

# 多车场多配送中心的物流配送问题研究

李敏, 郭强, 刘红丽

LI Min, GUO Qiang, LIU Hong-li

西北工业大学 理学院应用数学系, 西安 710072

Department of Applied Mathematics in School of Science Northwestern Polytechnical University, Xi'an 710072, China

LI Min, GUO Qiang, LIU Hong-li. Multiple depot multi-logistics center distribution problem. Computer Engineering and Applications, 2007, 43(8): 202-204.

**Abstract:** This paper discusses the multiple depot multi-logistics center problem on the general network. Using Floyd algorithm, it gives an algorithm of the minimum cost on condition that carrying capacity of the vehicles limited. This algorithm has a function of labeling the path, it can point out the sequence of the vehicles from the depot to distributions and users, at last return to the original depot.

**Key words:** distribution; shortest path; multiple depot; multi-logistics center; Floyd algorithm

**摘要:** 针对一般网络上的多车场多配送中心的物流配送问题, 借鉴 Floyd 算法, 构建出了一种在有车辆载重量限制的条件下, 使完成所有配送任务总费用最少的算法。该算法具有路径标记功能, 可以指出车辆从车场出发, 经过配送中心以及用户点, 最后返回原车场的次序与路径走向。

**关键词:** 配送; 最短路径; 多车场; 多配送中心; Floyd 算法

文章编号: 1002-8331(2007)08-0202-03 文献标识码: A 中图分类号: TP39

## 1 引言

多车场的物流配送问题是基本车辆路线问题的扩展, 它一般研究的是有多个车场同时对若干个用户进行服务, 每个用户有一定的货物需求, 要求对各车场的车辆和行驶路线进行适当的安排, 在保证满足各用户需求的前提下, 使总配送费用最低的问题。另外, 随着现代商业的发展, 为了提高经济效益, 有时候需要多个配送中心同时进行配送, 即利用多个配送中心为用户服务。因此, 多车场多配送中心的物流配送问题在现实生活中有着很强的应用背景。在现行研究的此类问题中, 文献[2, 3]首先将此类问题转化为单车场问题, 在此基础上用解决单车场配送问题的方法进行处理; 文献[4]探讨了一般条件下的多车场满载的配送路线问题, 提出了一种基于网络流模型最优解的启发式算法。本文根据多车场多配送中心物流配送问题的描述, 提出一种具有路径标记功能的算法, 比较容易转化为程序在计算机上实现, 具有很好的适用性。

## 2 问题的描述

将车场、配送点、用户点以及转运点都视为同一个网络上的节点, 车场、配送点、用户点以及转运点之间的单位运费都视为连接相邻两个节点之间弧上的权值, 那么一般网络上的多车场多配送中心物流配送问题的描述如下:

在  $n$  个节点的无向连通网络中, 节点  $i_\alpha (\alpha=1, 2, \dots, l)$  是配送点, 其存储的货物量为  $a_\alpha (\alpha=1, 2, \dots, l)$ ;  $j_\beta (\beta=1, 2, \dots, m)$  是用户点, 相应的需求量为  $b_\beta (\beta=1, 2, \dots, m)$ ;  $\sum_{\alpha=1}^l a_\alpha = \sum_{\beta=1}^m b_\beta$ ;  $g_v (v=$

$1, 2, \dots, p)$  是车场, 具有车辆  $Q_v (v=1, 2, \dots, p)$ , 单车装载量均为  $w$ 。网络中相邻两节点  $i$  与  $j$  之间车载货时, 单位运费  $c_{ij} = c_{ji}$  ( $i, j=1, 2, \dots, n$ ), 车空驶时, 单车费用为  $\varepsilon_{ij}$  ( $i, j=1, 2, \dots, n$ )。要求每辆车运完货物之后, 必须返回原车场。研究: 如何调度车辆, 选择运输路线, 使完成所有配送任务的总费用最少?

## 3 问题的数学模型

在  $n$  个节点的网络上, 首先做如下约定:  $x_{ij}$  表示节点  $i$  与节点  $j$  间通过的货物量,  $y_{ij}$  表示节点  $i$  与节点  $j$  间空车通过的次数, 则数学模型为:

$$\begin{aligned} \min & \sum_{i=1}^n \sum_{j=1}^n (c_{ij} x_{ij} + \varepsilon_{ij} y_{ij}) \\ & \sum_{\alpha=1}^l x_{i_\alpha, j_\beta} = b_{j_\beta} \quad (\beta=1, 2, \dots, m) \\ & \sum_{\beta=1}^m x_{i_\alpha, j_\beta} = a_{i_\alpha} \quad (\alpha=1, 2, \dots, l) \\ \text{s.t.} & \sum_{\alpha=1}^l y_{g_v, i_\alpha} \leq Q_v \quad (v=1, 2, \dots, p) \\ & \sum_{\beta=1}^m y_{j_\beta, g_v} = \sum_{\alpha=1}^l y_{g_v, i_\alpha} \quad (v=1, 2, \dots, p) \\ & \sum_{v=1}^p y_{g_v, i_\alpha} \leq \left\lceil \frac{a_{i_\alpha}}{w} \right\rceil + 1 \quad (\alpha=1, 2, \dots, l) \\ & x_{ij} \geq 0, y_{ij} \text{ 为非负整数 } (i, j=1, 2, \dots, n) \end{aligned}$$

目标函数中的  $\sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij}$  表示车上载有货物行驶时的总费用,  $\sum_{i=1}^n \sum_{j=1}^n \varepsilon_{ij}y_{ij}$  表示车辆空驶时的总费用;第一个约束是指第  $\beta$  个用户点接收的货物量为  $b_{j\beta}$  ( $\beta=1,2,\dots,m$ );第二个约束是指从第  $\alpha$  个配送中心发出的货物量为  $a_{i\alpha}$  ( $\alpha=1,2,\dots,m$ );第三个约束是指从第  $v$  个车场发出的车辆数不超过其具有的车辆数  $Q_v$  ( $v=1,2,\dots,p$ );第四个约束是指车辆从车场出发经过配送点  $i_\alpha$  ( $\alpha=1,2,\dots,l$ ) 和用户点  $j_\beta$  ( $\beta=1,2,\dots,m$ ),再返回到原车场;第五个约束是指发往第  $\alpha$  个配送中心的所有车辆数不超过其要配送的货物量所需车辆的总数;第六个约束是整数化约束。

#### 4 问题的解决

本算法的基本思路是首先利用 Floyd 算法找出车场、配送点以及用户点中所有两个节点之间的最小费用路径,并按各最小费用路径调运各车场的车辆到配送点,逐个将各配送点的货物依次运往各用户点,从而得到初始可行的配送方案。在实际配送过程中,将依据算法的最终确定方案对各车场车辆的调运以及各配送点到各用户点的配送同步进行。然后根据可行解和单位费用针对一般网络构造一个含有负权值的矩阵,对此矩阵用 Floyd 算法求其负值回路,并沿负值回路调优降费,改进可行解,重复上述过程,直到无负值回路为止,此时的可行解便是数学模型的最优解。具体的算法步骤如下:

- (1) 输入  $c_{ij}, \varepsilon_{ij}$  ( $i, j=1, 2, \dots, n$ ),  $a_{i\alpha}$  ( $\alpha=1, 2, \dots, l$ ),  $b_{j\beta}$  ( $\beta=1, 2, \dots, m$ ),  $Q_v$  ( $v=1, 2, \dots, p$ ),  $x_{ij}=0, y_{ij}=0, d_{ij}=c_{ij}, r_{ij}=j$  ( $i, j=1, 2, \dots, n$ ),  $k=1$  (注:当节点  $i$  与  $j$  不相邻时,  $c_{ij}=\infty$ ;当  $i=j$  时,  $c_{ij}=0$ );
- (2) 若  $d_{ik}+d_{kj}\Delta\mu < d_{ij}$ , 则  $d_{ij}=\mu, r_{ij}=r_{ik}$ ; 否则 ( $\mu \geq d_{ij}$ ),  $d_{ij}$  与  $r_{ij}$  都不变 ( $i, j=1, 2, \dots, n$ );
- (3) 若  $k < n$ , 则  $k=k+1$ , 转到 (2); 否则 ( $k=n$ ),  $\beta=1$ , 转到 (4);
- (4)  $\theta:=\min\{\alpha | a_{i\alpha} > 0, 1 \leq \alpha \leq l\}$

$$\lambda:=\min\{d_{i_\theta i_\alpha} | Q_v - \sum_{k=1}^l y_{i_\theta i_\alpha} > 0, a_{i_\alpha} > 0, 1 \leq v \leq p\}$$

$$\eta:=\min\{v | Q_v - \sum_{k=1}^l y_{i_\theta i_\alpha} > 0, d_{i_\theta i_\alpha} > \lambda, 1 \leq v \leq p\};$$

- (5) 若  $b_{j_\beta}=0$ , 转 (13); 否则 ( $b_{j_\beta} > 0$ ),  $E:=\min\{a_{i_\theta}, b_{j_\beta}, Q_{\eta^*} \cdot w\}$ ,  $a_{i_\theta}:=a_{i_\theta}-E, b_{j_\beta}:=b_{j_\beta}-E, s:=g_{\eta^*}$ , 转 (6);

- (6) 若  $E-w\left[\frac{E}{w}\right] > 0$ , 则  $R:=\left[\frac{E}{w}\right]+1, Q_{\eta^*}=Q_{\eta^*}-R$ ; 否则  $\left(E-w\left[\frac{E}{w}\right]=0\right)$ , 则  $R:=\left[\frac{E}{w}\right], Q_{\eta^*}=Q_{\eta^*}-R$ ;

- (7)  $t:=r_{i_\theta}$ ;
- (8)  $y_{st}:=y_{st}+R$ , 若  $t \neq i_\theta$ , 则  $s:=t$ , 转 (7); 否则 ( $t=i_\theta$ ),  $s:=i_\theta$ , 转 (9);
- (9)  $t:=r_{j_\beta}$ ;
- (10)  $x_{st}:=x_{st}+E$ , 若  $t \neq j_\beta$ , 则  $s:=t$ , 转 (9); 否则 ( $t=j_\beta$ ),  $s:=j_\beta$ , 转 (11);
- (11)  $t:=r_{i_\theta}$ ;

- (12)  $y_{st}:=y_{st}+R$ , 若  $t \neq g_{\eta^*}$ , 则  $s:=t$ , 转 (11); 否则 ( $t=g_{\eta^*}$ ), 转 (4);

- (13) 若  $\beta < m$ , 则  $\beta:=\beta+1$ , 转 (4); 否则 ( $\beta=m$ ), 转 (14);

- (14) 若  $x_{ij} > x_{ji}$ , 则  $x_{ij}:=x_{ij}-x_{ji}, x_{ji}:=0$ ; 否则 ( $x_{ij} \leq x_{ji}$ ),  $x_{ji}:=x_{ji}-x_{ij}, x_{ij}:=0$  ( $i, j=1, 2, \dots, n$ );

- (15) 若  $x_{ij}=x_{ji}=0, d_{ij}=d_{ji}=c_{ij}$ ; 否则 ( $\max\{x_{ij}, x_{ji}\} > 0$ ), 当  $x_{ij} > 0$  时,  $d_{ij}:=c_{ij}, d_{ji}:=c_{ij}$ ; 当  $x_{ji} > 0$  时,  $d_{ij}:=c_{ij}, d_{ji}:=c_{ij}, r_{ij}:=j$  ( $i, j=1, 2, \dots, n$ ),  $k=1$ ;

- (16) 若  $d_{ik}+d_{kj}\Delta\mu' < d_{ij}$ , 则  $d_{ij}:=\mu', r_{ij}:=r_{ik}$ ; 否则 ( $\mu' \geq d_{ij}$ ),  $d_{ij}$  与  $r_{ij}$  都不变 ( $i, j=1, 2, \dots, n$ );

- (17) 求  $\min\{d_{ik} | 1 \leq i \leq n\} \Delta d_{ik}$ , 若  $d_{ik} < 0$ , 则  $s:=u, \lambda':$  是一个充分大的数, 转到 (18); 否则 ( $d_{ik}=0$ ), 转到 (24);

- (18)  $t:=r_{ik}$ ;

- (19) 若  $x_{st} > 0$ , 则  $\lambda'=\min\{\lambda', x_{st}\}$ , 转到 (20); ( $x_{st}=0$ ), 转到 (20);

- (20) 若  $t \neq u$ , 则  $s:=t$ , 转到 (18); 否则 ( $t=u$ ),  $s:=u$ , 转到 (21);

- (21)  $t:=r_{ik}$ ;

- (22) 若  $x_{st} > 0$ , 则  $x_{st}:=x_{st}-\lambda'$ , 转到 (23); ( $x_{st}=0$ ),  $x_{st}:=x_{st}+\lambda'$ , 转到 (23);

- (23) 若  $t \neq u$ , 则  $s:=t$ , 转到 (21); 否则 ( $t=u$ ), 转到 (15);

- (24) 若  $k < n$ , 则  $k=k+1$ , 转到 (16); 否则 ( $k=n$ ), 转 (25);

- (25) 若  $y_{ij}=y_{ji}=0$ , 则  $y_{ij}:=y_{ij}-y_{ji}, y_{ji}:=0$ ; 否则 ( $y_{ij} \leq y_{ji}$ ),  $y_{ji}:=y_{ji}-y_{ij}, y_{ij}:=0$  ( $i, j=1, 2, \dots, n$ );

- (26) 若  $y_{ij}=y_{ji}=0, d_{ij}=d_{ji}=\varepsilon_{ij}$ ; 否则 ( $\max\{y_{ij}, y_{ji}\} > 0$ ), 当  $y_{ij} > 0$  时,  $d_{ij}:=\varepsilon_{ij}, d_{ji}:=\varepsilon_{ij}$ ; 当  $y_{ji} > 0$  时,  $d_{ij}:=\varepsilon_{ij}, d_{ji}:=\varepsilon_{ij}, r_{ij}:=j$  ( $i, j=1, 2, \dots, n$ ),  $k=1$ ;

- (27) 若  $d_{ik}+d_{kj}\Delta\mu'' < d_{ij}$ , 则  $d_{ij}:=\mu'', r_{ij}:=r_{ik}$ ; 否则 ( $\mu'' \geq d_{ij}$ ),  $d_{ij}$  与  $r_{ij}$  都不变 ( $i, j=1, 2, \dots, n$ );

- (28) 求  $\min\{d_{ik} | 1 \leq i \leq n\} \Delta d_{ik}$ , 若  $d_{ik} < 0$ , 则  $s:=u, \lambda'':$  是一个充分大的数, 转到 (29); 否则 ( $d_{ik}=0$ ), 转到 (35);

- (29)  $t:=r_{ik}$ ;

- (30) 若  $y_{st} > 0$ , 则  $\lambda''=\min\{\lambda'', x_{st}\}$ , 转到 (31); ( $y_{st}=0$ ), 转到 (31);

- (31) 若  $t \neq u$ , 则  $s:=t$ , 转到 (29); 否则 ( $t=u$ ),  $s:=u$ , 转到 (32);

- (32)  $t:=r_{ik}$ ;

- (33) 若  $y_{st} > 0$ , 则  $y_{st}:=x_{st}-\lambda''$ , 转到 (34); ( $y_{st}=0$ ),  $y_{st}:=y_{st}+\lambda''$ , 转到 (34);

- (34) 若  $t \neq u$ , 则  $s:=t$ , 转到 (32); 否则 ( $t=u$ ), 转到 (26);

- (35) 若  $k < n$ , 则  $k=k+1$ , 转到 (27); 否则 ( $k=n$ ), 得最优解  $x_{ij}, y_{ij}$  ( $i, j=1, 2, \dots, n$ ), 停。

算法注释:

- (1) 步骤 (1)、(2)、(7)、(9)、(21)、(26)、(27)、(29) 以及步骤 (32) 中的  $r_{ij}$  ( $i, j=1, 2, \dots, n$ ) 起路标的作用, 表示在节点  $i$  到节点  $j$  的路径上, 节点  $i$  首先应该到达的节点是  $r_{ij}$ ;

- (2) 步骤 (2) 和步骤 (3) 执行的是一种改进的 Floyd 算法, 其不但计算出了所有两个节点之间的最小费用路径值, 而且记录了任意两个节点之间的最小费用路径走向;

- (3) 步骤 (4) 至步骤 (13) 是沿最小费用路径按照供需要求安排各车场、配送点以及用户点之间的运量, 获得初始可行解

$x_{ij}, y_{ij} (i, j=1, 2, \dots, n);$

(4)步骤(15)至步骤(35)是寻找可以对可行解进行改进的调优负回路,并沿调优负回路改进可行解,直到达到最优为止。

## 5 例题演示

为了使算法更加直观,下面用一个具体的例题做一演示。

例:已知一般网络的连通情况及相邻两节点间的单位运费(元)如图1所示,其中③、⑨是车场,分别停有5辆和6辆车,单车装载量为30t。各配送点的存储量以及各用户点的需求量分别如表1和表2所示。

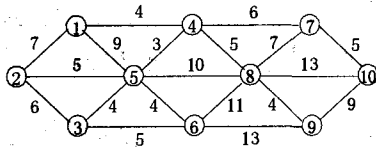


图1 例图

表1 各配送点的存储量

配送点	②	⑧
存储量	70t	75t

表2 各用户点的需求量

用户点	①	⑤	⑥	⑦	⑩
需求量	40t	30t	20t	15t	40t

节点*i*与节点*j*之间空驶路段的费用为 $e_{ij} (i, j=1, 2, \dots, 10)$ 。要求每辆车运完货物之后必须返回原车场。问:如何调度车辆,选择运输路线,使完成所有配送任务的总费用最少?

解:首先,不妨设 $e_{ij} (i, j=1, 2, \dots, 10)$ 相当于运载了1t货物由节点*i*到节点*j*所花的费用,即节点*i*与节点*j*之间的单位运费。

由步骤(1),得到 $(d_{ij}/r_{ij})_{non}$ ,见表3所示。

表3 步骤(1)得到的 $(d_{ij}/r_{ij})_{non}$

0/1	7/2	∞/3	4/4	9/5	∞/6	∞/7	∞/8	∞/9	∞/10
7/1	0/2	6/3	∞/4	5/5	∞/6	∞/7	∞/8	∞/9	∞/10
∞/1	6/2	0/3	∞/4	4/5	5/6	∞/7	∞/8	∞/9	∞/10
4/1	∞/2	∞/3	0/4	3/5	∞/6	6/7	5/8	∞/9	∞/10
9/1	5/2	4/3	3/4	0/5	4/6	∞/7	10/8	∞/9	∞/10
∞/1	∞/2	5/3	∞/4	4/5	0/6	∞/7	11/8	13/9	∞/10
∞/1	∞/2	∞/3	6/4	∞/5	∞/6	0/7	7/8	∞/9	5/10
∞/1	∞/2	∞/3	5/4	10/5	11/6	7/7	0/8	4/9	13/10
∞/1	∞/2	∞/3	∞/4	∞/5	13/6	∞/7	4/8	0/9	9/10
∞/1	∞/2	∞/3	∞/4	∞/5	∞/6	5/7	13/8	9/9	0/10

由步骤(2)和步骤(3),得到 $(d_{ij}/r_{ij})_{non}$ ,见表4所示。

表4 步骤(2)和(3)得到的 $(d_{ij}/r_{ij})_{non}$

0/1	7/2	11/4	4/4	7/4	11/4	10/4	9/4	13/4	15/4
7/1	0/2	6/3	8/5	5/5	9/5	14/5	13/5	17/5	19/5
11/5	6/2	0/3	7/5	4/5	5/6	13/5	12/5	16/5	18/5
4/1	8/5	7/5	0/4	3/5	7/5	6/7	5/8	9/8	11/7
7/4	5/2	4/3	3/4	0/5	4/6	9/4	8/4	12/4	14/4
11/5	9/5	5/3	7/5	4/5	0/6	13/5	11/8	13/9	18/5
10/4	14/4	13/4	6/4	9/4	13/4	0/7	7/8	11/8	5/10
9/4	13/4	12/4	5/4	8/4	11/6	7/7	0/8	4/9	12/7
13/8	17/8	16/8	9/8	12/8	13/6	11/8	4/8	0/9	9/10
15/7	19/7	18/7	11/7	14/7	18/7	5/7	12/7	9/9	0/10

根据表4指出的各车场、配送点以及用户点之间的最小费用路径分配车辆以及运量,即执行步骤(4)至步骤(20),得到初

始可行解为: $x_{14}=20t, x_{21}=60t, x_{25}=10t, x_{45}=20t, x_{47}=10t, x_{54}=10t, x_{65}=10t, x_{7,10}=40t, x_{86}=30t, x_{87}=45t$ ,其余的 $x_{ij}=0$ 。

行车路线分别为:③→②→①→②→③,③→②→①→④→⑤→③,③→②→⑤→③,⑨→⑧→⑥→⑤→④→⑦→⑧→⑨,⑨→⑧→⑦→⑩→⑨。此时成所有配送任务的总费用为1665元。

按照步骤(21)调整运量之后得到:

$x_{14}=20t, x_{21}=60t, x_{25}=10t, x_{45}=10t, x_{47}=10t, x_{54}=10t, x_{65}=10t, x_{7,10}=40t, x_{86}=30t, x_{87}=45t$ ,其余的 $x_{ij}=0$ 。

由步骤(22),得到 $(d_{ij}/r_{ij})_{non}$ ,见表5所示。

表5 步骤(22)得到的 $(d_{ij}/r_{ij})_{non}$

0/1	-7/2	∞/3	4/4	9/5	∞/6	∞/7	∞/8	∞/9	∞/10
7/1	0/2	6/3	∞/4	5/5	∞/6	∞/7	∞/8	∞/9	∞/10
∞/1	6/2	0/3	∞/4	4/5	5/6	∞/7	∞/8	∞/9	∞/10
-4/1	∞/2	∞/3	0/4	3/5	∞/6	6/7	5/8	∞/9	∞/10
9/1	-5/2	4/3	-3/4	0/5	-4/6	∞/7	10/8	∞/9	∞/10
∞/1	∞/2	5/3	∞/4	4/5	0/6	∞/7	-11/8	13/9	∞/10
∞/1	∞/2	∞/3	-6/4	∞/5	∞/6	0/7	-7/8	∞/9	5/10
∞/1	∞/2	∞/3	5/4	10/5	11/6	7/7	0/8	4/9	13/10
∞/1	∞/2	∞/3	∞/4	∞/5	13/6	∞/7	4/8	0/9	9/10
∞/1	∞/2	∞/3	∞/4	∞/5	∞/6	-5/7	13/8	9/9	0/10

执行步骤(23)、(24)、(31),直到 $k=8$ 时, $(d_{ij}/r_{ij})_{non}$ 中出现 $d_{88}=-10<0$ ,见表6所示。

表6  $(d_{ij}/r_{ij})_{non}$ 中出现 $d_{88}=-10<0$

0/1	-7/2	-1/2	4/4	-2/2	4/2	10/4	-7/2	17/2	15/4
7/1	0/2	6/3	11/1	5/5	11/3	17/1	0/3	24/3	22/1
13/2	6/2	0/3	17/2	4/5	5/6	23/2	-6/6	18/6	27/6
-4/1	-11/1	-5/1	0/4	-6/1	0/1	6/7	-11/1	13/1	11/7
-7/4	-14/4	-8/4	-3/4	-9/4	-4/6	3/4	-15/6	9/6	8/4
18/3	11/3	5/3	22/3	4/5	0/6	28/3	-11/8	13/9	22/9
-10/4	-17/4	-11/4	-6/4	-12/4	-6/4	0/7	-17/4	7/4	5/10
-3/7	-10/7	-4/7	1/7	-5/7	1/7	7/7	-10/7	4/9	12/7
31/6	24/6	18/6	35/6	17/6	13/6	41/6	2/6	0/9	9/10
-15/7	-22/7	-16/7	-11/7	-17/7	-11/7	-5/7	-22/7	2/7	0/10

说明网络中存在一条可调整降费的回路:⑧→⑦→④→①→②→③→⑥→⑧,如图2所示。

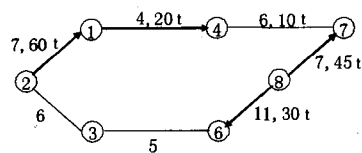


图2 一条可调整降费的回路

沿回路调整各节点之间的运量,即执行步骤(25)至步骤(32),得到这条回路上新的可行解,如图3所示。因此,整个配送网络上新的可行解为:

$x_{14}=10t, x_{21}=50t, x_{25}=10t, x_{26}=10t, x_{45}=10t, x_{65}=10t, x_{7,10}=40t, x_{86}=20t, x_{87}=55t$ ,其余的 $x_{ij}=0$ 。

行车路线分别为:③→②→①→②→③,③→②→①→④→⑤→③,③→②→⑤→③,③→②→③→⑥→③,⑨→⑧→⑥→⑤→④→⑦→⑧→⑨。

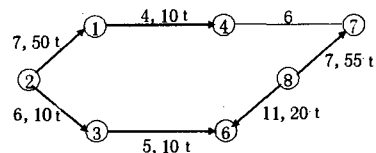


图3 得到这条回路上新的可行解

(下转 208 页)

- nal Processing, 2000, 14(3): 371-385.
- [15] 蒋东翔, 黄文虎, 徐世昌. 分形几何及其在故障诊断中的应用[J]. 哈尔滨工业大学学报, 1996, 28(2).
- [16] Chiang Leo Hao-tien. Fault detection and diagnosis for large-scale systems [D]. UMI, University of Illinois at Urbana-champaign, 2001.
- [17] Russell E L, Chiang L H, Braatz R D. Data-driven Techniques for Fault Detection and Diagnosis in Chemical Processes[M]. London: Springer Verlag, 2000.
- [18] Xu L, Oja E, Suen C Y. Modified hebbian learning for curve and surface fitting[J]. Neural Networks, 1992, 5(3): 441-457.
- [19] Kramer M A. Autoassociative neural networks[J]. Comput Chem Engng, 1992, 16(4): 313-328.
- [20] Dong D, Mcavoy T J. Nonlinear principal component analysis-based on principal curves and neural networks[J]. Computers chem Engng, 1996, 2(1): 65-78.
- [21] David Antory, Uwe Kruger, George W Irwin, et al. Industrial process monitoring using nonlinear principal component models[C]// Second IEEE International Conference on Intelligent Systems, Kine, 2004, 293-298.
- [22] Fourie S H, Vaal P de. Advanced process monitoring using an on-line non-linear multiscale principal component analysis methodology[J]. Computers and Chemical Engineering, 2000, 24(2): 755-760.
- [23] Schölkopf B, Smola A, Müller K. Nonlinear component analysis as a kernel eigenvalue problem[J]. Neural Computation, 1998, 10(5): 1299-1399.
- [24] Lee J M, Yoo C K, Choi S W, et al. Nonlinear process monitoring using kernel principal component analysis[J]. Chemical Engineering Science, 2004, 59(1): 223-234.
- [25] Choi Sang Wook, Lee Changkyu, Lee Jong-Min, et al. Fault detection and identification of nonlinear processes based on kernel PCA[J]. Chemometrics and Intelligent Laboratory Systems, 2005, 75(1): 55-67.
- [26] Cho Ji-Hoon, Lee Jong-Min, Choi Sang Wook, et al. Fault identification for process monitoring using kernel principal component analysis[J]. Chemical Engineering Science, 2005, 60(1): 279-288.
- [27] Choi Sang Wook, Lee In-Beum. Nonlinear dynamic process monitoring based on dynamic kernel PCA[J]. Chemical Engineering Science, 2004, 59(24): 5897-5908.
- [28] Ku W F, Storer R H, Georgakis C. Disturbance detection and isolation by dynamic principal component analysis[J]. Chemometrics and Intelligent Laboratory Systems, 1995, 30(1): 179-196.
- [29] Negiz A, Cinar A. Statistical monitoring of multivariable dynamic processes with state space models[J]. AIChE, 1997, 43(8): 209-221.
- [30] Simoglou A, Martin E B, Morris A J. Statistical performance monitoring of dynamic multivariate processes using state space modelling[J]. Computers and Chemical Engineering, 2002, 26(6): 909-920.
- [31] Zhang Haitao, Tangirala Arun K, Shah Sirish L. Dynamic process monitoring using multiscale PCA[C]// Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering Shaw Conference Center, Edmonton, Alberta, Canada, 1999: 1579-1584.
- [32] Zhou Shaoyuan, Xie Lei, Wang Shuqing. A variable moving window approach for on-line fault diagnosis in industrial processes [C]// Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, P R China, 2004: 1761-1765.
- [33] Zhang Jie, Martins Elaine, Morris A Julian. Fault detection and classification through multivariate statistical techniques [C]// Proceedings of the America Control Conference, Seattle, Washington, 1995: 751-755.
- [34] Thaddeus T Shannon, David Abercrombie, James McNames, et al. Improved process monitoring with independent components [C]// IEEE/SEMI Advanced Semiconductor Manufacturing Conference, 2004: 170-175.
- [35] 王海清, 宋执环, 李平. 改进 PCA 及其在过程监测与故障诊断中的应用[J]. 化工学报, 2001, 52(6): 471-475.

(上接 204 页)

⑥→⑤→④→⑧→⑨, ⑨→⑧→⑦→⑩→⑨。此时完成所有配送任务的总费用为 1 517 元, 费用值减少了 148 元。

重复执行步骤(22)至步骤(32), 直到  $(d_{ij}/r_{ij})_{n,m}$  中的  $d_{ij}=0$  ( $i, j=1, 2, \dots, n$ )。最终整个网络上的运量为:  $x_{21}=40$  t,  $x_{25}=30$  t,  $x_{7,10}=40$  t,  $x_{86}=20$  t,  $x_{87}=55$  t。

最终的行车路线分别为: ③→②→①→②→③, ③→②→⑤→③, ⑨→⑧→⑥→⑨, ⑨→⑧→⑦→⑩→⑨。完成所有配送任务的总费用为 1 326 元, 费用值减少了 339 元。

注: 以上数据都是人为给出的, 并不一定符合实际情况, 只是为了便于计算, 但是算法的本身是具有一般性的。如果采用的是实际数据, 其运算结果一定具有现实效应, 并对实际运输具有很好的指导作用。

## 6 结束语

本文中的算法是以费用最小为目标函数的, 如果以时间最少做为目标函数也同样适用, 只需把目标函数改为: 如何调度车辆, 选择运输路线, 使完成所有配送任务的总时间最少? 由于建立的数学模型不会改变, 因此提出的算法也不变, 只是算法

中的  $c_{ij}$  表示的含义有所不同, 在前一个问题中  $c_{ij}$  表示的是费用, 而在后一个问题中表示的是时间。(收稿日期: 2006 年 8 月)

## 参考文献:

- [1] 郭强. 一般网络上的运输问题及其算法[J]. 系统工程理论方法应用, 2005, 14(1): 92-96.
- [2] 杭省策, 李怀祖. 多车场车流分配的广义指派模型及其分解算法[J]. 西安交通大学学报, 1997, 31(12): 111-116.
- [3] 郭耀煌, 李军. 车辆优化调度问题的研究现状评述[J]. 西南交通大学学报, 1995, 30(4): 376-382.
- [4] 张明善, 唐小我. 多车场满载货运车辆优化调度的网络流算法[J]. 系统工程学报, 2002, 17(3): 216-220.
- [5] 付彤. 无向网络流的最小费用问题[J]. 计算机工程与应用, 2005, 41(28): 88-90.
- [6] Laporte G, Nohet Y, Taillefer S. Solving a family of multi-depot vehicle routing and location-routing problems [J]. Transportation Science, 1988, 22: 161-172.
- [7] Wren A, Holliday A. Computer scheduling of vehicles from one or more depots to a number of delivery points [J]. Operations Research, 1972, 23: 333-344.