

# 多车次同时集散货物路线问题研究<sup>①</sup>

李 建<sup>1,2</sup>, 达庆利<sup>2</sup>, 何瑞银<sup>1</sup>

(1 南京农业大学工学院, 南京 210031; 2 东南大学经济管理学院, 南京 210096)

**摘要:** 研究了配送车辆数和车辆工作时间内有限的多车次同时集散货物路线问题, 以车辆数和运输里程最小为目标建立了多目标模型, 提出了允许不可行解的禁忌搜索法. 该方法能同时解决路线安排和路线分配的问题, 其中路线安排采用了 4 种邻域搜索方法和重起策略, 路线分配采用了初次分配和二次调整策略. 算例分析表明, 该方法能有效地找到满意解, 且采用多车次安排路线比单车次安排路线更加经济合理.

**关键词:** 车辆路线问题; 同时集散; 多车次; 禁忌搜索算法

**中图分类号:** F505 **文献标识码:** A **文章编号:** 1007-9807(2010)10-0001-07

## 0 引言

车辆路线问题自文献<sup>[1]</sup>于 1959 年提出以来, 产生了大量的研究成果, 其中主要是单向车辆路线问题, 如文献 [2-4] 分别研究了带货物权重、多车型、开放式单向车辆路线问题. 近年由于零售业和逆向物流的发展, 集散货物路线问题 (vehicle routing problem with simultaneous deliveries and pickups VRPSDP) 引起了人们越来越多的关注. 该问题广泛应用于快捷物流、邮政及连锁超市等的配送活动中.

目前对多车次同时集散货物路线问题的研究几乎没有. 已有的相关研究主要集中在单车次的同时集散货物路线和多车次单向车辆路线问题的研究上.

文献 [5-10] 研究了单车次的同时集散货物的路线问题. 文献 [5] 设计了客户直接排列的解的表示方法, 用模拟退火算法求解了规模为 20 点的同时集散货物路线问题; 文献 [6] 设计了结合 2-opt 法和等级替换策略混合遗传算法, 以运输距离最小为目标求解了问题规模为 40 点的同时集散货物路线问题; 文献 [7] 修改大量的车辆路

线邻域搜索规则, 如 2-opt, 3-opt 并发展了 reverse, neck 等邻域搜索规则, 最后按一定顺序将这些规则组织起来, 形成启发式算法; 文献 [8] 也采用了基于插入的启发式算法, 其特点是考虑了运输距离、车辆容量剩余和离站点距离等三项插入客户的指标. 文献 [9] 用粒子群算法研究了同时集散货物路线问题; 文献 [10] 首先根据运输距离和装载容量设计了插入算法, 取得初始解, 然后用记录更新法 (record-to-record travel) 对初始解进行了改进. 已上文献对同时集散货物路线问题的算法设计作了研究, 但研究的问题还存在以下三个方面的不足: (1) 假设在计划期内 (一般为一天), 车辆只运行一车次; (2) 车辆的工作时间没有限定, 也没有考虑装卸时间; (3) 仅以旅行距离最短为目标. 这些假设使问题与实际存在明显的差距, 实际中集散货物的车辆一天一般要出行多车次; 由于司机的休息和车辆的检修, 车辆每天工作的时间是有限制的; 为了减少固定投入和司机人数, 应该尽量减少车辆数.

文献 [11-14] 虽然研究了多车次问题, 但仅为单向车辆路线问题, 没有整合逆向需求, 这样会

① 收稿日期: 2008-04-01 修订日期: 2009-10-21

基金项目: 国家自然科学基金资助项目 (70772059, 71001053); 江苏省农机局基金资助项目 (GXZ08002); 徐州市科技计划资助项目 (OPB028).

作者简介: 李 建 (1979-), 男, 四川仁寿人, 讲师, 博士生. Email: teloranc@126.com

导致车辆空载率高; 算法主要以多阶段求解为特征.

本文针对以上不足, 研究了以车辆数和旅行距离最小为目标的多车次同时集散货物问题. 由于单车次同时集散货物路线问题是 NP-hard 难题<sup>[8]</sup>, 而多车次同时集散货物问题是单车次同时集散货物路线问题的拓展, 一方面要求解单车次同时集散货物路线问题, 另一方面还要将路线分配给车辆, 这两者相互影响, 所以难度更大. 为了能同时解决路线安排和路线分配问题, 本文提出了用允许不可行解的禁忌搜索算法求解该问题, 并分析了车辆数的下界值.

### 1 模型的建立

#### 1.1 问题的描述

配送中心派遣一组车辆为客户服务, 客户可以同时收货和发货, 也可只发货或只取货, 每个客户必须被访问且只访问一次; 车辆从配送中心出发, 完成任务后返回配送中心; 货物的装载量不能超过车辆容量; 在每个计划期内, 每辆车可以服务多车次, 但工作的总时间不超过最大值  $T$ . 其中两点间的旅行时间等于两点间的距离除以平均速度  $v$ ; 客户点的装卸时间为  $s_{t_i}$ , 配送中心装货时间(或卸货时间)计为  $s_{t_0}$ . 目标是要找出车辆数和运输里程最少的车辆路线安排及路线分配. 由于车辆数直接影响到物流企业固定成本的投入和司机工资的开支, 这两项占到运输成本较大比重, 所以本文以车辆数最小为第一目标, 运输里程最小为第二目标.

#### 1.2 模型

模型符号

$V$ : 所有客户的集合;

$V_0$ : 所有节点的集合, 包括客户和配送中心;

$K$ : 车辆集合;

$R$ : 路线集合;

$Q$ : 车辆容量;

$c_r$ : 是路线  $r$  的旅行距离;

$pd_{ij}$ : 路线  $r$  中弧段  $ij$  的装载量;

$t_r$ : 路线  $r$  的旅行时间;

$T$  是车辆每天的最大工作时间;

$m$ : 是车辆数量,  $m = |K|$ ;

$y_{ir}$ : 是二进制变量, 表示路线  $r$  是否访问客户  $i$  若访问, 其值为 1, 否则为 0;

$x_r^k$ : 是二进制变量, 表示路线  $r$  是否分配给车辆  $k$  若分配, 其值为 1, 否则为 0

模型的建立

$$\min m \tag{1}$$

$$\min \sum_{k \in K} \sum_{r \in R} c_r x_r^k \tag{2}$$

$$s.t. \sum_{k \in K} \sum_{r \in R} y_{ir} x_r^k = 1, \forall i \in V \tag{3}$$

$$pd_{ij} \leq Q, \forall i \in V_0, \forall j \in V_0, i \neq j, \forall r \in R \tag{4}$$

$$\sum_{k \in K} (t_r + \sum_{i \in V} s_{t_i} y_{ir} + 2s_{t_0}) x_r^k \leq T, \forall k \in K \tag{5}$$

$$x_r^k \in \{0, 1\}, \forall k \in K, \forall r \in R \tag{6}$$

$$y_{ir} \in \{0, 1\}, i \in V, \forall r \in R \tag{7}$$

其中式 (1) 和式 (2) 是目标函数, 式 (1) 表示使车辆数最小; 式 (2) 表示旅行距离最小; 式 (3) 表示每个客户必须且仅被一条线路访问; 式 (4) 表示满足车辆容量约束; 式 (5) 是对车辆工作时间的约束; 式 (6) 和式 (7) 分别限制变量  $x_r^k$  和  $y_{ir}$  的取值.

### 2 算法设计

由于禁忌搜索算法在经典车辆路线 (vehicle routing problem, VRP) 研究中表现出较强的竞争力<sup>[15]</sup>, 所以本文选用禁忌搜索算法作为求解框架. 为了使目标 (1) 和目标 (2) 最小, 首先确定最小车辆数: 仅考虑容量约束, 计算出旅行距离最小的路线安排及其总运输时间, 进而由车辆最大工作时间, 确定出最小车辆数; 然后将容量约束和车辆工作时间约束一起考虑, 再确定旅行距离最小的路线安排.

由于多车次同时集散货物路线问题包含路线的安排和路线的分配两个子问题, 而这两个子问题相互作用, 无论先考虑哪一个子问题, 都难以确保解的可行性. 为了同时决策这两个子问题, 将车辆工作时间约束 (5) 修改后加入旅行距离目标函数 (2) 中, 允许产生不可行解. 目标函数 (2) 修改后的结果如式 (8) 所示, 其中  $M$  为惩罚值.

$$\min \sum_{k \in K} \sum_{r \in R} c_r x_r^k + M \sum_{k \in K} \max (T - \sum_{r \in R} (t_r + \sum_{i \in V} s_{-t_i} y_{ir} + 2s_{-t_0}) x_r^k, 0) \quad (8)$$

目标函数 (8) 需要计算总的旅行距离和总的时间违背量, 总的旅行距离通过安排的路线直接求出, 而总的时间违背量要通过路线分配的结果来计算. 其总体算法步骤为:

步骤 1 仅考虑容量约束构造初始解;

步骤 2 仅考虑容量约束, 以旅行距离最小为目标, 即目标函数 (2), 从当前解执行禁忌搜索算法, 重复 *outer\_iter\_max* 次步骤 2.1 和 2.2

步骤 2.1 对当前解依序使用 2-opt\*, Reverse, 1-0, Reverse, 1-1, Reverse 邻域搜索方法, 选出最好的非禁忌解作为当前解, 并更新目前最好解.

步骤 2.2 如果目前最好解超过一定代数 (*Inner\_iter\_max*) 没有更新, 则重起, 清空禁忌表; 从 2-opt\*, Reverse, 1-0 和 1-1 中随机选取两邻域搜索方法执行, 以改变当前搜索方向.

步骤 3 根据仅考虑容量约束, 以旅行距离最小为目标求得的路线安排, 确定最小车辆数;

步骤 4 以旅行距离和总的时间违背量之和最小为目标, 即目标函数 (8), 重复 *outer\_iter\_max* 次步骤 2.1 和步骤 2.2 其中, 在步骤 2.1 至步骤 2.2 中每执行一步后, 对目前最好解进行更新, 更新的规则是: 若时间违背量  $\sum_{k \in K} \max (T - \sum_{r \in R} (t_r + \sum_{i \in V} s_{-t_i} y_{ir} + 2s_{-t_0}) x_r^k, 0)$  减小, 则更新; 或者时间违背量相等, 而运输里程  $\sum_{k \in K} \sum_{r \in R} c_r x_r^k$  减小, 则更新. 目前最好解是指从程序开始到当前搜索到的最好解.

### 2.1 初始解的构造

用扫描法<sup>[16]</sup>构造初始路线, 其步骤如下:

步骤 1 根据客户极坐标值, 对客户进行排序;

步骤 2 随机选取一客户作为起点, 依次将客户分配给路线, 路线只需满足容量约束 (4), 如图 1 所示;

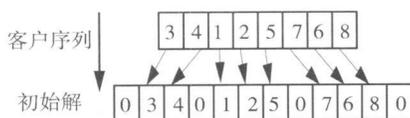


图 1 初始解的产生

Fig. 1 Generating the initial solution

步骤 3 计算旅行距离, 即目标函数 (2);

步骤 4 由于每个客户都可作为车辆路线的第一个客户, 这样有个客户就可形成种路线安排, 选择其中最好的路线安排 (即目标值最小) 作为禁忌搜索算法的初始解.

### 2.2 路线邻域搜索方法

#### 2.2.1 邻域结构设计

本文选用 4 种邻域搜索方法: Reverse, 1-0 交换, 1-1 交换, 2-opt\* 交换. 1-0 交换, 1-1 交换, 2-opt\* 交换属于线路间优化, 而 Reverse 属于线路内部优化. 采用客户和配送中心共同排列表示解的方法, 用 0 表示配送中心, 用 1, 2, ..., n 表示各需求点. 为了利于程序的编写和应用, 本文采用编码的方式来解释这些方法.

(1) Reverse 该方法出自文献 [7], 就是将线路的一部分子线路逆转, 属于线路内部变换. 如将某条路线 0 1 2345670 中的划线部分逆转后变为 01 6543270

(2) 1-0 交换, 1-1 交换. 它们也称为 Swap/shift 移动, 介绍在文献 [10] 中, 是路线间的客户移动. 1-0 交换就是一条路线的某一客户插入到另一路线中去, 如路线 012345067890 中的客户 2 插入到客户 8 前面, 得到路线 013450672890. 1-1 交换是指在两条路线中各选一个客户来相互交换, 如路线 012345067890 中的客户 2 与客户 8 交换, 得到路线 018345067290.

(3) 2-opt\* 交换. 该方法介绍于文献 [17], 它是 1-1 交换的扩展, 将两条路线的尾部进行交换, 如路线 012345067890 中的划线部分交换, 得到路线 012 890673450

在以上四种邻域方法中, Reverse 邻域方法的计算时间复杂性为  $O(n^2)$ , 1-0 交换, 1-1 交换和 2-opt\* 交换的时间复杂性均为  $O(n^3)$ .

#### 2.2.2 重起策略和扰动

重起就是选择一个新解作为搜索的当前解, 本文选择目前最好解作为重起的起点. 当目前最好解超过一定代数没有被更新, 则将该最好解作为当前解, 清空记忆表, 沿新的方向搜索解空间.

为了改变解的搜索方向, 加入扰动, 以使解沿着新的方向搜索, 开发更大的解空间. 本文采用的扰动方法是在以上四种邻域方法 (Reverse, 1-0 交换, 1-1 交换和 2-opt\*) 中随机选取两种方法

来执行,以改变固定搜索顺序.

### 2 3 车辆数的确定

通过取消车辆工作时间约束,先求出总的运输时间(包括旅行时间和装卸时间),再确定最小车辆数.具体方法为:首先将时间约束  $T$  设置为一大数  $M$ ,把问题变为只有容量约束的同时集散货物路线问题.通过 3 1和 3 2部分求解该问题,得出总运输时间  $\sum_{r \in R} (t_r + \sum_{i \in V} s_{-i} y_{ir} + 2s_{-t_0})$ . 由于总运输时间须满足  $\sum_{r=1}^{IR1} (t_r + \sum_{i \in V} s_{-i} y_{ir} + 2s_{-t_0}) \leq \sum_{k=1}^m T$ , 其中使该不等式满足的最小  $m$  值就是所求的最小车辆数,即  $m = \lfloor \sum_{r \in R} (t_r + \sum_{i \in V} s_{-i} y_{ir} + 2s_{-t_0}) / T \rfloor + 1$  “ $\lfloor \rfloor$ ”表示向下取整.由于车辆最大工作时间的约束,若在最小值  $m$  下找不到可行解,可将增加 1 重新搜索.

2 4 路线的分配和调整

#### (1) 路线的初次分配

路线初次分配的方法是将运输时间最长的路线分配给剩余时间最多的车辆,依次循环直到路线分配完为止,其中允许不满足时间约束.

#### (2) 路线的调整

若初次分配的结果不能满足时间约束,则对其路线进行调整,以减少时间违背量.对路线调整的策略是两两交换,即在两辆不同车中随机选取了两条路线进行交换,以取得时间违背量小的解.

### 3 数据实验

某第三方物流企业需要为 100 个客户集送货物,其中第三方物流企业的配送中心坐标为 (34 36), 客户的位置、货物需求量和发送量见表 1 所示,在每个客户点的装卸货时间设为 0.1 h 车辆在配送中心装货和卸货时间各为 0.1 h 车速设为 35 km/h 车辆装载容量为 10 t 每辆车的工作时间为 8 h 惩罚值  $M$  设为 3 000  $outer\_iter = 100$   $inner\_iter\_max = 20$  笔者利用 Visual Basic 6.0 编写了算法程序,运行计算机 CPU 为 Pentium IV 1.6GHz PC

### 3 1 多车次同时集散货物路线问题的计算

为了确定最小车辆数,首先将  $T$  设置为  $M$ ,即取消车辆工作时间约束,求解单车次同时集散货物路线问题.单车次同时集散货物路线问题的最好可行解目标值为 919 181km (运算 10 次), CPU 计算时间为 36 s 解为:

- 车辆 1 0- 70- 71- 26- 96- 48- 4- 24- 3- 25- 67- 12- 68- 0
  - 车辆 2 0- 99- 28- 93- 18- 5- 27- 94- 15- 95- 53- 59- 16- 0
  - 车辆 3 0- 62- 98- 17- 8- 29- 21- 72- 20- 41- 37- 0
  - 车辆 4 0- 79- 11- 19- 81- 45- 75- 76- 40- 80- 0
  - 车辆 5 0- 92- 6- 7- 46- 9- 74- 73- 100- 42- 30- 61- 91- 43- 10- 0
  - 车辆 6 0- 31- 82- 66- 49- 77- 35- 65- 34- 39- 44- 88- 89- 13- 87- 86- 0
  - 车辆 7 0- 63- 36- 14- 64- 90- 78- 22- 38- 0
  - 车辆 8 0- 54- 55- 57- 97- 23- 0
  - 车辆 9 0- 50- 83- 84- 85- 33- 32- 1- 2- 60- 51- 58- 47- 52- 69- 56- 0
- 解的相关数据如表 2 所示.

由于总发货量和总收货量的最大值为 83 79 吨,所以表 2 表明单车次解已取得最小车辆数 (9 辆). 利用  $m = \lfloor \sum_{r \in R} (t_r + \sum_{i \in V} s_{-i} y_{ir} + 2s_{-t_0}) / T \rfloor + 1$  计算多车次最小车辆数  $m = \lfloor 38.06/8 \rfloor + 1 = 5$  辆,求解多车次同时集散货物路线问题.最好可行解目标值为 940 662.1 km (运算 10 次), CPU 计算时间为 45 s 解为:

- 车辆 1 0- 23- 55- 54- 0- 93- 5- 18- 6- 46- 9- 74- 21- 73- 100- 42- 61- 19- 91- 43- 0
- 车辆 2 0- 66- 33- 77- 49- 35- 65- 34- 39- 89- 13- 87- 86- 10- 0- 37- 79- 41- 20- 72- 98- 0
- 车辆 3 0- 38- 22- 78- 90- 64- 14- 36- 63- 0- 11- 80- 40- 30- 76- 75- 81- 45- 44- 88- 0
- 车辆 4 0- 68- 12- 67- 25- 3- 24- 48-

4- 51- 60- 47- 58- 0- 62- 17- 29- 7- 8- 71- 95- 53- 59- 16- 0- 50- 31- 83- 82-  
 92- 28- 99- 0 84- 85- 32- 1- 2- 97- 57- 52- 69- 56- 0  
 车辆 5 0- 70- 15- 94- 27- 96- 26- 解的相关数据如表 3 所示.

表 1 客户数据

Table 1 Data of the clients

序号	X 坐标 (km)	Y 坐标 (km)	发货量 (t)	收货量 (t)	序号	X 坐标 (km)	Y 坐标 (km)	发货量 (t)	收货量 (t)	序号	X 坐标 (km)	Y 坐标 (km)	发货量 (t)	收货量 (t)	序号	X 坐标 (km)	Y 坐标 (km)	发货量 (t)	收货量 (t)
1	42	7	0.08	0.42	26	10	20	0.95	0.95	51	15	19	0.08	0.02	76	49	73	1.68	0.82
2	24	12	0.25	0.25	27	5	30	0.03	0.17	52	22	22	0.20	0.00	77	67	5	0.19	2.31
3	23	3	0.09	0.61	28	20	40	0.60	0.60	53	18	24	1.65	0.55	78	56	39	2.51	1.09
4	11	14	1.41	0.39	29	15	60	0.43	1.28	54	26	27	2.60	0.10	79	37	47	0.47	0.13
5	6	38	0.25	1.35	30	45	65	0.62	0.28	55	25	24	1.92	0.08	80	37	56	0.33	0.17
6	2	48	0.00	0.10	31	45	20	0.49	0.61	56	22	27	0.90	0.20	81	57	68	1.26	0.24
7	8	56	0.39	2.31	32	45	10	0.40	1.40	57	25	21	1.01	0.19	82	47	16	0.85	1.65
8	13	52	0.90	2.70	33	55	5	0.26	2.64	58	19	21	0.90	0.10	83	44	17	0.35	0.55
9	6	68	0.26	2.74	34	65	35	0.16	0.14	59	20	26	0.69	0.21	84	46	13	0.23	0.57
10	47	47	1.30	0.00	35	65	20	0.18	0.42	60	18	18	1.70	0.00	85	49	11	0.40	1.40
11	41	49	0.84	0.16	36	45	30	1.13	0.57	61	49	58	0.84	0.16	86	49	42	1.11	0.19
12	35	17	0.34	0.36	37	35	40	1.40	0.20	62	27	43	0.57	0.33	87	53	43	1.14	0.26
13	55	45	1.06	0.24	38	41	37	1.44	0.16	63	37	31	1.17	0.23	88	61	52	0.26	0.04
14	55	20	0.69	1.21	39	64	42	0.59	0.31	64	57	29	0.92	0.88	89	57	48	1.94	0.36
15	15	30	1.30	1.30	40	40	60	1.40	0.70	65	63	23	0.07	0.13	90	56	37	0.40	0.20
16	25	30	0.25	0.05	41	31	52	1.61	1.09	66	53	12	0.14	0.46	91	55	54	2.55	0.05
17	20	50	0.20	0.30	42	35	69	1.17	1.13	67	32	12	0.26	0.44	92	15	47	0.51	1.09
18	10	43	0.21	0.69	43	53	52	1.08	0.02	68	36	26	1.30	0.50	93	14	37	0.42	0.68
19	55	60	1.47	0.13	44	65	55	1.18	0.22	69	21	24	2.45	0.35	94	11	31	0.25	0.45
20	30	60	0.80	0.80	45	63	65	0.78	0.02	70	17	34	0.15	0.15	95	16	22	2.98	1.12
21	20	65	0.37	0.83	46	2	60	0.02	0.48	71	12	24	0.65	0.65	96	4	18	0.78	2.72
22	50	35	1.33	0.57	47	20	20	0.80	0.00	72	24	58	0.79	1.11	97	28	18	1.67	0.93
23	30	25	1.92	0.38	48	5	5	1.60	0.00	73	27	69	0.39	0.61	98	26	52	0.45	0.45
24	15	10	1.33	0.67	49	60	12	0.62	2.48	74	15	77	0.18	0.72	99	26	35	1.11	0.39
25	30	5	0.13	0.67	50	40	25	0.56	0.34	75	62	77	1.61	0.39	100	31	67	0.14	0.16

表 2 单车次同时集散货物路线问题解的相关数据

Table 2 Related data of the solution to one-trip VRPSDP

数据	车辆 1	车辆 2	车辆 3	车辆 4	车辆 5	车辆 6	车辆 7	车辆 8	车辆 9	平均值
	路线 1	路线 2	路线 3	路线 4	路线 5	路线 6	路线 7	路线 8	路线 9	
最大装载量	9.45	9.75	9.33	9.83	9.45	9.98	9.59	9.12	9.57	9.56
满载率	94.5%	97.5%	93.3%	98.3%	94.5%	99.8%	95.9%	91.2%	95.7%	95.6%
路线时间	4.80	3.84	3.69	4.39	5.91	5.60	3.02	1.88	4.93	4.23
车辆使用时间利用率	60.0%	48.0%	46.1%	54.9%	73.9%	70.0%	37.8%	23.5%	61.6%	52.9%

表 3 多车次同时集散货物路线问题解的相关数据

Table 3 Related data of the solution to multi-trip VRPSDP

数据	车辆 1		车辆 2		车辆 3		车辆 4		车辆 5		平均值
	路线 1	路线 2	路线 3	路线 4	路线 5	路线 6	路线 7	路线 8	路线 9	路线 10	
最大装载量	6.44	9.58	9.61	5.85	9.59	9.95	9.96	9.23	9.68	9.84	8.97
满载率	64.4%	95.8%	96.1%	58.5%	95.9%	99.5%	99.6%	92.3%	96.8%	98.4%	89.7%
路线时间	1.4	6.4	5.3	2.5	3.0	4.7	4.5	3.2	3.6	4.3	3.9
车辆使用时间	7.84		7.79		7.68		7.62		7.95		7.78
车辆使用时间利用率	98.0%		97.3%		96.0%		95.2%		99.4%		97.2%

### 3.2 允许不可行解的禁忌搜索算法与两阶段算法的比较

从单车次同时集散货物路线问题和多车次同时集散货物路线问题的计算可知,该算法均能搜索到的满意解:车辆数均取得下界值,单车次同时集散货物路线问题的解具有较高的平均满载率(95.6%),多车次同时集散货物路线问题的解具有较高的平均时间利用率(97.2%)。为了更好地体现允许不可行解的禁忌搜索算法的优势,将其与两阶段算法作比较,如表 4 所示。两阶段算法就是

先不考虑车辆工作时间的限制,用禁忌搜索算法计算单车次值;然后在第二阶段根据单车次解用装箱算法生成多车次路线安排。从表 4 可以看出允许不可行解的禁忌搜索算法求得的车辆数比两阶段算法取得的车辆数节约了 2 辆(28.6%),而运输距离仅增加了 21.48 km(2.3%);车辆使用平均时间利用率提高了 29.2%,而车次平均满载率仅降低 5.9%。由于本文以车辆数为第一目标、运输距离为第二目标,所以表 4 表明允许不可行解的禁忌搜索算法优于两阶段算法。

表 4 允许不可行解的禁忌搜索算法与两阶段算法的比较

Table 4 Comparison of tabu search algorithm allowing infeasible solutions with two-phase algorithm

比较	车辆数	运行距离 (km)	车次平均满载率	车辆使用平均时间利用率
两阶段算法	7	919.18	95.6%	68.0%
允许不可行解的禁忌搜索算法	5	940.66	89.7%	97.2%

### 3.3 多车次同时集散货物路线问题与单车次同时集散货物路线问题的比较

为了体现多车次安排的优势,将多车次安排和单车次安排的主要参数作比较,如表 5 所示。从表 5 的比较可以看出采用多车次安排能够节约 4 辆车辆的使用,相应减少 44.4% 的车辆固定使用费(或租赁费)和司机工资费用,而运行距离只增

加了 21.48 公里,占总旅行距离的 2.3%;车次平均满载率虽然降低 5.9%,但车辆使用平均时间利用率提高了 44.3%。由于车辆固定成本和司机工资费用在运输总费用中占较大比重,所以采用多车次安排,能在车辆满载率和车辆使用时间利用率之间取得更好的平衡,使配送操作更加经济合理。

表 5 多车次同时集散货物路线问题与单车次同时集散货物路线问题的比较

Table 5 Comparison of multi-trip VRPSDP with one-trip VRPSDP

比较	车辆数	运行距离 (km)	车次平均满载率	车辆使用平均时间利用率
单车次同时集散货物路线问题	9	919.18	95.6%	52.9%
多车次同时集散货物路线问题	5	940.66	89.7%	97.2%

## 4 结束语

多车次同时集散货物路线问题的研究对配送操作具有十分重要的现实意义. 本文针对现实中配送车辆和工作时间有限的特点, 建立了更符合实际的多车次同时集散货物路线问题模型. 由于模型属于 NP-hard 难题, 本文提出根据

单车次同时集散货物路线问题的解确定最小使用车辆数, 再采用允许不可行解的禁忌搜索法同时决策路线安排和路线分配的问题. 最后算例的计算表明该方法能有效的求得满意的可行解; 采用多车次路线安排更加经济合理. 本文的模型和求解算法可为广大物流企业 (尤其是第三方物流企业) 的实际配送操作提供优化技术支持.

## 参考文献:

- [1] Dantzig G, Ramser J. The truck dispatching problem[J]. *Management Science*, 1959, 6(1): 80-91
- [2] 潘震东, 唐加福, 韩毅. 带货物权重的车辆路径问题及遗传算法[J]. *管理科学学报*, 2007, 10(3): 23-29  
Pan Zhen-dong, Tang Jia-fu, Han Yi. Vehicle routing problem with weight coefficients[J]. *Journal of Management Sciences in China*, 2007, 10(3): 23-29. (in Chinese)
- [3] 李建, 张永, 达庆利. 第三方物流多车型硬时间窗路线问题研究[J]. *系统工程学报*, 2008, 23(1): 74-80  
Li Jian, Zhang Yong, Da Qing-li. Research on heterogeneous vehicle routing problem with hard time windows for the third party logistics[J]. *Journal of Systems Engineering*, 2008, 23(1): 74-80. (in Chinese)
- [4] Li F, Golden B, Wasil E. The open vehicle routing problem: Algorithms, large scale test problems and computational results[J]. *Computers & Operations Research*, 2007, 34(10): 2918-2930
- [5] 郎茂祥. 装卸混合车辆路径问题的模拟退火算法研究[J]. *系统工程学报*, 2005, 20(5): 485-491.  
Lang Mao-xiang. Study on simulated annealing algorithm for vehicle routing problem with backhauls[J]. *Journal of Systems Engineering*, 2005, 20(5): 485-491. (in Chinese)
- [6] 张建勇, 李军. 具有同时配送和回收需求的车辆路径问题的混合遗传算法[J]. *中国公路学报*, 2006, 19(4): 118-122  
Zhang Jian-yong, Li Jun. Hybrid genetic algorithm to vehicle routing problem with simultaneous delivery and pick-up[J]. *China Journal of Highway and Transport*, 2006, 19(4): 118-122. (in Chinese)
- [7] Nagy G, Salhi S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries[J]. *European Journal of Operational Research*, 2005, 162(1): 126-141.
- [8] Dehboff J. Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up[J]. *OR Spectrum*, 2001, 23(1): 79-96
- [9] Aït J, Kachitichyanukul V. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery[J]. *Computers & Operations Research*, 2009, 36(5): 1693-1702
- [10] Chen J-F, Wu T-H. Vehicle routing problem with simultaneous deliveries and pickups[J]. *Journal of the Operational Research Society*, 2006, 57(5): 579-587.
- [11] Taillard E, Laporte G, Gendreau M. Vehicle routing with multiple use of vehicles[J]. *Journal Operational Research Society*, 1996, 47(8): 1065-1070
- [12] Petch R J, Salhi S. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips[J]. *Discrete Applied Mathematics*, 2004, 133(1): 69-92
- [13] Brandao J, Mercer A. A tabu search algorithm for the multi trip vehicle routing and scheduling problem[J]. *European Journal of Operational Research*, 1997, 100(1): 180-191.
- [14] Oliveira A, Viera O. Adaptive memory programming for the vehicle routing problem with multiple trips[J]. *Computers & Operations Research*, 2007, 34(1): 28-47.
- [15] Toth P, Vigo D. *The Vehicle Routing Problem*[M]. Philadelphia: Siam, 2002
- [16] Gillett B E, Miller L R. A heuristic algorithm for the vehicle dispatch problem[J]. *Operations Research*, 1974, 22(4): 340-349.

(下转第 62 页)

## Pricing credit risk with multivariate stochastic volatility model

WU Hua<sup>1</sup>, YI Chuan-he<sup>1</sup>, DU Jun<sup>2</sup>, HE Zheng-chu<sup>2</sup>

1 School of Business, South Central University, Changsha 410083 China;

2 School of Economics and Management, Changsha University of Science and Technology, Changsha 410114 China

**Abstract** This paper extends the closed-form solution for options with stochastic volatility to the multi-asset framework. And introduces a risk premium into the return equation and considers Wishart dynamics for the process of the stochastic volatility matrix which is the multi-asset analogue of the model of Cox and Ross. Moreover, it extends Merton's model for corporate default to a framework with stochastic liability. At last, we analyze the applicability and stability of the solutions by numerical simulation. The results show that the multivariate stochastic volatility model could explain the credit of corporations under condition of stochastic default risk more distinctly. Multi-assets model helps us to deeply capture the dynamic relationship between value of corporation's assets and the bank, which is much important to the risk management of financial institution.

**Key words** stochastic volatility; derivative pricing; wishart process; credit risk

(上接第 7 页)

[17] Potvin JY, Rousseau JM. An exchange heuristic for routing problems with time windows [J]. Journal of the Operational Research Society, 1995, 46(12): 1433-1446

## Simultaneous deliveries and pickups

LI Jian<sup>1,2</sup>, DA Qing-li<sup>2</sup>, HE Rui-yin<sup>1</sup>

1 College of Engineering, Nanjing Agricultural University, Nanjing 210031, China;

2 College of Economics and Management, Southeast University, Nanjing 210096, China

**Abstract** Multi-trip vehicle routing problem with simultaneous deliveries and pickups is studied, in which the number and working time of the vehicles are limited. Based on the minimum number and travel distance of vehicles, a multiobjective model is formulated. The tabu search algorithm allowing infeasible solutions is proposed, which can simultaneously solve two problems, that is, planning and assigning routes for vehicles. For planning routes, four neighborhood search methods and restart strategy are employed; initial routes and improved routes assignment are adopted for assigning routes. Computational results show that this algorithm is effective to search the satisfactory solutions and the multi-trip arrangement is more economic and reasonable than one-trip arrangement.

**Keywords** vehicle routing problem; simultaneous deliveries and pickups; multi-trip; tabu search algorithm