

动态团队定向问题的模型及其优化算法

柯良军, 尚可, 冯祖仁

(西安交通大学机械制造与系统工程国家重点实验室, 710049, 西安)

摘要: 针对物流配送系统优化设计中关键难题之一的团队定向问题, 提出了一种部分顾客需求动态到达的动态团队定向问题, 并建立了该问题的模型. 采用把规划周期分成一系列时间段的策略, 将动态问题转化成一系列的静态子问题求解. 提出了一种蚁群算法, 其特点是利用上一时间段的信息来加速算法寻优能力, 并用一种基于分支定价的离线精确性算法来求解动态团队定向问题. 实验结果表明, 与基于分支定价的离线精确性算法相比, 所提出的蚁群算法能在 1 ks 内求解 4 个测试算例, 并且在 2 个算例中得到的最好解优于离线精确性算法的解.

关键词: 动态团队定向问题; 蚁群算法; 分支定价

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 0253-987X(2011)06-0001-06

On the Model and Optimization Algorithm for Dynamic Team Orienteering Problem

KE Liangjun, SHANG Ke, FENG Zuren

(State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A dynamic team orienteering problem of partial dynamic arrival customers is studied, and a model of the problem is established to deal with the team orienteering problem, which is one of the most critical problems in the optimization design for logistic distribution system. The dynamic problem is converted into a series of static sub-problems by partitioning the planning horizon into a series of time segments. An ant colony optimization (ACO) based algorithm is proposed to solve the resulting problem. The prominent characteristic of the algorithm is to use the information obtained from the last time segment to enhance the search behavior. An exact offline algorithm based on branch and price is also presented to solve the problem. Experimental results and comparisons with the exact offline algorithm based on the branch and bound show that the proposed ACO-based algorithm can solve four test instances within 1 000 seconds, and that the solutions of two test instances obtained by the proposed ACO-based algorithm are better than those obtained by the exact offline algorithm.

Keywords: dynamic team orienteering problem; ant colony optimization; branch and price

团队定向问题(TOP)是一类特殊的物流配送路径优化问题. 该问题是指在满足一定约束条件下, 为了服务一组具有一定报酬的顾客, 规划车辆行程以最大化车队的总收益. 1994 年, Butt 等^[1]最先研究团队定向问题, 不过他们称这类问题为多路最大

收益收集问题. 团队定向问题这一名称是由 Chao 等^[2]于 1996 年确立的. 近几年, 国内外学者针对该问题提出了许多高效的算法, 文献[3-4]中分别提出了列生成和分支定价两种精确性算法. 由于团队定向问题是一类 NP-hard 问题, 现有研究主要集中于

启发式算法。Chao 等^[2]提出了 5 步法, Tang 等^[5]提出了一个禁忌算法, Archetti 等^[6]提出了两个禁忌算法和一个变邻域算法, Ke 等^[7]用蚁群算法来求解等。文献[8]综述了现有的主要研究成果。

不过, 现有研究都只考虑静态模型, 即在规划过程中模型是不变的, 实际上随着时间的延续不断有新的客户需要服务。比如某家电公司维修员要给分布在城市各地的顾客提供维修服务, 但是在规划周期的开始(例如一天的上午 9 点钟), 只有一定比率(例如 45%)的顾客需求已知, 其余的顾客需求会在以后的时间内随机到达。只有等到顾客需求到达的时刻, 顾客需求的地理位置和收益才被确定。这就意味着在规划周期开始, 规划好的最优路线可能不再是最优的。在这种情形下, 需要重新规划行程安排以满足需求。为解决该问题, 本文研究这类顾客信息变化的团队定向问题, 简称动态团队定向问题。

所考虑的动态团队定向问题与静态团队定向问题的主要区别在于, 动态团队定向问题是把规划周期分割成一系列时间段, 在每个时间段上解决相应的静态团队定向问题。每个静态团队定向问题只考虑当前存在的并且未被服务的顾客需求。在文献[4]的基础上, 采用一种离线精确性算法——离线分支定价算法来求解。注意到团队定向问题是 NP-hard 的^[7], 这种离线精确性算法的计算复杂度不是多项式的, 只适用于小规模问题。为此, 本文给出动态团队定向问题的数学模型, 提出了一种蚁群优化算法, 它利用上一时间段的信息来加速算法的寻优能力。实验表明, 所提出的蚁群算法能在较短的时间内搜索到高质量的解。

1 问题描述

假设顾客 i 出现的时间为 t_i , 其中 $0 \leq t_i \leq H$, 0 表示规划周期的开始, H 表示规划周期的结束, 在区间 $[0, H]$ 动态出现的顾客信息不能提前预知。令 (x_i, y_i) 为顾客所在的位置(本文假设所有顾客位置分布在一个平面上), r_i 为服务该顾客的收益, 则每个顾客的信息可以用一个向量 $\mathbf{P}_i = (t_i, x_i, y_i, r_i)$ 来表示。车辆数目为 m , 车辆最长运行时间为 $T_{\max}$, 车辆只能从起点出发一次。另外, 不失一般性, 假设车辆速度恒定为 1。

动态团队定向问题可以描述为: 在规划周期内指派车辆去服务已经存在的顾客(包括在 0 时刻已经存在的和以后随机到达的顾客), 使得所服务顾客

的收益最大化。

考虑到当一个新的顾客需求出现时就立即进行路径调整是不实际的, 本文将规划周期划分成一系列时间段, 在每个时间段上的路径规划问题就转化为一个静态团队定向问题。每个时间段上的静态团队定向问题只考虑当前存在的并且未被服务的顾客需求。注意到, 所确定的静态团队定向问题的车辆是多起点类型的, 而不是单起点的, 并且车辆可以转向, 即当车辆在行驶途中, 如果一个新的解产生, 车辆可以立即执行新的规划, 有可能离开当前目的地转向新目的地。

每个时间段上的静态团队定向问题描述如下。

给定一个完全无向图 $G = (V, E)$, 其中 $V = \{1, \dots, n\}$ 是顶点集, $E = \{(i, j) | i, j \in V\}$ 是边集。每个顶点 i 都有收益 r_i 。起点为多起点, 包括出发点 1 和目前车辆所在点, 终点为点 n , 点 i 和 j 之间的距离表示为 c_{ij} 。相应的静态团队定向问题就是找出最多 m 条从多起点出发到终点终止的路径使得所经过点的收益最大化。每个点最多只能经过一次, 每辆车的剩余最长运行时间表示为 $T'_{\max}$ 。根据文献[7]中对静态团队定向问题数学模型的描述, 本文给出每个时间段上的静态团队定向问题的数学模型。

采用以下记号:

- (1) 如果车辆 k 经过点 i , 则 $y_{ik} = 1$, 否则 $y_{ik} = 0$;
- (2) 如果车辆 k 经过边 (i, j) , 则 $x_{ijk} = 1$, 否则 $x_{ijk} = 0$;
- (3) n_t 为时间段数量;
- (4) N_t 为第 t 个时间段的顾客集, $t = 0, \dots, n_t$;
- (5) Q_t 为第 t 个时间段的车辆集, $t = 0, \dots, n_t$ 。

基于以上记号, 第 t 个时间段上的静态团队定向问题的数学模型为

$$\begin{aligned}
 P_t: \max \quad & \sum_{i \in N_t} \sum_{k \in Q_t} r_i y_{ik} \\
 \text{s. t.} \quad & \sum_{i < j} x_{ijk} + \sum_{i > j} x_{jik} = 2y_{jk}, \quad j \in N_t \\
 & \sum_{k \in Q_t} y_{ik} \leq 1, \quad i \in N_t \\
 & \sum_{i \in N_t} \sum_{j > i} c_{ij} x_{ijk} \leq T'_{\max} \\
 & x_{ijk} \in \{0, 1\}, \quad i < j \in N_t \\
 & y_{1k} = y_{nk} = 1; y_{ik} \in \{0, 1\}, \quad i \in N_t
 \end{aligned}$$

其中第 1 个约束规定路径的连通性, 第 2 个约束规定每个顾客最多经过一次, 第 3 个约束描述了时间约束。最后 2 个约束保证了变量必须取值为整数。

2 蚁群算法

蚁群算法(ACO)是一类应用广泛的智能优化算法,其突出特点是正反馈和分布式计算,具有良好的适应性、鲁棒性、分散控制及自组织等特点。下面给出一种求解动态团队定向问题的蚁群算法。

本文所提出的动态蚁群算法是基于划分时间段的思想,把规划周期 $[0, H]$ 划分成 n_{ts} 个等间距的时间段,每个时间段的长度为 H/n_{ts} ,在一个时间段的结束时刻处理在这个时间段内出现的顾客需求。在每个时间段内,确定一个静态团队定向问题,然后用ACO求解相应的静态团队定向问题。

在每个时间段的结束时刻求得一个解,同时也可估计车辆在该规划下经过一个时间段后的位置、行驶时间等信息。

这里需要考虑3个问题:①如何构造时间段上的静态团队定向问题;②如何用ACO算法去解决相应的静态多起点团队定向问题;③需要一种信息素传递策略,把一个时间段求得的规划信息传递给下一个时间段,以利于提高算法搜索性能。下面将详细讨论这3个问题。

整个ACO-DTOP算法的伪代码表示为:

初始化车辆起点为1,在0时刻之前已知的顾客集为 N_0 ;

For $i=0$ to n_{ts}

利用车辆当前位置和 N_i 构造 P_i ;

对 P_i 运行ACO-TOP;

$O_i = \{\text{在过去的 } H/n_{ts} \text{ 内出现的顾客}\}$;

$C_i = \{\text{在将来的 } H/n_{ts} \text{ 内将被服务的顾客}\}$;

更新 $N_{i+1} = N_i \setminus C_i \cup O_i$;

更新车辆位置和行驶时间;

信息素传递策略;

End for

2.1 构造时间段上的静态团队定向问题

设初始顾客集为 N_0 。记第 i 个时间段上的静态团队定向问题为 P_i , P_i 在第 i 个时间段的开始时刻产生,结果在第 i 个时间段的结束时刻求得,记为 S_i ,即求解 P_i 的时间长度为时间段的长度 H/n_{ts} 。在 P_i 求解的过程中, S_{i-1} 执行,记 C_{i-1} 为 S_{i-1} 执行期间被服务的顾客集, O_i 为 S_{i-1} 执行期间新到达的顾客集。 P_i 考虑的所有顾客集记为 N_i ,则 $N_i = N_{i-1} \setminus C_{i-1} \cup O_{i-1}$,整个过程可以由图1描述。

在0时刻之前,首先要解决一个静态团队定向问题,记为 P_0 。值得注意的是,在第一个时间段上,

构造 P_1 ,考虑的顾客集 $N_1 = N_0 \setminus C_0$ 。

2.2 ACO 算法

现在文献[7]的基础上求解每个时间段上的静态团队定向问题,采用ACO-TOP算法解决 P_i 的伪代码如下:

初始化参数、启发信息 η 和信息素 τ ;

s_{gb} 赋值为空;

s_{gb} 赋值为空;

$N_{ni} \leftarrow 0$;

While(计算时间小于 H/n_{ts})

For $i=1$ to n_a (n_a 为蚂蚁数量)

构造解 s_i ;

对 s_i 局部搜索;

End for

$s_{gb} = \arg \max(F(s_1), F(s_2), \dots, F(s_{n_a}))$;

If $F(s_{gb}) > F(s_{gb})$

$s_{gb} \leftarrow s_{gb}$;

$N_{ni} \leftarrow 0$;

Else

$N_{ni} \leftarrow N_{ni} + 1$;

End if

更新信息素;

End while

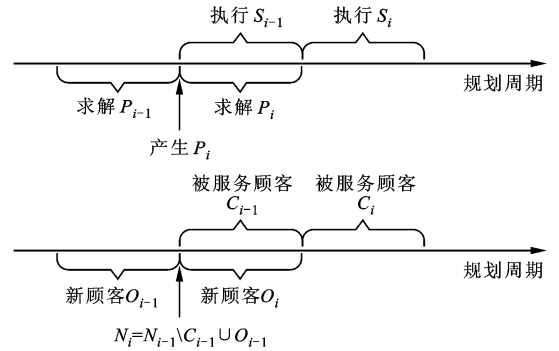


图1 求解方法的结构说明

2.2.1 信息素和启发信息 依据静态团队定向问题的定义,该问题可以表示为一个构造图,其顶点就是原问题的顶点。任意边 (i, j) 都赋有一定量的信息素,用于表征从一个顶点转移到下一个顶点的偏好。

假定一条未完成路径的最后一个点是 i ,对于未被选取的点 j ,启发信息用来表征一种先验的偏好。由于团队定向问题的优化目标是在规定的时间内使得团队的总收益最大化,因此蚂蚁更加偏好于那些具有较高收益且离 i 和 n 较近的点。

启发信息的定义如下

$$\eta(i, j) = \frac{r_j}{c_{ij}} \exp(\gamma \theta_{ij}) \quad (9)$$

式中: r_j 为 j 点的收益; $\theta_{ij} = (c_{ij}^2 + c_m^2 - c_m^2) / 2c_{ij} c_m$, c_{ij} 为点 i 和 j 之间的距离; γ 是一个参数, 它决定 θ_{ij} 的重要性。

2.2.2 解的构造 在构造解的过程中, 蚂蚁相当于一个具有简单智能的决策者, 它负责为每个车辆选取一条可行路径(即在规定时间内从起点出发到达终点的路径)。具体而言, 在每个构造步, 蚂蚁选取一个车辆并为该车辆选取一个可行点, 直到所有的车辆都到达终点。

本文利用文献[7]中提出的串行法构造一个解。在这种方法中, 蚂蚁先为一个车辆选择一条可行路径, 然后再为下一个车辆选择一条可行路径, 直到所有车辆都安排了一条可行路径。

蚂蚁按照以下决策规则概率地选择一个点

$$p(v_{k+1} = v \mid \tau, v_k = u) = \begin{cases} \frac{\tau(u, v)^a \cdot \eta(u, v)^\beta}{\sum_{w \in C_u} \tau(u, w)^a \cdot \eta(u, w)^\beta}, & v \in C_u \\ 0, & \text{其他} \end{cases}$$

式中: C_u 是可选择点集; v_k 表示第 k 步选取的点。

2.2.3 信息素的更新 一旦所有的蚂蚁都构造了一个解, 信息素按照最大最小蚂蚁系统给出的规则更新。具体而言, 每个边的信息素按照如下方式更新

$$\tau(u, v)^{l+1} = \rho \tau(u, v)^l + \Delta \tau(u, v)$$

$$\tau(u, v)^{l+1} < \tau_{\min}, \quad \text{那么} \quad \tau(u, v)^{l+1} = \tau_{\min}$$

$$\tau(u, v)^{l+1} > \tau_{\max}, \quad \text{那么} \quad \tau(u, v)^{l+1} = \tau_{\max}$$

式中: $\tau_{\min}$ 和 $\tau_{\max}$ 分别是信息素的下界; $\tau(u, v)^l$ 是边 (u, v) 在第 l 次循环时的信息素。如果 (u, v) 在 l 代被最优蚂蚁经过, 则 $\Delta \tau(u, v)$ 等于 $F(s_{\text{best}})$, 否则 $\Delta \tau(u, v) = 0$ 。最优解 s_{best} 可能是本次迭代获得的最优解 s_{gb} , 也可能是当前最优解 s_{gb} 。质量函数由下式给出

$$F(x) = \sum_{i=2}^{n-1} \sum_{k=1}^m r_i y_{ik} / \sum_{i=2}^{n-1} r_i$$

设置信息素上下界的目的是为了 避免停滞, 它们按照下式设置为

$$\tau_{\max} = F(s_{\text{gb}}) / (1 - \rho)$$

$$\tau_{\min} = (1 - P_{\text{best}}^{1/n}) / ((n/2 - 1) P_{\text{best}}^{1/n}) \tau_{\max}$$

式中: P_{best} 是参数, $0 \leq P_{\text{best}} \leq 1$ 。

2.2.4 局部搜索 局部搜索常用来改善蚁群算法的性能, 但是在应用中要折中考虑解的质量和计算

时间。本文利用文献[7]中的局部搜索, 其基本思想是: 先用 2-opt 法缩短解中的每条路径, 再插入尽可能多的可行点。在算法中, 局部搜索对每个解进行改进, 它的执行时间是在所有蚂蚁都构造完解之后且在信息素还未更新之前。

2.3 信息素传递策略

在每个时间段的结束时刻, 通过 ACO 算法求得相应的静态团队定向问题的解, 这时信息素矩阵包含了当前静态团队定向问题解的信息。

考虑到前后两个时间段上的团队定向问题的相关性, 上一时间段的信息素可以传递给下一个时间段的静态团队定向问题。这一继承操作使得下个时间段的静态团队定向问题的信息素不用重新开始初始化, 这样有助于加速寻优。

如果 (i, j) 同时出现在上一时间段和当前时间段中, 则相应的信息素值按照下式更新

$$\tau(i, j) = \xi \tau(i, j)^{\text{old}}$$

式中: ξ 是控制信息素传递的参数; $\tau(i, j)^{\text{old}}$ 是上一时间段结束时的信息素值。

对于新加入的顾客, 相应的信息素矩阵中的值初始化为 $\tau_0 = (\tau_{\min} + \tau_{\max}) / 2$ 。

2.4 算法的计算复杂度

由于节点数为 n 、车辆数为 m 的静态团队定向问题的蚁群算法的计算复杂度是 $O(n^2 m)$, 因此所提出的蚁群算法的计算复杂度也是 $O(n^2 m)$ 。

3 分支定价算法

分支定价算法是一类求解组合优化问题的精确性算法, 主要由列生成算法和分支定界算法两部分组成。其基本思想是首先通过列生成算法求得一个线性松弛最优解, 这个解为后续的分支定界算法提供了一个很好的上界(或者下界), 然后通过分支定界算法求得问题的整数解。

文献[4]利用分支定价算法求解静态团队定向问题。静态团队定向问题是一个混合整数规划问题, 首先通过线性松弛策略把混合整数规划问题转化为一个线性规划主问题, 子问题为求解一个有资源约束的基本最短路径问题。利用文献[9]中求解动态车辆路径问题的一些思想, 本文采用一种离线的分支定价算法来解决动态团队定向问题, 与 ACO 方法类似, 该方法同样是基于划分时间段的思想, 把规划周期 $[0, H]$ 划分成 n_s 个等间距的时间段。在每个时间段上, 确定一个静态团队定向问题, 然后用分支定价算法求解相应的静态团队定向问题。

由于团队定向问题是一类 NP-hard 问题,即使是求解顾客数为 50 的问题,分支定价算法也需要很长的计算时间^[4],而对于问题规模较大的动态团队定向问题,分支定价算法往往难以适用. 本文采用一种离线的分支定价算法来求解,算法求解每个时间段的时间限制在 2 h 内.

4 实验结果

4.1 算例说明

本文采用 Chao^[2]中的测试数据,所采用的数据集中提供了车辆数量 m 、车辆的最大行驶时间 $T_{\max}$ 、节点数量 n 、节点的收益 r_i 和地理位置 (x_i, y_i) . 为实现动态团队定向问题,本文在原始数据集中加入了以下新的设定.

(1)顾客请求出现时间. 把数据集分为两部分,一部分是在 0 时刻之前存在的顾客,请求时间均设为 0;另一部分是动态到达的顾客,请求时间在规划周期 $[0, H]$ 内随机生成.

(2)规划周期 $[0, H]$ 内的时间段数量. 由于动态团队定向问题属于动态问题,为了使系统能够快速响应顾客的请求,时间段数量应设置合理. 在实验中,时间段数量设置为 10.

本文 4 个算例分别表示为 Data1、Data2、Data3 和 Data4,它们分别由文献[2]中算例 $p3.2. j$ 、 $p5.3. n$ 、 $p5.3. p$ 和 $p7.4. t$ 生成.

4.2 实验分析

算法用 C++ 编写,在 Pentium(R) 4 CPU 3.00 GHz,1.50 GB 内存的机器上测试.

4.2.1 参数的选择 Data1、Data2 和 Data3 的时间段长度均设置为 30 s,Data4 的时间段长度设置为 100 s.

为使得蚁群算法较满意地收敛,算法中的参数设置为:蚂蚁数为 20,在每个时间段上算法运行到该时间段结束时刻终止; $\alpha = 1, \rho = 0.98, P_{\text{best}} = 0.05, \xi = 0.8$. 参数 β 和 γ 对于算法性能起着关键的作用,实验结果表明,当两者都取为 0.5 时,算法性能较好. 每个算例测试 10 次.

4.2.2 结果分析 表 1 给出了 4 个测试算例的实验结果,同时给出了由分支定价算法求得的结果.

由表 1 可以看出,动态蚁群算法求解 Data1 的最好值与分支定价算法的结果相同,平均值与最好值相差不到 2%. 动态蚁群算法求解 Data2 和 Data3 的最好值比分支定价算法的结果要好,这主要是因为求解结果与初始规划路径有关,不同的初始规划

路径在未来时间段内可能求得不同的规划结果. 由于本文没有考虑未来顾客信息,算法求得的只是当前的局部最优解,不一定是全局最优解. 因此,在某些情况下,动态蚁群算法可能求得比分支定价算法更好的解. 在 Data4 算例上,分支定价算法在规定时间内(2 h)内无法求得问题解,而蚁群算法能够求得较好的解,远优于贪婪算法得到的解(贪婪算法求解 Data4 的解为 567.0).

表 1 两种算法求解的实验结果

算例	m	N	蚁群算法解		分支定价算法解
			平均值	最好值	
Data1	2	21	373.3	380.0	380.0
Data2	3	34	656.7	660.0	655.0
Data3	3	50	908.3	915.0	910.0
Data4	4	80	947.3	954.0	—

注: N 为初始顾客数.

实验结果表明,动态蚁群算法在实时性和有效性方面都达到了很好的效果.

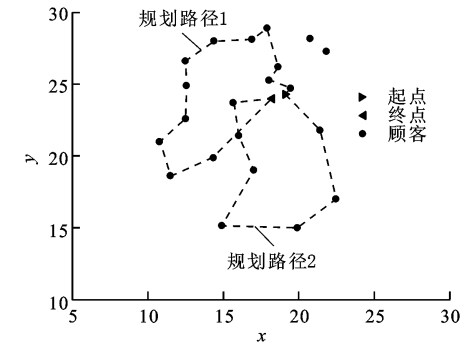
图 2 给出了动态蚁群算法求解 Data1 的最好解. 表 2 给出了动态蚁群算法求解 Data1 的优化路径结果,其中 1 表示起点,车辆当前位置及过去位置作为一种虚拟任务点加入到顾客点集中,规划路径的开始标号为车辆当前位置,结尾标号为终点,随着新顾客的加入和车辆虚拟点的加入,终点标号不断变化,实际路径表示车辆已经经过的轨迹,规划路径表示车辆将要经过的轨迹.

5 结 论

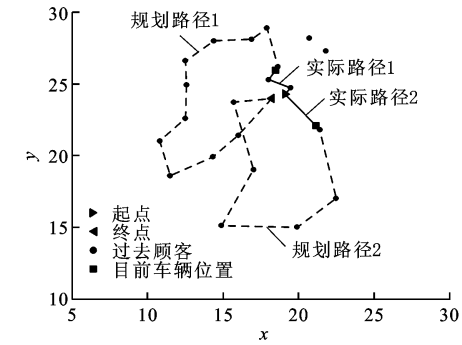
本文在建立动态团队定向问题模型的基础上,把动态问题转化为一系列静态问题求解,并提出了一种解决该问题的蚁群算法. 通过与基于分支定价的精确性算法相比较,实验结果表明了所提算法的有效性.

但是,本文研究的动态团队定向问题没有考虑未来顾客信息,仅考虑当前时刻下已经存在的顾客,在每个时间段上只是局部寻优,因此后续研究将要考虑如何引入未来顾客信息以便全局优化. 同时,本文假定顾客分布在欧氏空间的平面上,两点间的距离为欧氏距离,这显然与现实环境有一定差距,后续研究还包括基于地理信息系统的路径优化问题以及

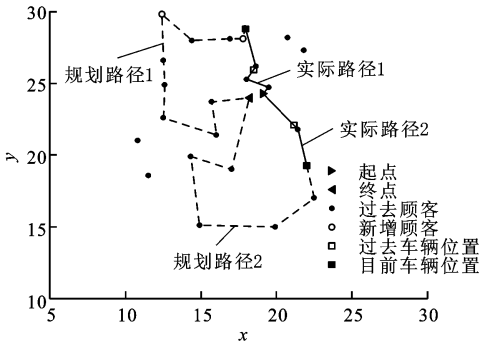
高效的优化算法.



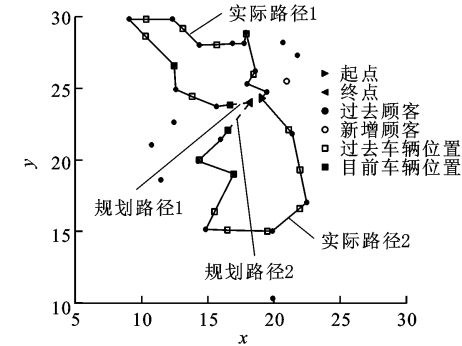
(a) $i=0$



(b) $i=1$



(c) $i=2$



(d) $i=9$

图 2 动态蚁群算法在 Data1 算例上的结果演示

表 2 Data1 的动态蚁群算法优化路径

<i>i</i>	路径
0	规划路径 1:1 13 17 18 4 3 6 2 8 21 12 19 16 23
	规划路径 2:1 14 5 7 15 9 10 20 23
1	实际路径 1:1 13 17 23*
	实际路径 2:1 24*
	规划路径 1:23* 18 4 3 6 2 8 21 12 19 16 10 25
2	规划路径 2:24* 14 5 7 15 9 20 25
	实际路径 1:1 13 17 23* 18 27*
	实际路径 2:1 24* 14 28*
3	规划路径 1:27* 4 26 3 6 25 2 8 21 10 20 29
	规划路径 2:28* 5 7 15 16 9 29
	实际路径 1:1 13 17 23* 18 27* 4 26 3 30*
4	实际路径 2:1 24* 14 28* 5 31*
	规划路径 1:30* 6 25 29 2 8 20 29
	规划路径 2:31* 7 15 9 16 10 29
5	实际路径 1:1 13 17 23* 18 27* 4 26 3 30* 6 34*
	实际路径 2:1 24* 14 28* 5 31* 7 35*
	规划路径 1:34* 25 29 2 8 20 36
6	规划路径 2:35* 15 9 16 10 36
	实际路径 1:1 13 17 23* 18 27* 4 26 3 30* 6 34 25 37*
	实际路径 2:1 24* 14 28* 5 31* 7 35 38*
7	规划路径 1:37* 29 2 8 20 39
	规划路径 2:38* 15 9 16 10 39
	实际路径 1:1 13 17 23* 18 27* 4 26 3 30* 6 34 25 37* 29 41*
8	实际路径 2:1 24* 14 28* 5 31* 7 35 38* 15 42*
	规划路径 1:41* 2 8 20 43
	规划路径 2:42* 9 16 10 43
9	实际路径 1:1 13 17 23* 18 27* 4 26 3 30* 6 34 25 37* 29 41* 44*
	实际路径 2:1 24* 14 28* 5 31* 7 35 38* 15 42* 45*
	规划路径 1:44* 2 8 20 46
10	规划路径 2:45* 9 16 10 46
	实际路径 1:1 13 17 23* 18 27* 4 26 3 30* 6 34 25 37* 29 41* 44* 2 8 46*
	实际路径 2:1 24* 14 28* 5 31* 7 35 38* 15 42* 45* 9 47*
11	规划路径 1:46* 20 48
	规划路径 2:47* 16 10 48
	实际路径 1:1 13 17 23* 18 27* 4 26 3 30* 6 34 25 37* 29 41* 44* 2 8 46* 20 49*
12	实际路径 2:1 24* 14 28* 5 31* 7 35 38* 15 42* 45* 9 47* 16 10 50*
	规划路径 1:49* 51
	规划路径 2:50* 51

注:带 * 数字为车辆虚拟任务点;不带 * 数字为实际顾客点.

参考文献:

[1] BUTT S E, CAVALIER T M. A heuristic for the multiple path maximum collection problem[J]. Computers and Operations Research, 1994, 21(1):101-111.

[2] CHAO I M, GOLDEN B L, WASIL E A. The team orienteering problem[J]. European Journal of Operational Research, 1996, 88(3): 464-474.

[3] BUTT S, RYAN D. An optimal solution procedure for the multiple path maximum collection problem using column generation[J]. Computer and Operations Research, 1999, 26(4):427-441.

[4] BOUSSIER S, FEILLET D, GENDREAU M. An exact algorithm for team orienteering problems [J]. 4OR: A Quarterly Journal of Operations Research, 2007, 5(3): 211-230.

[5] TANG Hao, MILLER-HOOKS E. A tabu search heuristic for the team orienteering problem[J]. Computers and Operations Research, 2005, 32(6):1379-1407.

[6] ARCHETTI C, HERTZ A, SPERANZA M G. Meta-heuristics for the team orienteering problem[J]. Journal of Heuristics, 2007, 13(1):49-76.

[7] KE Liangjun, ARCHETTI C, FENG Zuren. Ants can solve the team orienteering problem[J]. Computers and Industrial Engineering, 2008, 54(3):648-665.

[8] VANSTEENWEGEN P, SOURIAU W, OUDHEUSDEN D. The orienteering problem: a survey[J]. European Journal of Operational Research, 2011, 209(1):1-10.

[9] CHEN Zhilong, XU Hang. Dynamic column generation for dynamic vehicle routing with time windows [J]. Transportation Science, 2006, 40(1):74-88.

(编辑 荆树蓉 武红江)