**基于蚁群算法的多配送中心的车辆调度问题的研究**

**一、问题介绍**

车辆调度问题可以描述为：在一个存在供求关系的系统中，有若干台车辆、若干个配送中心和客户，要求合理的安排车辆的行驶路线和时间，在给定的约束条件（载重、容积等）下，把客户需求的货物从配送中心送到客户手中，并使得目标函数最优。

车辆的调度问题可以用如下的网络模型来表示：设G=(V,E,A)是一个连通的网络，V是顶点集(比如客户、配送中心等)，E是无向的边集，A是有向的弧集。E的边和A的弧均被赋值(表示路程、时间费用等)，分别是V、A、E的子集。在满足约束条件(客户的需求、车辆的载重、车辆的容积等)下，使得目标函数最优，目标函数可以取配送的费用最低、配送时间最短、车辆的总行程最短、配送的车辆最少等。

根据配送中心的多少，车辆调度问题可以分为单配送中心的车辆调度问题和多配送中心的车辆调度问题。本文将要研究的是一个多配送中心的车辆调度问题。

**二、建立模型**

现实中的多配送中心的车辆调度问题是十分复杂的，为了方便模型的建立和求解，我们对实际调度中的一些问题进行了简化。现对所研究的多配送中心的车辆调度问题作出如下的假定：

1. 每个配送中心的位置已知，且配送中心的货物能够满足所有顾客的要求，有足够多的车辆来进行配送。
2. 每个客户的需求一定，且需求的货物可以混装；每个顾客所需的货物不能超过每辆车的载重限制；每个顾客所需的货物只能由一辆车来进行运输。
3. 车辆有载重限制；车辆从配送中心出发，最终要回到配送中心。
4. 配送中心到顾客的距离以及顾客到顾客的距离通过坐标计算得到，也就是说是一个已知条件。

通过查找相关文献，发现解决多配送中心的车辆调度问题一般有两种方法：①分解法。将多配送中心的车辆调度问题转化成单配送中心的车辆调度问题，然后进行优化组合。关键在于多配送中心分离策略的选择和最后解的优化组合方案的选择。②整体法。将多配送中心的车辆调度问题整体看成一个复杂的组合优化问题进行研究。由于不能事先确定配送中心位置，所以随机选择的结果可能会导致问题的解空间很大。设计一个合理的算法使得问题的解空间尽可能的小成为了关键。

本文采用分解法来解决多配送中心的车辆调度问题。首先要解决的是确定配送中心的位置。在这里，采用的是最近距离分配法。考虑到多配送中心的车辆调度问题多以配送车辆的总行程最短为目标，所以我们可以先计算各个顾客到各个配送中心的距离，对顾客而言，他会选择一个离他最近的配送中心，由此，给每个顾客分配相应的配送中心。设表示第个顾客到第个配送中心的距离，其中,表示配送中心的总数。记集合,，则第个顾客由第个配送中心来进行配送。

基于以上分析，建立多配送中心的车辆配送模型，相关符号说明：

 表示配送中心的个数

 表示顾客需求点的个数

 第个配送中心的配送车的个数

 第个配送中心的第辆车的载重

 第个配送中心的第辆车的车容积

 第个需求点的货物重量

 第个需求点的货物体积

 需求点到需求点的距离

 第个配送中心的第辆车配送的需求点的个数

 第个配送中心的第辆车配送的路径

 第个配送中心的第辆车配送的第 个需求点，表示配送中心

本文研究以配送的路径最短为目标函数，建立如下的数学模型：

 (1)

  (2)

 (3)

 (4)

 (5)

 (6)

 (7)

在上述模型中，(1)为目标函数：总的配送路线最短；(2)式表示每条路径上的客户的货物总重量不得超过配送车辆的载重；(3)式表式每条路径上的客户的货物体积不得产过配送车辆的车厢容积；(4)式表式每个顾客的需求都能得到满足；(5)式表示每条路径的客户组成；（6）式限制了每个顾客只能由一个配送车辆来进行配送；(7)式表示当第个配送中心的第辆车服务的客户数大于等于1时，即，说明这辆车参与了配送，则，当第个配送中心的第辆车服务的客户数大于等于1时，即，表示该车辆未参与服务，则。

1. **蚁群算法设计**

本文采用分解法来解决多配送中心的车辆调度问题，确定配送中心的位置采用的是最近距离分配法。确定配送中心以后，对需求点进行分类汇总，将需求点放入分别属于每一个配送中心的集合中。

确定配送中心及其所要配送的需求点之后，开始对每一个配送中心的配送路线进行算法设计。 首先是相关参数的设定，蚁群规模Pop为60，迭代次数MAXGEN为50，重要度系数Alpha为1，能见度系数Beta为1,挥发度系数Rho为0.15，信息更新参数Q为15，车辆载重量W为9，车厢容积T为10，启发因子Eta设为距离的倒数，初始的信息素矩阵为单位矩阵。

开始迭代时，每只蚂蚁分别从各个配送中心出发，根据转移概率





其中，分别表示信息素和启发式因子的相对重要程度。计算出可达到顾客的选择概率，通过遗传算法的轮盘赌法来选择要访问的下一个顾客，即各个可选顾客的选择概率在区间[0,1]排列开，然后产生一个0-1之间的随机数，根据随机数出现在上述哪一个概率区域来确定下一个要访问的客户。确定下一个到达的需求点后，计算一下前面所有顾客需求货物的重量，如果大于车辆的载重量，则返回；如果没有超过车辆的载重量，则把该需求点作为下一个到达的需求点。

当所有蚂蚁完成一次迭代后，然后采用2-opt的方法对最优解进行更新。对由转移概率确定的路线，每两个需求点进行交换，然后计算新路径的长度，与交换之前相比，有没有优化，如果优化了，就更新路径，否则，路径不变。接下来各个路径上的信息素要进行更新：

整个蚁群算法的流程如图1所示：

图一

1. **算法的结果分析**

通过分解法将多配送中心的车辆配送问题转化成单配送中心的车辆调度问题，采用最近距离分配法确定配送中心的位置，蚁群算法求得该问题的近似最优解。

得到的最优路径为:

从10配送中心出发：



从7配送中心出发：



如图2：

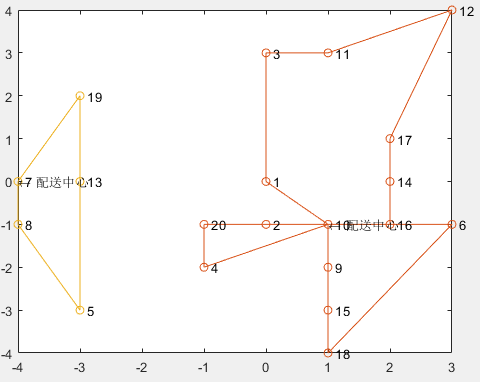


图2

注：箭头所指的坐标点为配送中心的位置

利用MATLAB仿真，得到每一次迭代得到的最优解以及每一次迭代每个可行解的平均值，如图3：

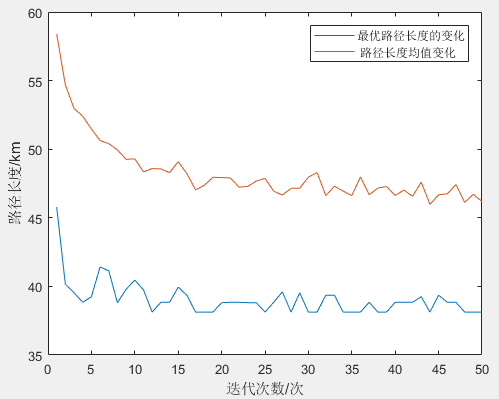


图3

得到的最优路径的总长度为：38.1263

采用分解法求解多配送中心的车辆调度问题时，在确定配送中心的位置上由于使用的是最近距离分配法，虽然有简单快捷的特点，但只单独考虑了各个配送点与配送中心距离的因素，而没有考虑配送点之间的距离因素。分解法不易发现全局最优，所以不适合大规模的车辆调度问题。本文所采用的案例所涉及的顾客较少，问题的规模不大，因而也可以用分解法来进行路径优化。