

# 基于改进遗传 $K$ -均值算法的多品种 小批量订单分批方法

邵泽熠,董宝力

(浙江理工大学机械与自动控制学院,杭州 310018)

**摘 要:** 针对货品种类多且库位分散的多品种小批量订单问题,构建了以订单相似度最高为目标的分批拣选优化模型,并采用改进遗传  $K$ -均值算法对模型进行求解。针对传统  $K$ -均值算法中  $K$  值人为确定造成聚类结果误差大的缺点,采用密度和最小距离综合最优指标确定多个初始聚类中心,并运用改进遗传  $K$ -均值算法确定最优分批数量,进行订单分批优化;得到分批结果后,用穿越式路径方法计算拣选距离,并与简单分批得到的距离进行对比。以某机械设备生产公司售后服务备件仓的订单分批拣选问题为对象进行仿真实验,结果表明该方法有效。

**关键词:** 订单分批拣选;初始聚类优化; $K$ -均值算法;遗传算法;穿越型路径

**中图分类号:** F253.4

**文献标志码:** A

**文章编号:** 1673-3851(2018)11-0732-07

## 0 引 言

拣选作业是指仓库根据客户订单要求或配送计划,将商品从储位拣取出来并进行分类集中的作业过程<sup>[1]</sup>。简单拣选作业效率低,特别是对于多品种小批量订单,采用单个订单拣选耗时耗力,所有订单统一拣选又不切实际,所以分批拣选就显得尤为重要。分批拣选可以提升拣选效率<sup>[2]</sup>,订单分批策略有时窗分批和订单相似度分批两种<sup>[3]</sup>。时窗分批适用于短时间内订单密集达到或有明显的时间延迟惩罚机制的情况;订单相似度分批适用范围广,常用方法有种子算法、优先规则算法、智能算法<sup>[4]</sup>等。针对订单拣选的实际情况,选择合理的分批方法对提升拣选效率具有重要意义。

王旭坪等<sup>[5]</sup>针对在线订单达到时间不确定的特性,把基于完成期限的动态时窗作为分批条件来进行订单的分批优化。文坚等<sup>[6]</sup>将制造延迟思想引入配送中心的拣选作业中,提出了基于时间延迟的动态时窗分批策略。李晓杰<sup>[7]</sup>将移动货架仓库系统作为背景,验证货位分配和订单组合策略对分批拣选

产生的影响。马廷伟等<sup>[8]</sup>对二进制粒子群算法进行了改进,并用改进算法求解订单分批问题。李诗珍等<sup>[9]</sup>采用聚类算法,以储位特征向量和坐标特征向量为条件进行订单分批,并用启发式算法计算拣选距离长短。胡小建等<sup>[10]</sup>考虑到主观选择聚类中心容易导致计算结果偏差,采用了基于 Canopy 算法的改进  $K$ -均值算法进行分批。

在上述研究中,有些从时间窗角度进行分批研究,但这些方法多应用于电子商务环境,对一般仓库分批拣选的适用性较低;还有些从订单相似度入手进行研究,只考虑了算法的部分优化,难以避免初始聚类个数和对象选择的盲目性。本文以某企业售服仓的备件订单分批拣选问题为研究对象,构建了以订单相似度最高为目标的分批优化模型。该模型先用密度和最小距离作为参数确定初始聚类中心,再用改进遗传  $K$ -均值算法获取  $k$  值,并进行聚类中心迭代以求出最优分批结果。本文通过计算拣选距离来衡量订单分批的有效性,为多品种小批量订单的分批方法提供了一种思路。

收稿日期:2018-06-29 网络出版日期:2018-09-03

基金项目:国家自然科学基金项目(51475434)

作者简介:邵泽熠(1994-),男,浙江绍兴人,硕士研究生,主要从事精益生产方面的研究。

通信作者:董宝力,E-mail:tydbl@zstu.edu.cn

## 1 问题描述及模型建立

### 1.1 问题描述

本文研究某售后服务备件仓某一段时间内到达的多个订单的分批问题。该问题中订单的需求货物种类多但数量少,需根据特定规则将订单分成几个批次进行拣选作业,目的是在满足拣选约束的情况下,降低拣选作业的时间成本,即拣选距离,订单分批拣选流程如图 1 所示。订单分批拣选描述为:在某一个时间段内共有  $s$  个订单  $O_1, O_2, \dots, O_s$  需要

进行拣选,每一个订单中有若干种货物,每一种货物数量不定。第  $t$  张订单  $O_t$  中共有  $p_t$  种货物,有  $q_t$  个需要被拣选的货物,这  $q_t$  个货物的总重量为  $w_t$ ,总体积为  $v_t$ 。假定仓库中每一种货物有且只有一个货位,以仓库入口集货点为坐标原点,横向为  $X$  轴,纵向为  $Y$  轴建立坐标系,把每个库位的中心位置视为该库位的坐标,那么订单  $O_t$  中货物  $a$  的坐标为  $(x_a, y_a)$ 。之后按照特定的分批规则,将  $s$  个订单进行分批处理,目标是使订单相似程度最高,总的拣选距离最短。

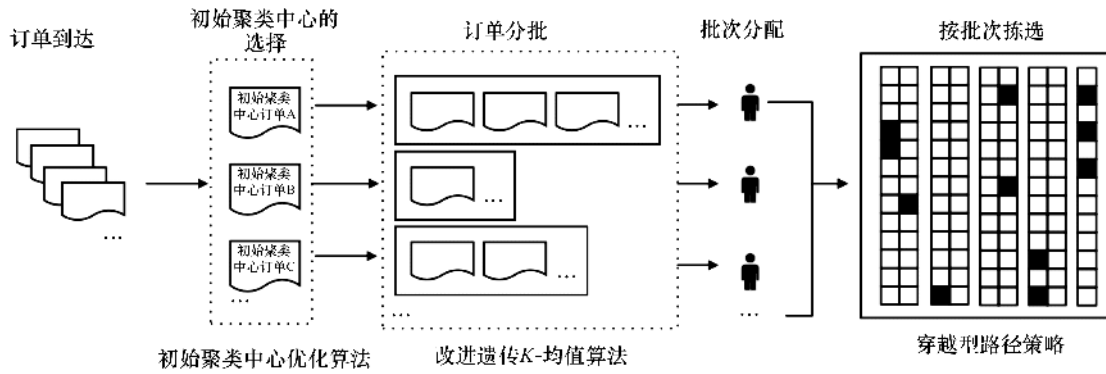


图 1 订单分批拣选流程

考虑到实际问题的复杂性和不确定性,在对订单分批问题建立模型之前,需要对该问题进行必要的简化并设定条件,本文中对于该问题的假设如下:

- 拣货小车的最大载重为  $W$ ,最大容积为  $V$ ;
- 每张订单必须包含不少于一种的货物,且一个订单中货物的总重量  $w_t \leq W$ ,总体积  $v_t \leq V$ ;
- 不考虑货物放置时产生的空隙部分而造成的容积浪费,即假设货物之间紧密放置;
- 同一张订单中的货物不可分割,必须一次拣选完毕;
- 不考虑缺货和插单等意外情况;
- 拣选员在拣选前明确拣选流程和路线,不考虑因失误导致的往返情况;

g) 由于是多品种小批量的订单分批拣选,采用穿越型路径策略较为合适。

### 1.2 模型建立

分批拣选的目标是把客户订单分成几批,每一批订单之间的相似度较高,在考虑拣选限制的前提下使总的拣选距离最短。建立订单分批拣选优化的数学模型如下:

目标函数:

$$\max Z = \sum_{k=1}^K b_k \times \delta_k \quad (1)$$

$$\delta_k = \begin{cases} 1, & \text{第 } k \text{ 批被选中} \\ 0, & \text{第 } k \text{ 批没有被选中} \end{cases} \quad (2)$$

约束函数:

$$\sum_{t=1}^s v_t \times \theta_{tk} < V \quad (3)$$

$$\theta_{tk} = \begin{cases} 1, & \text{订单 } t \text{ 被分配到第 } k \text{ 批次中} \\ 0, & \text{订单 } t \text{ 没被分配到第 } k \text{ 批次中} \end{cases} \quad (4)$$

$$\sum_{t=1}^s w_t \times \theta_{tk} < W \quad (5)$$

$$\bigcup_{k=1}^K \text{Batch}_k = O \quad (6)$$

$$\bigcap_{k=1}^K \text{Batch}_k = \phi \quad (7)$$

其中: $Z$  表示分批后所有批次的总相似程度,是由被选中的批次中的客户订单相似程度叠加得到; $K$  表示分批后的批次,若最终所有订单分成 5 批则  $K$  值取 5; $k$  为分批后的某一个批次; $b_k$  为分批后某个批次的相似程度, $\delta_k$  为决策变量,当第  $k$  批被选中时, $\delta_k$  取值为 1,没被选中时取值为 0; $v_t$  表示订单编号为  $t$  的订单中货物总体积, $w_t$  表示订单编号为  $t$  的订单中货物总重量, $\theta_{tk}$  为决策变量,当订单  $t$  被选中到第  $k$  批中时, $\theta_{tk}$  取值为 1,反之则取值为 0; $\text{Batch}_k$  表

示分批之后的第  $k$  批的订单集合,即第  $k$  批中包含的所有订单信息, $O$  表示为未分批订单的订单集合。

目标函数(1)表示订单分批后的总目标是实现订单相似度之和的最大值;式(2)表示控制变量的取值范围为  $\{0,1\}$ ,当第  $k$  批被选中时  $\delta_k$  为 1,否则为 0;式(3)表示分批后的每个批次的订单总体积不得超过拣选车的容积  $V$  的限制;式(4)表示控制变量的取值范围为  $\{0,1\}$ ,当订单  $t$  被归入到批次  $k$  时,  $\theta_{tk}$  为 1,否则为 0;式(5)表示分批后的每个批次的订单总中联不得超过拣选车的载重  $W$  的限制;式(6)表示所有批次中的订单的并集等于未分批订单集合  $O$  本身,即所有订单必须要都被拣选到;式(7)表示所有批次中的订单的交集必须是空集,即一个订单只能属于一个批次,不能再被拆分。

## 2 算法设计

K-均值算法是划分聚类算法之一,它的实现简单高效,虽然因为运行效率而降低了聚类精度,和一些智能算法相比不够精确,但在实际操作中并不会太大的误差,所以被广泛应用在实际生产中。K-均值算法一般以距离作为数据对象间相似性度量的标准,将相似程度高的对象聚合在一起代表一个聚类<sup>[11]</sup>,较适用于以坐标距离作为划分的数据集,例如仓库中的货位坐标信息。然而,K-均值算法也有其局限性,首先,该算法中的  $K$  值是人工给定或是系统随机的,因为事先不知道需要将数据集分成多少类;其次,需要人工确定初始聚类中心,再对初始划分进行优化,初始聚类中心的选择会对聚类结果产生明显影响<sup>[12]</sup>。基于上述缺点,本文提出一种改进遗传 K-均值算法来求解上述模型。

### 2.1 综合距离密度最优的初始聚类中心

步骤 1:计算订单  $O_t$  的中心点坐标,一个订单中有  $p_t$  种货物,每种货物数量为  $n_a$ ,那么该订单的中心点坐标为:

$$X_t = \frac{\sum_{a=1}^{p_t} n_a \times x_a}{\sum_{a=1}^{p_t} n_a} \quad (8)$$

$$Y_t = \frac{\sum_{a=1}^{p_t} n_a \times y_a}{\sum_{a=1}^{p_t} n_a} \quad (9)$$

步骤 2:计算两个订单之间的欧式距离,根据步骤 1 计算出的订单中心点坐标,计算两两中心点之

间的欧式距离  $d(t,i)$ :

$$d(t,i) = \sqrt{(X_t - X_i)^2 + (Y_t - Y_i)^2} \quad (10)$$

步骤 3:计算截断半径  $d_j$ ,就是计算出所有订单中心点之间的平均距离:

$$d_j = \frac{\sum_{i=1}^s \sum_{t=1}^s d(t,i)}{s^2} \quad (11)$$

步骤 4:计算订单  $O_t$  的密度  $\rho_t$ ,将所有订单的中心点作为画到坐标轴上为圆心,以截断距离为半径作圆,包含在该圆内的中心点坐标的个数就是该订单的密度,圆内的中心点数量越多,密度越大,该订单就越可能成为初始聚类中心:

$$\rho_t = \sum_{i=1}^{t-1} \alpha_i + \sum_{i=t+1}^{s-1} \alpha_i \quad (12)$$

$$\alpha_i = \begin{cases} 1, & \text{订单 } i \text{ 的中心点坐标位于圆内或圆上} \\ 0, & \text{订单 } i \text{ 的中心点坐标位于圆外} \end{cases} \quad (13)$$

步骤 5:计算订单  $O_t$  到更高密度点的最小距离  $l_t$ :

$$l_t = \min(\gamma_{\rho_i - \rho_t} \times d(t,i)) \quad (14)$$

$$\gamma_{\rho_i - \rho_t} = \begin{cases} 1, & \rho_i - \rho_t > 0 \\ 0, & \rho_i - \rho_t \leq 0 \end{cases} \quad (15)$$

步骤 6:计算密度和最小距离的乘积  $m_t$ ,该值越大,就说明该中心点被选为初始聚类中心的可能性越大:

$$m_t = \rho_t \times l_t \quad (16)$$

### 2.2 基于 K 值优化的改进遗传 K-均值聚类算法

相比于传统聚类算法,改进遗传算法的  $K$  不需要事先确定,而是利用遗传算法全局搜索替代穷举。步骤 1:对十进制的聚类中心数  $K$  进行编码。由杨善林等<sup>[13]</sup>对  $K$  值问题的研究可以得到  $K$  的范围为大于 0 且小于样本个数的算数平方根,例如有 100 个样本订单,则  $0 < \max(K) \leq 10$ 。而十进制数 10 对应的二进制数是 1010,所以染色体的长度为 6 位,前 4 位基因是十进制类别数转换成的二进制数,第 5 位基因是十进制类别数,最后一位基因是适应度函数。

步骤 2:初始化种群。确定种群的规模  $s$ ,一般  $s$  的取值范围为  $[30,150]$ ;选择交叉概率  $s_c = 0.5 \sim 1.0$  和变异概率  $s_m = 0.001 \sim 0.050$ 。

步骤 3:初始聚类,把种群中每个染色体解码成十进制,即对应的类别数  $K$ <sup>[14]</sup>。对于每个订单,根据之前初始聚类中心优化得到的  $m_t$  的大小按照先后顺序选取  $K$  个数据作为类中心。将种群中的其它订单按照新的分配方式进行分配:

$$\begin{aligned} (X_{k'} - X_t)^2 + (Y_{k'} - Y_t)^2 &\leq \\ (X_{k''} - X_t)^2 + (Y_{k''} - Y_t)^2 \end{aligned} \quad (17)$$

其中:  $t \in s$  且  $k' \neq k''$ ,  $k'$  和  $k''$  都表示为某一批。该公式表示若订单  $O_t$  的中心点距离  $O'_{k'}$  最近, 则把订单  $O_t$  归到以订单  $O'_{k'}$  为聚类中心的类中, 反之归到以订单  $O'_{k''}$  为聚类中心的类中。依次对所有没有被选为初始聚类中心的订单进行上述操作。

步骤 4: 适应度函数的设计, 需要同时考虑组内间距最小和组间间距最大的综合最优。组内间距, 即同一批中的订单中心点与聚类中心的距离和:

$$E_K = \sum_{k'=1}^K \sum_{t \in L_{k'}} \sqrt{(X_{k'} - X_t)^2 + (Y_{k'} - Y_t)^2} \quad (18)$$

组间间距, 即不同批次聚类中心之间的距离:

$$D_K = \max_{k', k''=1}^K \sqrt{(X_{k'} - X_{k''})^2 + (Y_{k'} - Y_{k''})^2} \quad (19)$$

适应度函数:

$$fitness(K) = \frac{1}{K} \times \frac{E_1}{E_K} \times D_K \quad (20)$$

根据适应度公式,  $K$  值越大则适应度函数值越小, 所以要尽量使得  $K$  值接近于最小值,  $k'$  和  $k''$  都为其中的某一类,  $L'_{k'}$  为该类中的所有订单,  $t$  为该类中的一个订单; 当订单信息已知时,  $E_1$  是确定的,  $E_K$  随着  $K$  值的增加而减小, 适应度函数值随着  $K$  的增加而增加, 即  $D_K$  随着  $K$  值的增加而增加。所以该函数能实现分类数和组内组间间距的综合最优。

步骤 5: 遗传操作。

a) 选择因子: 对于染色体的选择, 采用精英策略和轮盘赌相结合的选择方法<sup>[15]</sup>。精英策略是把群体在进化过程中迄今出现的最好个体。轮盘赌是依据个体的适应度值计算每个个体在子代中出现的概率, 并按照此概率随机选择个体构成子代种群。

b) 交叉因子: 交叉算子是对种群中选出的两个个体进行交叉操作。

c) 变异因子: 变异是指对群体中个体串某些基因座上的基因值产生变动<sup>[16]</sup>。本例中对编码后的  $K$  值进行 0 与 1 互换的变异, 使  $K$  值产生变动。

d) 生存因子: 需要判断迭代之后的分批组合是否符合拣选车的载重和容积的限制, 选择拣选车的载重和容积作为生存因子<sup>[17]</sup>。

### 2.3 拣选路径策略选择

拣选路径选择策略主要包括穿越型路径策略、

中点型路径策略、往返型路径策略、组合型路径策略等<sup>[18]</sup>。不同策略的适用环境有所差别, 穿越式路径策略适用于货物密度高但数量少的情况, 拣货人从通道一端进入, 同时拣选通道两侧货架上的物品, 最后从通道另一端离开, 在返回出入口之前, 拣货人会走遍所有包含拣取位置的巷道。

穿越型路径策略拣选示意图如图 2 所示, 拣选员从仓库左下角集货点出发, 遇到有需求货物的货架就从一端进入, 拣选货物并从另一端离开, 当一整排货架都没有需求货物的时候就忽略该通道, 依次进行判断和拣选, 最后回到集货点, 完成本次拣选, 整个拣选路线不往返。

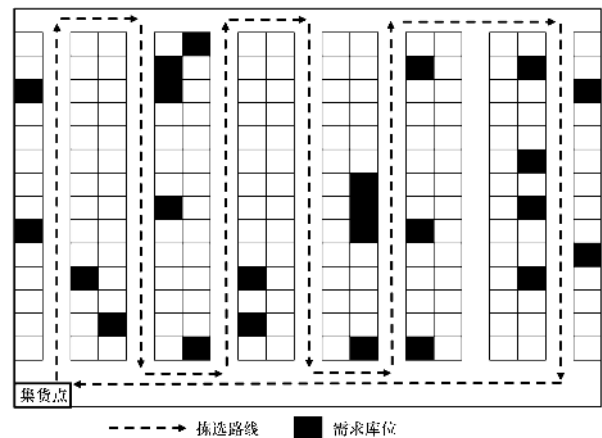


图 2 穿越型路径策略拣选示意图

## 3 仿真实验

### 3.1 订单分批拣选实例描述

本文以某机械设备生产公司售后服务备件仓为例, 对该仓库一段时间内的订单需求数据进行仿真实验。该仓库为一个双分区仓库, 货架纵向排列, 前分区共有 12 排单层货架, 靠墙两侧为单排货架, 中间为双排货架和巷道的间隔排列, 每排货架有 10 个货位, 货位简化为长宽都为 1.5 m 的矩形, 前后分区布局相同, 两个分区中间贯穿一条横向巷道, 巷道宽同样为 1.5 m。拣选小车的容积和载重限制分别量化为 600 dm<sup>3</sup> 和 120 kg。拣选员在某个时间段内一共收到 40 个客户订单, 总体积为 1298 dm<sup>3</sup>, 总重量为 544 kg, 每个订单的货物需求不完全相同且单个订单的需求货物总体积和总重量均不超过拣选小车的限制。

如表 1 所示, 每个订单货物坐标的中点为中心点坐标,  $X$  和  $Y$  值为中心点的横纵向位置, 订单体积和重量均为该订单中所有货物的体积和重量的和。

表1 订单信息表

订单编号	中心坐标/m	体积/dm <sup>3</sup>	重量/kg	订单编号	中心坐标/m	体积/dm <sup>3</sup>	重量/kg
1	(9.5,11.0)	30.0	12.0	21	(4.5,9.0)	50.0	25.0
2	(8.2,8.2)	25.0	11.0	22	(13.0,2.0)	34.0	10.0
3	(3.0,3.0)	30.0	15.0	23	(18.0,9.0)	20.0	8.0
4	(6.0,2.0)	25.0	10.0	24	(3.0,2.0)	30.0	9.0
5	(12.0,6.7)	25.0	7.0	25	(1.0,6.0)	23.0	8.0
6	(10.0,1.0)	26.0	14.0	26	(9.0,10.0)	19.0	7.0
7	(9.0,10.0)	18.0	8.0	27	(18.0,17.0)	30.0	13.0
8	(9.5,2.0)	55.0	25.0	28	(6.0,13.0)	40.0	11.5
9	(10.0,6.0)	52.0	31.0	29	(7.0,20.0)	60.0	20.0
10	(16.0,10.0)	21.0	12.0	30	(12.0,16.0)	12.0	9.0
11	(10.0,20.0)	46.0	22.0	31	(12.0,1.0)	30.0	7.0
12	(6.0,1.0)	37.0	17.0	32	(16.0,16.0)	31.0	20.0
13	(16.0,15.0)	15.0	3.0	33	(12.0,19.0)	17.0	10.0
14	(4.0,1.0)	25.0	7.0	34	(16.0,7.0)	16.0	6.5
15	(6.5,16.0)	58.5	25.0	35	(18.0,7.0)	42.0	29.0
16	(13.0,17.0)	35.0	12.0	36	(1.0,15.0)	50.0	23.0
17	(18.0,1.0)	37.0	19.0	37	(1.0,1.0)	47.0	13.0
18	(12.0,14.0)	37.0	12.0	38	(10.0,15.0)	16.0	7.0
19	(16.0,15.0)	12.0	5.0	39	(18.0,17.0)	49.0	17.0
20	(4.0,19.0)	43.5	17.0	40	(12.0,17.0)	29.0	7.0

### 3.2 确定初始聚类中心

根据表1中每个订单的中心坐标,绘制相应的散点图,计算两两订单的中心点之间的欧氏距离。表2为其中10个订单中心点之间的距离表,例如订单1与订单2中心点距离为3.1 m,与订单3中心点距离为8.1 m。

表2 部分订单中心点距离 m

编号	1	2	3	4	5	6	7	8	9	10
1	0.0	3.1	8.1	11.1	4.8	10.0	1.1	9.0	5.6	2.4
2	3.1	0.0	5.2	8.1	3.6	7.4	2.0	6.3	2.5	3.9
3	8.1	5.2	0.0	5.3	4.9	2.7	7.0	1.6	3.2	7.8
4	11.1	8.1	5.3	0.0	9.7	7.1	10.0	6.5	5.7	11.8
5	4.8	3.6	4.9	9.7	0.0	5.9	4.2	5.1	4.5	3.3
6	10.0	7.4	2.7	7.1	5.9	0.0	9.1	1.1	5.8	9.2
7	1.1	2.0	7.0	10.0	4.2	9.1	0.0	8.0	4.5	2.7
8	9.0	6.3	1.6	6.5	5.1	1.1	8.0	0.0	4.7	8.3
9	5.6	2.5	3.2	5.7	4.5	5.8	4.5	4.7	0.0	6.1
10	2.4	3.9	7.8	11.8	3.3	9.2	2.7	8.3	6.1	0.0

求得两两订单中心点之间距离后计算截断距离 $d_j$ 并依次选取每个中心点,以截断距离为圆心作圆,统计包含在圆内的点的个数,即为该中心点对应订单的密度。中心点密度示意图如图3所示,以中心点坐标为(6,2)的订单为例,其对应的密度为5,即以该订单的中心点为圆心,以截断距离为半径的

圆包含了其它4个订单的中心点。之后计算该点到更高密度中心点的最小距离,与密度的乘积就是该点所对应的订单被选为初始聚类中心的概率。

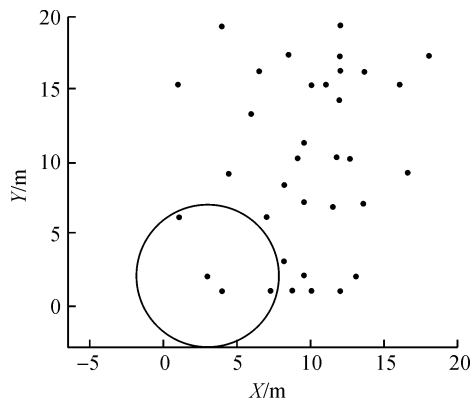


图3 中心点密度示意图

### 3.3 改进遗传 K-均值算法

确定改进遗传 K-均值算的参数为:交叉算子为随机选择交叉点的单点交叉,交叉概率 $s_e=0.7$ ,变异算子为单点变异,变异概率 $s_m=0.01$ ,获取最佳聚类数之后再配合优选的初始聚类中心进行聚类分析。

根据图4所示,结合拣选小车的容积和载重约束。当聚类数小于4时,平均每批订单的体积及重量均超过限制。当聚类数为5时适应度函数的值最大,所以选择将样本订单分成5批。确定初始聚类中心及分批数之后,进行订单分批的操作。

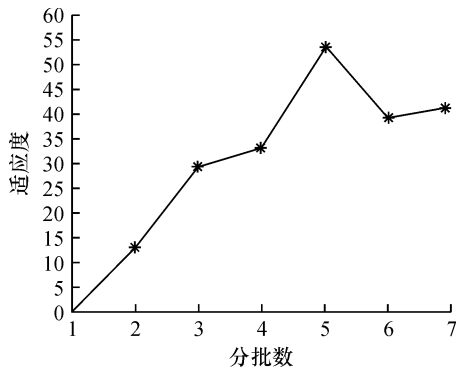


图 4 分批数对应的适应度函数值

运用改进的聚类算法,经过 7 次迭代获得局部最优解如图 5 所示,图中共有 6 种不同形状的点,其中无填充色的圆表示为迭代之后的聚类中心,其余 5 种形状,相同形状的点表示为同一类点,即该中心点所对应的订单是归在同一批中进行拣选的。

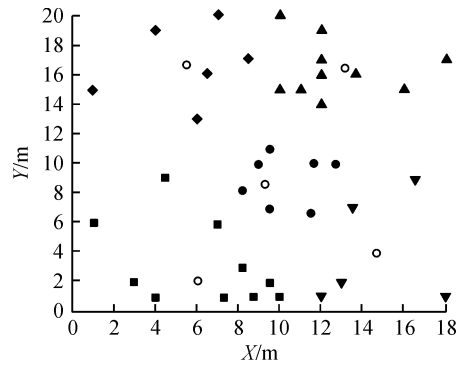


图 5 本文算法第 7 次迭代结果

为验证本文方法的有效性,选择 K-均值算法对该订单数据进行简单分批,指定能产生较优分批结果的初始聚类中心并确定分批数为 5,得到分批结果后用穿越型路径策略分别计算两种算法的总拣选距离长进行比较。表 3 表示两种算法对于订单分批拣选的验证结果。

表 3 两种算法分批结果对比

算法	迭代数	分批数	批次号	每批所包含的订单号	单批拣选路程/m	总路程/m
K-均值算法	4	5	1	11,15,20,28,29,36	58.0	321.0
			2	13,18,19,27,30,32,33,38,39,40	39.0	
			3	4,9,14,21,24,25	57.0	
			4	1,2,5,7,10,16,23,26,34,35	104.0	
			5	3,6,8,12,17,22,31,37	63.0	
改进遗传 K-均值算法	7	5	1	15,16,20,28,29,36	58.5	277.5
			2	11,13,18,19,27,30,32,33,38,39,40	39.0	
			3	1,2,5,7,9,10,21,26,34	93.0	
			4	3,4,6,8,12,14,24,25,37	63.0	
			5	17,22,23,31,35	24.0	

本实验样本大小为 40,结果表明,K-均值算法迭代 4 次取到局部最优,总的拣选路程为 321.0 m,而本文算法优选了初始聚类中心并求得最优的分批数 K 为 5,经过 7 次迭代取得局部最优解,分批后总的拣选路程为 277.5 m,缩短了 14%的拣选路程,提升效果较为显著。同时算法复杂度也有所增加,因为聚类算法的复杂度上界与样本大小、分批数和迭代次数线性相关,本文算法与传统聚类算法相比,迭代次数由 4 次增至 7 次,导致算法复杂度增加 1.75 倍。

#### 4 结 论

本文针对售服仓多品种小批量订单的分批问题,以订单相似性为切入点,构建了订单分批优化模型。针对传统 K-均值算法中聚类中心人为确定而影响聚类结果的缺陷,采用了初始聚类中心的优化方法,按照密度和最小距离综合最优的原则,优选了初始聚类订单;对于 K 值确定问题,采用了改进遗

传算法,通过遗传迭代来确定 K 值后再进行聚类,最后用穿越型路径方法计算拣选距离长短来衡量分批结果的优劣。仿真实验结果表明,本文采用的方法迭代次数较多,不易陷入局部最优,分批效果也更好,能够显著缩短拣选距离,减轻拣选作业的工作强度,验证了该算法的有效性。

本文方法消除了函数对于初始聚类的依赖,但无法避免聚类过程中对于聚类中心生成次序的依赖性。算法采用迭代方法,当面对大规模数据量时,算法需要不断进行分类并重新调整新的聚类中心,会造成算法运行时间成倍增加,算法复杂程度增加,上述问题有待于进一步的研究。

#### 参考文献:

- [1] 赵兰. 基于拣选和分拣时间的订单分批优化方法研究与应用[D]. 武汉:华中科技大学,2015:2-3.
- [2] Petersen C G, Aase G. A comparison of picking,

- storage, and routing policies in manual order picking [J]. *International Journal of Production Economics*, 2005, 92(1):11-19.
- [3] 李诗珍,王转,张克诚. 配送中心拣货作业中的订单分批策略研究[J]. *物流技术*, 2002(4):31-33.
- [4] Koster R D, Le-Duc T, Roodbergen K J. Design and control of warehouse order picking: A literature review [J]. *European Journal of Operational Research*, 2007, 182(2):481-501.
- [5] 王旭坪,张琨,马骏. 考虑完成期限的电子商务在线订单分批模型及算法[J]. *管理科学*, 2014(6):103-113.
- [6] 文坚,马士华. 基于时间延迟的订单分批策略研究[J]. *物流技术与应用*, 2005, 9(4):92-95.
- [7] 李晓杰. 移动货架仓库系统中货位分配和订单分批联合优化研究[D]. 北京:清华大学, 2016:8-13.
- [8] 马廷伟,雷全胜,李军,等. 粒子群算法及其在订单分批中的应用研究[J]. *物流科技*, 2016, 39(3):4-8.
- [9] 李诗珍,杜文宏. 基于聚类分析的订单分批拣货模型及启发式算法[J]. *统计与决策*, 2008(12):53-56.
- [10] 胡小建,韦超豪. 基于 Canopy 和  $k$ -means 算法的订单分批优化[J]. *合肥工业大学学报(自然科学版)*, 2017, 40(3):414-419.
- [11] 薛卫,杨荣丽,赵南,等. 空间密度相似性度量  $K$ -means 算法[J]. *小型微型计算机系统*, 2018, 39(1):53-57.
- [12] 吴凤慧,成颖,郑彦宁,等.  $K$ -means 算法研究综述[J]. *数据分析与知识发现*, 2011, 27(5):28-35.
- [13] 杨善林,李永森,胡笑旋,等.  $K$ -means 算法中的  $k$  值优化问题研究[J]. *系统工程理论与实践*, 2006, 26(2):97-101.
- [14] 刘婷,郭海湘,诸克军,等. 一种改进的遗传  $k$ -means 聚类算法[J]. *数学的实践与认识*, 2007, 37(8):104-111.
- [15] 张奇飞,林剑,王兆锐,等. 基于改进遗传算法的物流路径优化方法[J]. *物流技术*, 2018(1):78-81.
- [16] Bharathi C, Rekha D, Vijayakumar V. Genetic algorithm based demand side management for smart grid[J]. *Wireless Personal Communications*, 2017, 93(2):481-502.
- [17] 卢子甲,韩义民,张少卿. 基于遗传算法的配送中心订单拣选路径优化案例研究[J]. *物流技术*, 2013, 32(17):228-230.
- [18] 冉文学,刘会娟. 动态立体仓库中穿梭车货到人多目标拣选路径优化[J]. *物流工程与管理*, 2017, 39(6):69-73.

## Multi-variety small batch order batching method based on improved genetic $K$ -means algorithm

SHAO Zeyi, DONG Baoli

(Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, Hangzhou 310018, China)

**Abstract:** Aiming at the problem of multi-variety small batch order problem with multiple types of goods and scattered storehouses, a batch sorting optimization model with the aim of the highest order similarity was constructed, and the improved genetic  $K$ -means algorithm was used to solve the model. In traditional  $K$ -means algorithm, the  $K$  value is artificially determined, resulting in the large error of clustering results. For this shortcoming, the density and minimum distance were used to determine the multiple initial cluster centers, and the improved genetic  $K$ -means algorithm was used to determine the optimal batch number for optimizing the order in batches. After the batch results were obtained, the picking distance was calculated through the traversing path method and compared with the simple batching. The order picking of after-sale service spare parts warehouse of a mechanical equipment production company was taken as the object for simulation experiment. The result showed that, method is effective.

**Key words:** order picking in batches; initial clustering optimization;  $K$ -means algorithm; genetic algorithm; traversing path

(责任编辑: 康 锋)