

文章编号:1001-5132(2007)04-0503-04

# 面向中小企业集群的非满载车辆调度问题研究

夏文明<sup>1</sup>, 李国富<sup>2</sup>, 朱双东<sup>1</sup>, 叶飞帆<sup>2</sup>

(1.宁波大学 信息科学与工程学院, 浙江 宁波 315211; 2.宁波大学 工学院, 浙江 宁波 315211)

**摘要:** 基于中小企业集群的特点, 给出最少配送车辆数的计算模型, 从最少配送车辆数、最短距离、各路线里程平衡和时间窗等方面对集送一体化车辆调度问题建立模型, 用改进的模拟退火算法求解该调度模型. 通过仿真实例验证了该算法的可行性, 且符合实时性的要求.

**关键词:** 车辆调度; 非满载; 时间窗; 配送车辆数

中图分类号: TP183

文献标识码: A

现在企业之间的物流配送特别是中小企业集群之间的物流配送一般是小范围、近距离、小批量、多批次、为多用户服务的经济活动, 这时每个任务点的货物量小于车辆容量, 用一辆车即可执行任务. 本文将基于这些特点, 建立最少配送车辆数的计算模型, 从最少配送车辆数、最短距离、各路线里程平衡和时间窗等方面对集送一体化车辆调度问题建立模型. 目前, 一些研究车辆调度问题的文献[1-3]没有给出完成任务所需要的最少车辆数, 而是预先给定车辆数, 在这个基础上优化调度路线. 合理的配送车辆数应该根据具体任务和约束条件确定, 而不应预先给定. 而且这些文献主要从最短距离, 最短时间来优化调度路线, 没有考虑司机的疲劳驾驶, 本文在这些方面将予以完善.

## 1 问题描述

有里程和时间窗约束的非满载车辆调度问题

可描述为: 某车场拥有一定数量载重量均为  $Q$  的车辆, 现有  $n$  项货物运输任务要完成, 每个任务都有自己的集货点和送货点, 要求车辆在一定的集货点装货后运至一定的送货点卸货, 已知每个任务的货运量  $G_i$  满足  $G_i < Q$ , 且  $1/2Q < G_i < Q$ , 这时不同的任务难以用一辆车来完成. 要求每项任务在规定的时间内完成, 求满足货运需求的费用最小的路线调度方案, 且要求该方案能使各车辆的工作量比较均衡, 即不能出现有的车辆很快就完成了给定的任务, 而有的车辆需要很长时间才能完成给定任务的状况. 为不造成疲劳驾驶, 要求每个司机的运输里程不超过给定的值.

## 2 模型建立

### 2.1 最少车辆数估计

每个任务都派一辆车出去固然可以完成任务, 但如果某车辆在完成某项任务后能够在不违反约

收稿日期: 2007-06-06.

宁波大学学报(理工版)网址: <http://3xb.nbu.edu.cn>

基金项目: 国家自然科学基金(70540023); 浙江省自然科学基金(M703100, Z604342); 教育部科学技术研究重点项目(205066); 宁波大学教授基金(XJ0709004).

作者简介: 夏文明(1982-), 男, 湖北咸宁人, 在读硕士研究生, 主要研究方向: 车辆调度与优化组合. E-mail: xwmzip2006@yahoo.com.cn

束的条件下,直接再去完成另一项任务,一般来说,这比每个任务都从车场派一辆车出去的成本要低得多,因此完成某些任务派出去的车辆数越少成本越低.为减少成本,首先估计完成所有任务所需要的最少车辆数  $m$ .

设有  $n$  项运输任务,规定第  $j$  项任务  $T_j (j=1, 2, \dots, n)$  开始执行的时间窗为  $[E_j, L_j]$ , 任务  $T_i$  完成的时刻为  $O_i$ , 允许某运输车辆执行完任务  $T_i$  后再转回来执行任务  $T_j$ , 只要该车辆的转移时间  $T_{ij}$  (从任务  $i$  的完成地点到任务  $j$  出发地点所用的时间) 在任务  $T_j$  的开始执行时间窗内, 即:

$E_j - O_i + T_{ij} \leq L_j$ . 显然, 完成  $n$  项运输任务最多需要  $n$  辆车. 定义  $Y_{ij}$  为:

$$Y_{ij} = \begin{cases} 1 & \text{完成任务 } T_i \text{ 后再转去完成任务 } T_j, \\ 0 & \text{否则 } (i, j=1, 2, \dots, n), \end{cases}$$

则需要派出的最少车辆数  $m$  为:

$$m = n - \sum_{i=1}^n \sum_{j=1}^n Y_{ij}. \quad (1)$$

## 2.2 约束条件

为便于建立有时间窗、非满载封闭车辆调度问题的数学模型, 定义变量  $X_{ij}$  如下:

$$X_{ij} = \begin{cases} 1, & \text{车辆由货栈 } i \text{ 驶向货栈 } j, \\ 0, & \text{否则.} \end{cases}$$

假设货栈  $i$  的集合为  $N$ ,  $W_{ij}$  表示车辆由货栈  $i$  到货栈  $j$  的费用, 则有下面的约束条件.

(1) 运输费用  $Z_1$  最小.

$$\min Z_1 = \sum_{i=0}^n \sum_{j=1}^n W_{ij} X_{ij}. \quad (2)$$

(2) 容量约束, 即每辆车所完成的任务的货物量不能超过车的载重量.

$$\sum_{i=1}^n G_i Y_{ik} \leq Q, \quad i \in N, \forall k=1, 2, \dots, m.$$

用容量约束的目标函数为:

$$Z_2 = \sum_{i=1}^n C_1 \max(G_i Y_{ik} - Q, 0). \quad (3)$$

权值  $C_1$  视对容量约束的重视程度而定, 为了严格满足容量约束, 应有  $C_1 = +\infty$ , 但考虑到计算机处理的不便,  $C_1$  可取一个适当大的正数.

(3) 时间窗口约束. 时间窗口是随着零库存 (Just In Time, JIT) 的提出而产生的, 它要求零件只在正好需要的时间范围内送达生产线, 以减少库存, 降低成本. 假设  $S_i$  表示车辆到达货栈  $i$  的时刻,  $D_i$  表示车辆在货栈  $i$  等待的时间,  $D_{ij}$  表示车辆由货栈  $i$  行驶到货栈  $j$  的时间, 且假设货栈  $i$  的装卸货速度为  $V_i$ , 则有:

$$S_0 = 0, \quad i=0 \text{ 表示车辆从车场出发};$$

$$\text{若 } X_{ij} = 1 \text{ 则有 } S_i + D_i + D_{ij} + G_i/V_i = S_j, \quad i \neq j;$$

$D_i = \max\{E_i - S_i, 0\}$ , 即若车辆在  $E_i$  之前到达货栈  $j$ , 则车辆需在此等待; 若车辆在  $L_i$  之后到达货栈  $i$ , 则要进行相应的处罚.

这里时间窗的目标函数  $Z_3$  为:

$$Z_3 = \sum_{i=1}^n C_2 \max(S_i - L_i, 0) + \sum_{i=1}^n C_3 \max(E_i - S_i, 0), \quad (4)$$

其中权值  $C_2, C_3$  的取值机制与  $C_1$  相同.

(4) 任务均衡约束. 总成本最小的路线不一定能使各车辆的工作量均衡, 若其中某些车辆工作量较大, 可能会使司机疲劳驾驶导致发生事故, 转而使成本增加, 故需对均衡提出一定的要求, 即各车辆间运输路线的长度差不大于某给定值  $L$ , 即  $(f_{\max} - f_{\min}) < L$ , 其中  $f_{\max}$  为  $m$  辆车中最长的运输路程,  $f_{\min}$  为  $m$  辆车中最短的运输路程, 则均衡约束的目标函数可表示为:

$$Z_4 = C_4 \max(f_{\max} - f_{\min} - L, 0), \quad (5)$$

其中  $C_4$  为惩罚因子, 它与权值  $C_1$  的取值机制相同. 为不造成疲劳驾驶, 还规定每辆车的行驶路程不超过一个给定的值  $R$ , 即  $f_{\max} \leq R$ . 目标函数可表示为:

$$Z_5 = C_5 \max(f_{\max} - R, 0), \quad (6)$$

其中  $C_5$  为惩罚因子, 它与权值  $C_1$  的取值机制相同.

综上, 可得有里程和时间窗约束的非满载车辆调度问题的能量函数:

$$E = C_0 Z_1 + Z_2 + Z_3 + Z_4 + Z_5, \quad (7)$$

其中  $C_0$  为  $Z_1$  的权值, 视  $Z_1$  在整个函数  $E$  中的比重

而定, 避免在收敛过程中  $Z_1$  被边缘化.

### 3 模型求解

为求精确解, 本文欲用改进的模拟退火算法优化函数  $E$ . 为了防止算法从全局最优跳回局部最优, 本算法设置了一个记忆因子  $p$ , 使它记住所有搜索到的局部最优中的最小值, 这样就算又跳回局部最优, 也可以通过记忆因子  $p$  重拾经过的全局最优.

#### 3.1 主要执行步骤

假设任务  $T_i$  为将货物从货栈  $i_1$  运输到  $i_2$ , 则  $T_i = \{i_1, i_2\}$  为一条有向边. 整个算法的框架如下:

(1) 构造一个初始可行解  $S_0 = \{0, T_1, 0, T_2, 0, \dots, 0, T_n, 0\}$ , 0 表示车场, 即  $S_0$  表示每项任务都派一辆车出去执行, 执行完后返回车场;

(2) 给定一个较大的起始温度  $T = T_0$ , 温度  $T$  的下降率  $\sigma$ , 最低温度  $\min T$  和一个浮点数  $p$ , 其中浮点数  $p$  是用来记住最小的局部最优点, 可防止程序跳过全局最优点后跳不回来了;

(3) 在  $S_0$  邻域内产生一个新解  $S_0'$ ;

(4) 令  $\Delta E = E(S_0') - E(S_0)$ , 计算能量函数差值;

(5) 如果  $\Delta E \leq 0$ , 令  $S_0 = S_0', p = S_0$ , 执行(7);

(6) 如果  $\Delta E > 0$ , 计算接受路线调整的概率  $\exp(-\Delta E/T) > 0$ , 如果

$$\min[1, \exp(-\Delta E/T)] > \text{random}(0, 1),$$

即路线调整接受函数在 0 到 1 之间, 则令  $S_0 = S_0'$ , 否则保留该值;

(7) 温度下降, 令  $T = \sigma T$ ,  $\sigma$  为大于 0 小于 1 的参数, 称为温度递减率,  $\sigma$  越大意味着搜索过程越慢. 一般取  $\sigma = 0.8 \sim 0.9999$ , 若  $T < \min T$  则算法终

止, 否则执行(3).

从算法流程上看, 在该算法中, 邻解的产生方法和邻域结构的构造在很大程度上会影响算法的效果. 本文欲用以下算法找邻解.

(1) 初始化  $S_0'$ , 令  $S_0' = \{\text{除首尾 2 个零外, 依次将 } S_0 \text{ 中的零去掉}\}$ , 即完成某项任务后不回车场, 直接去完成下 1 个任务;

(2) 如果  $\Delta E \leq 0$ , 则  $S_0 = S_0'$ , 否则将零补上, 即若再接着去完成另一件任务就会违反约束, 则需要再从车场派 1 辆车出去;

(3) 令整数  $i = 1, j = i + 1$ ;

(4) 判断  $j$  是否小于  $n$  ( $n$  为任务数), 如果是则交换  $S_0'$  中  $i, j$  位置处的元素得到  $S_0''$ , 如果不是则算法结束;

(5) 计算  $\Delta E = E(S_0') - E(S_0'')$ , 如果  $\Delta E > 0$ , 则  $S_0' = S_0''$ ;

(6)  $i = i + 1, j = j + 1$ , 执行(3).

### 4 模型仿真

设有 8 个货栈, 编号分别为 1, 2, ..., 8, 由车场 0 派出车辆执行任务, 车辆的容量为 10 t, 且规定  $t = 2, C_2 = 100, C_2' = 1000, C_3 = 10, C_3' = 100$ . 假设每辆车的行驶速度为  $50 \text{ km} \cdot \text{h}^{-1}$ , 运费为每公里 0.5 元, 装货卸货速度单位为  $\text{t} \cdot \text{h}^{-1}$ . 要求各车辆间运输路线的长度差不大于 8 h 的车程, 车场及各货栈的坐标  $(X_i, Y_i)$  由 C 语言函数库中的 random () 函数在  $[0, 100] \times [0, 100]$  的区域内随机产生, 各货栈的货运量  $G_i$  在  $(50, 100)$  内随机产生, 单位为吨, 相关数据见表 1, 其中比例为 1:10. 任务分别为:  $1 \rightarrow 8, 2 \rightarrow 5, 4 \rightarrow 6, 7 \rightarrow 6, 8 \rightarrow 3$ , 试确定符合约束的最优

表 1 车场、各货栈的坐标  $(X_i, Y_i)$ 、各货栈的任务量  $G_i$ 、卸货速度  $V_i$  及货栈的时间窗

车场	0	1	2	3	4	5	6	7	8
坐标	42, 34	11, 38	68, 77	33, 58	23, 8	62, 61	4, 12	8, 9	17, 56
任务量/t	0	55	65	0	52	0	0	56	90
卸货速度/( $\text{t} \cdot \text{h}^{-1}$ )	-	45	50	50	62	53	61	60	62
时间/h	0, 100	3, 10	0, 11	12, 25	10, 20	10, 20	5, 30	6, 23	5, 15

行车路线.

经过仿真可知完成这 5 个任务至少需要 3 辆车, 车辆 1, 2, 3 的行车路线分别为:  $0 \rightarrow 1 \rightarrow 8 \rightarrow 3 \rightarrow 0$ ,  $0 \rightarrow 7 \rightarrow 6 \rightarrow 4 \rightarrow 6 \rightarrow 0$ ,  $0 \rightarrow 2 \rightarrow 5 \rightarrow 0$ .

从图 1 可看出该网络经过多次迭代达到稳定, 可肯定图 1 中能量函数值的最低点是全局满意解.

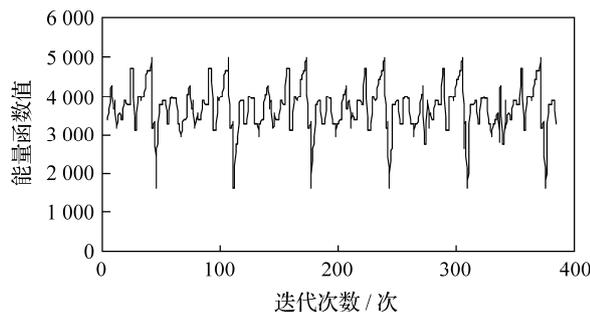


图 1 能量函数收敛图

## 5 结语

对现代企业特别是企业集群资源调度中的重要环节——车辆调度问题进行了建模, 建模的基本

思想是不仅要里程最短, 而且要在恰当的时间到达指定地点. 最后用改进的模拟退火算法对该模型进行求解, 从仿真实验可以看出该算法具有较快的收敛速度, 在短时间内能获得较高质量的全局最优解. 企业集群通过该方法进行车辆调度可以快速有效地进行任务派送, 从而降低企业的运营成本, 提高竞争力, 因此本算法具有一定的实际意义.

### 参考文献:

- [1] K C Tan, Y H Chew, L H Lee. A hybrid multi-objective evolutionary algorithm for solving vehicle routing problem with time windows[J]. Computational Optimization and Applications, 2006, 34(1):115-151.
- [2] 姜山佐, 史忠科. 有限车辆调度问题的模型和改进遗传算法[J]. 计算机应用研究, 2006(4):60-62.
- [3] Li Haibing, Lim Andrew. Evolutionary computing and optimization: local search with annealing-like restarts to solve the vehicle routing problem with time windows[C]. Proceedings of the 2002 ACM Symposium on Applied Computing, 2002.

## Study on Vehicle Scheduling Problems of Non-full Loads for Small and Medium Enterprises Clusters

XIA Wen-ming<sup>1</sup>, LI Guo-fu<sup>2</sup>, ZHU Shuang-dong<sup>1</sup>, YE Fei-fan<sup>2</sup>

( 1.Faculty of Information Science and Technology, Ningbo University, Ningbo 315211, China;

2.Faculty of Engineering, Ningbo University, Ningbo 315211, China )

**Abstract:** The algorithm presented in this paper establishes the model of minimal vehicle dispatch based on characteristics of the small and medium enterprises clusters. The model takes into consideration the minimum number of vehicle used, shortest distance, distance balance of each routing, time window constraints, etc. The analytical solution to the problem is given in the simulated annealing algorithm. The simulation results demonstrate the algorithm's feasibility and satisfaction to real-time requirements.

**Key words:** vehicle scheduling problem; non-full loads; time windows; number of vehicle used

**CLC number:** TP183

**Document code:** A

( 责任编辑 史小丽 )