1) 染色体编码

采用 0,1 矩阵编码方式表示 m 种物品在 n 个货架上的储位分配方案。具体编码方法如下:矩阵共有 m 行 n 列,每一行对应一种物品,每一列对应一个货架。由于每一种物品至少摆放在一个储位上,每一个货架最多有 h 个储位,因此矩阵中每一行至少有一个 1,每一列最多有 h 个 1。如:

$$S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

表示 5 种物品在 3 个货架上的储位分配结果,其中第 1 种和第 3 种物品摆放在第一个货架上,第 2 种和第 5 种物品摆放在第 2 个货架上,第 4 种物品同时摆放在第 1 和第 3 个货架上。本文只考虑物品和货架之间的对应关系,至于摆放在同一个货架上

的多种物品,可以随机分配储位。编码矩阵中的元素等于整数非线性规划模型中的决策变量 x_{ij} 。

(2) 种群初始化

根据染色体编码方式,随机产生若干个 m 行 n 列的 0,1 矩阵,矩阵的每行至少一个 1,每列中 1 的个数不超过 h 个。

(3) 适应度

对于每个个体解码后得到的储位分配方案,首先根据每个待拣选订单中包含的物品是否在某个货架上出现确定拣选订单过程中是否需要搬运该货架(如果待拣选订单中的一种或多种物品出现在某个货架上,则需要搬运该货架,即 $y_{kj} = 1$;否则该货架不需要搬运,即 $y_{kj} = 0$);然后计算出拣选每个订单需要搬运货架的总时间,进一步求出拣选所有订单需要搬运货架的总时间,即整数规划模型的目标函数值。

$$z = \sum_{k=1}^R \sum_{j=1}^n t_j y_{kj}$$

其中 t_j 表示 AGV 搬运货架 j 到拣选台的来回行走时间; y_{kj} 为决策变量,拣选第 k 个订单需要搬运货架 j 时为 1,不需要搬运时为 0。

由于整数规划模型的目标函数为极小化的,本文定义适用度函数为目标函数的倒数,即 $fitness = 1/z$ 。

(4) 交叉操作

采用单亲染色体多点基因倒位算子实现交叉操作。具体交叉操作规则是:从个体编码矩阵中随机选择连续的若干行,将选中的行倒序排列得到交叉后的染色体编码矩阵。

如交叉前的染色体编码矩阵为:

$$S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

如果选中第 2,3,4 行,则交叉后的染色体编码矩阵为:

$$S^1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

采用这种交叉操作,交叉后的个体仍然对应可行解。

(5) 变异操作

根据储位分配问题的约束条件和染色体编码

规则,本文采用单点突变变异算子进行变异操作。即从染色体编码矩阵中随机选择两行,交换两行的位置。如:

变异前的染色体编码矩阵为:

$$S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

如果选择了第 2,3 两行,则变异后的染色体为:

$$S^1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

(6) 精英保留策略

本文采用精英保留策略,将每一代中适用度最大的前 10% 的个体直接保留至下一代,下一代中的其余 90% 的个体则是从变异后的个体中选择的适用度最大的前 90%。

(7) 终止规则

本文设定的最大迭代次数作为终止条件,当迭代次数到达最大值时,终止运行并将最后一代中适用度最大的个体对应的解作为最优解。

3.2.2 单亲进化遗传算法的计算步骤

初始化:设置种群规模、最大迭代次数、交叉概率、变异概率等参数;

第 1 步 随机产生初始种群,计算每个个体对应的适用度值;令迭代次数 $t = 0$;

第 2 步 将种群中的个体按照适用度值由大到小排序;判断迭代次数是否达到预先设定的最大迭代次数,若是则结束计算,将适用度最大的个体对应的解作为近似最优解输出;否则转第 3 步;

第 3 步 根据精英保留策略,选择适用度最高的 10% 的个体,直接保留进入下一代;令当前代个体全部进入交配池;

第 4 步 按照交叉概率从交配池中选择部分个体进行交叉操作;

第 5 步 按照变异概率从交配池中选择部分个体进行变异操作;

第 6 步 计算变异后交配池中个体的适应度,从中选择适应度最高的前 90% 的个体进入下一代;与第 3 步保留下来的 10% 的个体合并在一起,构成下一代种群;令迭代次数 $t = t + 1$,返回第 2 步;

输出:最优的储位分配方案及对应的目标函数值。

3.3 算法复杂性分析

定理 2 基于聚类的贪婪算法的时间复杂度为 $O(Rmn^2)$, 单亲进化遗传算法时间复杂度为 $O(TPmn)$ 。

证明 在基于聚类的贪婪算法中, 第 1 步 K-means 聚类的计算量不超过 $O(tkmn)$, 其中 t 为 K-means 聚类的迭代次数, k 为聚类个数, 这里 k 最大可以取值为 R ; 第 2 步二次分组的计算次数不超过 $O(m)$, 第 3 步确定每个订单拣选过程中需要搬运的货架, 计算量不超过 $O(Rmn^2)^{[17]}$, 综上, 基于聚类的贪婪算法时间复杂度为 $\max\{O(tRmn), O(Rmn^2)\}$ 。当 K-means 聚类的迭代次数 t 不超过货架个数 n 的时候, 算法的时间复杂度为 $O(Rmn^2)$ 。

对于单亲进化遗传算法, 假设种群规模为 P , 最大迭代次数为 T , 则每一次迭代中, 第 1 步计算适用度函数的复杂度为 $O(Pmn)$, 第 2 步按照适用度函数值排序的复杂度为 $O(P\log P)$, 第 3 ~ 5 步的复杂度为 $O(Pmn)$, 第 6 步的复杂度为 $O(Pmn)$ 。假设且 $\log P \leq mn$, 则整个算法的时间复杂度为 $O(TPmn)$ 。

由以上分析可以看出, 当仓库中的货架个数 $n > TP$ 时, 基于聚类的贪婪算法复杂度大于单亲进化遗传算法, 当单亲进化遗传算法的迭代次数 T 或种群规模 P 较大时, 其计算复杂度将超过基于聚类的贪婪算法。

4 模拟计算与分析

4.1 小规模算例的分析

为了分析两种算法的求解效果, 本文利用不同参数(物品种类、货架数目、待拣选订单数目等)模拟产生了几组小规模算例。首先利用 Lingo 软件编程求解整数非线性规划模型得到各个问题的全局最优解。然后分别利用本文设计的基于聚类的贪婪算法和单亲进化遗传算法求解, 两种算法均使用 Matlab 编程在 i3-2367M CPU@ 1.40GHz 笔记本电脑上运行, 其中遗传算法设置种群规模为 50, 最大迭代次数为 1000 次。表 1 记录了两组算法 10 次运算求得的最优目标函数值和 10 次运算中得到全局最优解的次数, 表 2 记录了各个算法 10 次运算对应的平均运行时间。

表 1 两种算法求解小规模算例得到的目标函数值分析

物品种类	货架数目	待拣选 订单数目	各种算法得到的最优目标函数值 (10 次运算得到全局最优解的次数)		
			贪婪算法	遗传算法	全局最优解
20	5	50	620(10)	620(10)	620
20	5	100	1220(8)	1160(10)	1160
30	5	50	680(6)	650(10)	650
30	5	100	1320(5)	1210(9)	1210
50	10	100	1460	1280	超过 24 小时仍未得到最优解

表 2 各种算法求解小规模算例的平均运行时间

物品种类	货架数目	待拣选 订单数目	各种算法的平均运行时间(单位:秒)		
			贪婪算法	遗传算法	Lingo 软件求解整数规划
20	5	50	2.62	1.84	269
20	5	100	5.16	2.08	1061
30	5	50	3.96	1.90	644
30	5	100	7.81	2.26	302429
50	10	100	14.39	4.08	超过 24 小时

从表 1 可以看出, 对于小规模问题, 遗传算法和贪婪算法均能以很大的概率得到问题的全局最优解, 遗传算法得到全局最优解的概率明显高于贪婪算法。从表 2 可以看出, 随着问题规模的增大, 遗传算法和贪婪算法的计算时间增加不明显。利用 Lingo 软件直接求解整数规划模型的计算时间随着问题规模的增加呈指数增长, 对于包含 30 种物品、5 个货架、100 个订单的问题, 利用 Lingo 软件求解整数规划模型的运算时间已经超过 12 小

时, 但遗传算法和贪婪算法的计算时间均不足 10 秒钟。当问题规模增大到 50 种物品、10 个货架、100 个订单时, 利用 Lingo 软件求解整数规划模型运算 24 小时仍然未得到问题的最优解, 遗传算法迭代 1000 次的计算时间为 4.08 秒, 贪婪算法的运算时间为 14.39 秒。由此可见, 随着问题规模的增大, 本文设计的两种算法的计算速度优势非常明显。

4.2 不同规模算例的模拟计算与分析

为了进一步对比分析两种算法的计算效果, 我

们模拟产生了一批不同规模(物品种类、货架数目)的算例,假设各个算例对应的每个货架上货位数均为 10,每个货架对应的搬运时间服从[15,25]秒之间的均匀分布。对于每个算例分别用贪婪算法和遗传算法确定储位分配方案,其中遗传算法设置种群规模为 50,最大迭代次数为 1000 次。每个

算例重复运行 10 次,表 3 中列出了 10 次模拟计算的最短运行时间、平均运行时间。根据两种算法 10 次运算得到的最优储位分配方案,分别计算了拣选 100 个订单对应的搬运货架总次数、搬运货架总成本(时间),见表 4。

表 3 两种算法求解不同规模算例的运行时间表(单位:秒)

物品种类	货架数目	最短运行时间		平均运行时间	
		贪婪算法	遗传算法	贪婪算法	遗传算法
200	20	8.1	6.2	9.8	6.8
400	40	18.7	11.6	20.3	12.1
600	60	29.1	17.2	33.4	18.6
800	80	46.3	20.3	49.6	22.7
1000	100	50.4	25.4	55.2	27.9
2000	200	72.1	48.6	76.1	51.2

表 4 两种算法得到的最优储位分配结果对应于 100 个订单的拣选成本

物品种类	货架数目	搬运货架总次数		搬运货架的总时间(单位:秒)	
		贪婪算法	遗传算法	贪婪算法	遗传算法
200	20	328	279	6567	5586
400	40	335	281	6710	5624
600	60	339	284	6783	5689
800	80	341	296	6834	5918
1000	100	346	298	6926	5957
2000	200	352	302	7043	6045

从表 3 中可以看出,对于大规模问题,两种算法均可以在较短的时间内得到问题的解,随着问题规模的增大,两种算法的运行时间均有所增长,且遗传算法的运行时间均小于贪婪算法的运行时间。从表 4 中可以看出,对于各种规模的问题,遗传算法得到的储位分配结果对应于拣选 100 个订单的货架搬运次数和货架搬运总时间均小于贪婪算法。

4.3 实际案例分析

某小型电子商务公司采用基于 AGV 的货到人拣选模式,该公司仓库中共有 22 个货架,每个货架上有 10 个储位,每个储位最多可以摆放一种物品。根据货架在仓库中的位置以及 AGV 搬运货架时的行走速度等可以计算出 AGV 搬运每个货架到拣选口的来回行走时间,见表 5。

表 5 各个货架对应的来回搬运时间(单位:秒)

货架编号	搬运时间	货架编号	搬运时间	货架编号	搬运时间
1	20	9	21	17	22
2	20	10	21	18	22
3	20	11	21	19	22
4	20	12	21	20	22
5	20	13	21	21	22
6	20	14	21	22	22
7	20	15	21		
8	20	16	21		

该公司主要经销 140 种日用品和食品饮料等,目前公司采取的储位分配策略是:直接根据历史订单统计物品的出库频率,依次将出库频率较大的物品摆放在离拣选口较近的货架上,140 种物品全部

摆放在仓库中的 1~14 号货架上。

接下来我们将利用该公司 2017/2/17 和 2017/2/18 两天共 100 个订单的数据,分别使用贪婪算法和遗传算法对该公司仓库中的储位进行优

化,并分析优化前后的订单拣选过程中需要搬运货架的次数和搬运时间等。

(1)利用贪婪算法优化储位

由于 100 个订单中包含了该公司经销的全部 140 种物品,首先根据每个订单中包含的物品详细信息,利用 K-means 聚类方法将 140 种物品聚为 13 类;再根据每类中包含的不同物品种类数,按照每个货架上最多可以摆放 10 种物品计算出每类物品需要占用的货架个数。每类包含的物品种类、需要占用货架个数等信息如表 6 所示。

表 6 140 种物品的聚类结果统计表

类序号	包含物品种类	需要占用货架个数
1	46	5
2	1	1
3	2	1
4	5	1
5	4	1
6	21	3
7	2	1
8	2	1
9	16	2
10	7	1
11	3	1
12	29	3
13	2	1
合计	140	22

表 7 各组物品对应的总出库频次及对应货架的总搬运次数

分类序号	二次分组序号	包含物品种类	物品总出库频次	对应货架总搬运次数	分类序号	二次分组序号	包含物品种类	物品总出库频次	对应货架总搬运次数
1	1	10	45	31	6	12	1	1	1
	2	10	26	21	7	13	2	15	13
	3	10	16	15	8	14	2	15	13
	4	10	10	7	9	15	10	32	23
	5	6	6	5		16	6	6	5
2	6	1	11	11	10	17	7	27	16
3	7	2	17	13	11	18	3	16	12
4	8	5	21	16	12	19	10	38	26
5	9	4	73	28		20	10	17	15
6	10	10	38	24		21	9	8	7
	11	10	15	13	13	22	2	14	11

根据每组物品对应货架的总搬运次数,由大到小对分组进行排序,再根据仓库中每个货架对应的搬运时间由大到对货架进行排序,依次确定每个分组对应的货架位置。如第一组物品对应的货架总搬运次数最高,因此第一组物品应该摆放在搬运时间最短的 1 号货架上,依次类推,第 12 组物品对应的货架搬运次数最少,因此第 12 组物品摆放在搬运时间最长的 22 号货架上。按照这种储位分配结果,拣选 100 个订单需要搬运货架的总次数为

由表 6 可知,当某个类别中包含的物品种类不超过 10 种时,可以直接摆放在 1 个货架上,当某个类别中包含的物品种类超过 10 种时,需要占用多个货架,如第 1 类包含 46 种物品,需要占用 5 个货架($\lceil \frac{46}{10} \rceil = 5$),第 6 类和第 12 类均需要占用 3 个货架,第 9 类物品需要占用 2 个货架等。

对于需要占用多个货架的类别(如第 1 类)进行二次分组。将同一类中各种物品在 100 个订单中出现的次数(即物品的出库频次)从大到小排序,把出现频次最高的 10 种物品分为一组,出库频次排序在 11~20 的物品分为第二组,依次类推,直到所有物品分组完毕为止。二次分组后每一组物品均可以摆放在一个货架上,根据以上方法可以将表 2 中的 13 类物品分为 22 组,见表 7。

由于每一组物品可以摆放在一个货架上,当进行订单拣选时,每次搬运一组物品对应的货架时都可以同时拣选到货架上的所有物品,因此该组物品对应的货架搬运次数并不等于物品的出库频次之和。为此,需要根据每个订单中包含的物品种类计算拣选每个订单时各组物品对应的货架搬运次数,进一步计算出拣选 100 个订单的过程中,每组物品对应的货架总搬运次数,结果见表 7。

326 次,总搬运时间为 6697 秒。

由于以上算法对应的有些分组中的物品数量较少,为了提高仓库中的货位利用率,减少远距离货架的使用。可以考虑把包含物品种类较少的分组合并在一起。如本例中可以把第 5、7、22 组合并;6、8、9 组合并;13、14、16 组合并;17、18 组合并;12、21 组合并。合并后的变成 14 组,每组均包含 10 种物品。分别计算合并后每组物品对应货架的出库频次,按照出库频次从大到小为每组物品安

排货架得到货位分配方案,此时 140 种物品可以全部存放在搬运时间较短的 1~14 号货架上。该方案对应的拣选 100 个订单的货架搬运次数仍然为 326 次,总搬运时间为 6599 秒。由于分组合并以后减少了远距离货架的使用,因此总的货架搬运时间有所降低。

(2)利用单亲进化遗传算法优化储位

利用单亲进化遗传算法直接对 140 种物品进行储位分配时,设置参数为:种群规模为 50,最大迭代次数为 1000,交叉概率 0.5,变异概率为 0.1。利用 Matlab 编写遗传算法实现程序,在 i3-2367M CPU@ 1.40GHz 处理器、32 位操作系统配置的电

脑上运行 5.59 秒,得到近似最优的储位分配方案。140 种物品全部存放在 1~14 号货架上。按照该储位分配方案拣选 100 个订单需要搬运货架的次数为 273 次,总搬运时间为 5656 秒。

(3)结果分析

表 8 分别记录了两种算法得到的储位分配方案对应的拣选 100 个订单过程中需要搬运货架的总次数、搬运货架的总时间、以及算法运行总时间,表 8 中最后一行是按照该公司目前的基于出库频率的储位分配方案拣选 100 个订单的货架搬运总次数、货架搬运总时间等。

表 8 计算结果分析

算法类型	占用货架数量	拣选 100 个订单需要搬运货架总次数	拣选 100 个订单搬运货架的总时间(秒)	算法运行时间(秒)
遗传算法	14	273	5656	5.59
贪婪算法(分组合并后)	14	326	6599	9.074
贪婪算法(分组合并前)	22	326	6697	8.062
基于出库频率的储位分配方案	14	337	6801	-

从表 8 中可以看出,无论是使用遗传算法还是贪婪算法得到的储位分配方案都比该公司目前采用的基于出库频率的储位分配方案好。与该公司目前采用的基于出库频率的储位分配方案相比,两种算法得到的储位分配结果对应拣选 100 个订单时需要搬运货架的总次数分别减少了 19% 和 3.26%。拣选 100 个订单的货架搬运总时间分别降低了 16.8%、3% (采取分组合并策略) 和 2% (未采取分组合并策略)。这说明了考虑物品之间的关联性在基于 AGV 的货到人拣选模式下的储位分配问题中确实能大幅提高拣选的效率。使用贪婪算法时,如果采取二次分组合并策略,可以减少货架的使用数量,提高货架利用率,同时也能降低货架总搬运时间,可见在贪婪算法中加入二次分组合并策略可以有效改善解的质量。从计算时间来看,加入分组合并策略以后贪婪算法的计算时间会有所增加,二次分组合并问题本身也是一个组合优化问题,随着问题规模的增大,计算时间可能增加比较明显。综合考虑各项指标可以发现,遗传算法无论是计算速度还是计算效果都明显优于贪婪算法。

对于本节的实际问题,我们也尝试利用优化软件 Lingo 和 CPLEX 分别编程直接求解整数规划模

型,两个程序连续运行 700 多个小时都没有得到最终计算结果,因此该问题不适合使用优化软件直接求解整数规划模型。

4.4 算法优劣分析

本文设计的基于聚类的贪婪算法结合了聚类和贪婪算法的优点,通过聚类将经常出现于同一订单并且类型相近的物品划分为一类,体现了订单与物品、物品与物品之间的关联关系,另一方面将二次分组以后的物品,按照对应的货架搬运次数由大到小依次安排储位,有利于提高订单拣选效率,但是使用该方法时如果不采取二次分组合并策略,就会出现某些货架上物品种类很少的情况,导致占用较多的货架,在一定程度上会造成储位资源的浪费。如果在贪婪算法中加入二次分组合并策略,则可以将包含物品种类较少的分组合并在一起,从而有效减少货架的使用数量,提高储位利用率,同时也能减少订单拣选过程中的货架搬运总时间。因此加入分组合并策略的贪婪算法优于不加入分组合并策略的贪婪算法。

本文设计的基于精英保留的单亲进化遗传算法直接求解储位分配模型,不仅可以使关联度较大的物品尽可能摆放在相同的货架上,以减少订单拣

选过程中搬运货架的总次数,降低搬运货架的总时间,而且还能有效避免出现空缺的货位,减少货架的使用数量,提高仓库中的储位利用率。由于该算法计算速度快,效果好,因此可以作为智能仓库管理信息系统的核心算法。

本文研究的问题中需要根据历史订单的信息进行储位分配,再根据待拣选订单信息对储位分配结果进行评价,实际问题中历史订单和待拣选订单的数量往往都很多,即便是小规模的问题,其整数非线性规划模型的决策变量和约束条件也很多。由于储位分配问题属于 NP 难问题,对于中等以上规模的问题不适合直接利用整数非线性规划模型求解。利用本文设计的两种算法均可以在短时间内得到大规模问题的近似最优解。

5 结论

本文研究了基于 AGV 的货到人拣选模式下的静态储位分配问题,建立了以拣选订单过程中搬运货架总时间最短为目标的整数非线性规划模型,设计了求解模型的贪婪算法和单亲进化遗传算法。分别利用模拟数据和电商企业的真实数据进行了计算,并分析了各个算法的计算效果。

基于 AGV 的智能仓库自出现以来受到国内外大型电商企业的高度关注,亚马逊等大型企业已经将基于 AGV 的货到人智能仓库系统成功应用于仓储管理中,并大大提高了订单拣选效率。现阶段某些智能仓库系统中采用的是绑定货品组合的随机储位分配策略,即根据历史订单信息,首先将订单与物品绑定形成固定的货品组合,然后以固定的货品组合为存储单元在有空闲储位的货架中随机为其分配储位^[24]。由于历史订单种类繁多(且很多历史订单出库频率较低),因此订单与物品绑定以后形成的货品组合种类繁多,由于每一类货品组合都至少需要占用一个储位,所以以历史订单绑定形成的货品组合为存储单元的存储方式必然需要占用大量的货架(储位)。本文给出的储位分配策略是以物品为存储单元的,由于物品的种类远远小于物品组合的种类,因此按照本文的储位分配策略可以大大减少货架的使用数量,另外,本文在分配储位的时候考虑了物品之间的关联关系,将关联度较大的物品尽可能存放在一个货架上,因此可以有效

减少订单拣选过程中搬运货架的次数,降低拣选成本、提高拣选效率。

本文研究中假设一个储位上的物品数量足够满足一段时间内的订单拣选需求。实际中,由于不同物品出库频率相差较大,一个储位上的存储空间有限,对于出库频率较高的物品,一个储位上的存储量可能无法满足一段时间内的订单拣选需求,此时需要搬运多个包含该物品的货架到拣选口,这种情况下,可以结合文献[17]中的货架选择方法,确定拣选订单时需要搬运的多个货架的最佳组合。

本文研究的问题是在仓库中所有的储位上均为空闲储位情况下的静态储位分配问题,接下来我们将研究仓库中同时存在空闲储位、需要补货的储位和不需要补货的储位等多种情况下,如何进行动态储位分配的问题。我们还将研究较长时间内订单拣选过程中的在线储位分配问题,为解决智能物流仓储系统设计问题提供理论依据和算法支持。

参考文献:

- [1] Peter R. Wurman, raffaello D'andrea, mick mountz. coordinating hundreds of cooperative [J]. Autonomous Vehicles in Warehouses. AI Magazine, 2008, 29(1): 9-20.
- [2] Christopher J Hazard, Peter R. Wurman, raffaello D'andrea. alphabet soup: a testbed for studying resource allocation in multi-vehicle systems. In Proceedings of the {AAAI} Workshop on Auction-Based Robot Coordination, 2006, Key: citeulike:1585924.
- [3] John Enright J, Peter R. Wurman, optimization and coordinated autonomy in mobile fulfillment systems, automated action planning forAutonomous Mobile Robots: Papers from the 2011 AAAI Workshop(WS-11-09).
- [4] Jason Chao-Hsien Pan, Po-Hsun Shih, Ming-Hung Wu. Storage assignment problem with travel distance and blocking considerations for a picker-to-part order picking system [J]. Computers and Industrial Engineering, 2012, 62: 527-535.
- [5] Muppani V R, Adil G K. A branch and bound algorithm for class based storage location assignment[J]. European Journal of Operational Research, 2008, 189: 492-507.
- [6] Muppani V R, Adil G K. Efficient formation of storage classes for warehouse storage location assignment: simulated annealing approach [J]. Omega, 2008, 36: 609-618.
- [7] Renaud J, Ruiz A. Improving product location and order picking activities in a distribution centre[J]. Journal of the Operational Research Society, 2008, 59: 1603-1613.
- [8] Chen F T S, Chan H K. Improving the productivity of warehouse through the order picking of a manual-pick and multi-level rack distribution warehouse through the

- implementation of class-based storage[J]. *Expert System with Applications*, 2011, 38: 2686-2700.
- [9] David Ming-Huang Chiang, Chia-Ping Lin, Mu-Chen Chen. Data mining based storage assignment heuristics for travel distance reduction[J]. *Expert Systems*, 2014, 31(1): 81-90.
- [10] Eleonora Bottani, Margherita Cecconi, Giuseppe Vignali & Roberto Montanari. Optimisation of storage allocation in order picking operations through a genetic algorithm [J]. *International Journal of Logistics research and Applications: a leading Journal of Supply Chain Management*, 2012, 15(2): 127-146. DOI: 10.1080/13675567.2012.694860.
- [11] Seval Ene & Nursel Öztürk. Storage location assignment and order picking optimization in the automotive industry [J]. *Int J Adv Manuf Technol*, 2012, 60: 787-797.
- [12] Christoph H. Glock Eric H. Grosse Storage policies and order picking strategies in Ushaped order-picking systems with a movable base[J]. *International Journal of Production Research*, 2012, 50(16): 4344-4357, DOI: 10.1080/00207543.2011.588621.
- [13] Li J, Moghaddam M, Nof S Y. Dynamic storage assignment with product affinity and ABC classification-a case study[J]. *International Journal of Advanced Manufacturing Technology*, 2016(9): 1-16.
- [14] Wutthisirisart P, Noble J S, Chang C A. A two-phased heuristic for relation-based item location[J]. *Computers & Industrial Engineering*, 2015, 82: 94-102.
- [15] 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: 1-12.
- [16] 周方圆,李珍萍.基于“货到人”拣选模式的储位分配模型与算法[J]. *物流技术*, 2015, 34(9): 242-246.
- [17] Li Z P, Zhang J L, Zhang H J, Hua G W. Optimal selection of movable shelves under “cargo-to-person” picking mode [J]. *International Journal of Simulation Modelling*, 2017, 16(1): 145-156. Doi: 10.2507/IJSIMM16(1)C02.
- [18] Yuan R, Cezik T, Graves S. Stowage decisions in multi-zone storage systems, *international journal of production research*. doi: 10.1080/00207543.2017.1398428.
- [19] Yuan R. (2016) Velocity-based storage and stowage decisions in a semi-automated fulfillment system. Ph. D. Dissertation, Massachusetts Institute of Technology.
- [20] 李珍萍,于洋涛.并列选择单亲遗传算法在自动化立体仓库货位优化中的应用[J]. *数学的实践与认识*, 2016, 46(20): 179-186.
- [21] 陈璐,陆志强.自动化立体仓库中的储位分配及存取路径优化[J]. *管理工程学报*. 2012, 26(1): 42-47.
- [22] 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): 229-236.
- [23] Chvatal V. A greedy heuristic for the set-covering problem[J]. *Mathematica of Operations research*, 1979, 4(3): 233-235.
- [24] http://www.360doc.com/content/16/0121/23/22094781_529674705.shtml.