可行解优先蚁群算法对车辆路径问题的求解①

A Feasible Priority Solution of Vehicle Routing Problem with Ant Colony Algorithm

白　明　张　健　(五邑大学 信息学院 广东 江门 529020)

摘　要： 针对车辆路径问题，给出了一种利用蚁群算法求解该问题的新方法。借鉴 K-TSP 问题的求解方法，优先构造可行解，通过对较优解路径上信息素的增强，最终得到问题的最优解或较优解。实验结果表明，用本方法求解车辆路径问题，简化了求解过程，缩短了求解时间，解决了无可行解的问题。

关键词： 车辆路径问题 蚁群算法 K-TSP 可行解 信息素

1　引言

在物流配送供应领域中，一个常见问题是：有一批客户，各客户点的位置坐标和货物需求已知，供应商具有若干可供派送的车辆，运载能力给定，每辆车都从起点出发，完成若干客户点的运送任务后再回到起点。现要求以最少的车辆数、最小的车辆总行驶里程来完成货物的派送任务，该问题被称为车辆路径问题（vehicle routing problem，VRP），有时也称之为“车辆计划”、“货车派遣”等。

近年来，许多学者利用蚁群算法对各种 VRP 进行了大量的研究，设计了各种类型的蚁群算法[1-3]，具有代表性的求解 VRP 的自适应蚁群算法的设计思路是[4]：以求解 TSP 的蚁群算法为基础，考虑 VRP 的具体要求，对算法的选择机制、更新机制以及协调机制做进一步改进，引入自适应的转移策略，并融合节约法，以克服基本蚁群算法计算时间长、易出现停滞等缺陷。

2　自适应蚁群算法求解 VRP 的缺陷

自适应蚁群算法解决车辆路径问题时，针对无可行解问题，通过采取“事前”预防措施和“事后”治理措施，以确保至少得到所求问题的一个可行解。“事前”预防措施主要是通过增加蚂蚁数和蚂蚁初始分布均匀的策略，当问题规模较大时，增多蚂蚁数，必然导致搜索时间增长，而且仍然会有大量蚂蚁所找的回路是相同的。对此问题，采用近似化策略，即“事后”治理措施，以保证至少得到所求问题的一个可行解。然而，事后的治理是非常复杂烦琐的，首先要判断解的集合是“未满非可行解”还是“溢出非可行解”甚至为“混合非可行解”，然后对溢出非可行解中的公共节点“擦除”，将溢出非可行解转换成未满非可行解，再应用节约法将未在路径中的节点插入到路径中。对于混合非可行解的处理就更为复杂。

以上问题的复杂性主要是车辆的载重量限制所造成的，假若车辆的载重量不受限制，则 VRP 就完全等同 TSP。因此，在处理 VRP 问题时，借鉴蚁群算法在 K-TSP 问题中的应用[5]，将 VRP 转化为 K-TSP 问题，与 K-TSP 问题不同的是，K 的数值由车辆的载重量及各点的需求所决定，是动态变化的。而 K-TSP 问题中的 K 值是事先确定的。当各点的需求量均相同时，类似于求解 K-TSP 问题。在求解 VRP 时，优先考虑可行解的获取，在获得可行解的前提下，再寻找最优解。

3　可行解优先的蚁群算法

蚁群算法对 K-TSP 问题的求解时，采用 k 只蚂蚁

① 收稿时间:2008-07-17

共同构造问题的一个解，即一组蚂蚁（共 k 只）与 K-TSP 问题的一个可行解相对应[5]。基于此方法，在求解 VRP 问题时，车辆在配送中心装满货物，按照确定性选择和随机性选择相结合的策略选择下一个节点，当车辆到达第一个节点后，减去该节点所需货物，再在剩余满足需求的节点集合中按照前面的策略选择下一个节点，当车辆货物为空或剩余货物不能满足所有剩余节点集合中任一节点的需求时，车辆返回配送中心，重新装满货物，再在剩余节点集合中进行配送，重复上面的过程，直至所有节点得到配送服务，得到问题的一个解。

在算法设计中，每次给定一组蚂蚁，每个蚂蚁完成所有节点的配送，当一组蚂蚁全部完成配送，即得到一组可行解，从中找出路径最短的那只蚂蚁，将其所走路径的信息素加强，并记录所走节点的顺序。

可行解优先的蚁群算法主要由选择策略，局部搜索机制和信息素全局更新组成。

3.1 选择策略

在车辆路径问题中，第 k 只蚂蚁从城市 i 转移到城市 j 的状态转移概率定义为

$$P_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\cdot\eta_{ij}^{\beta}(t)}{\sum_{s\in allowed_k}\tau_{is}^{\alpha}(t)\cdot\eta_{is}^{\beta}(t)}, & if\ j\in allowed_k\\ 0, & otherwise\end{cases} \tag{1}$$

其中 $allowed_k$，表示蚂蚁 k 下一步允许选择的城市，$\tau_{ij}(t)$ 表示 t 时刻在城市 i 与城市 j 之间路径上信息素的浓度，$\eta_{ij}=1/d_{ij}$ 表示路径的启发信息，d_{ij} 表示边（i，j）之间的距离。

3.2 局部搜索机制

第 k 只蚂蚁完成一次配送，则对该解使用 2-opt 法进行局部优化，以加快局部最优解的得到。

3.3 信息素全局更新

当一组 m 只蚂蚁全部完成配送，即得到一组可行解，从中找出路径最短的那只蚂蚁，将其所走路径的信息素加强，表示为

$$\tau_{ij}(t+1)=\rho\cdot\tau_{ij}(t)+\Delta\tau_{ij} \tag{2}$$

$$\Delta\tau_{ij}=\sum_{k=i}^{n}\Delta\tau_{ij}^{k} \tag{3}$$

其中，$\Delta\tau_{ij}^{k}$ 表示蚂蚁 k 在本次循环中经过城市 i 与城市 j 之间路径上所留下的信息素，其计算方法可根据计算模型而定，在求解 VRP 问题时，通常利用整体信息，即 Ant-Cycle 模型

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, & kth\ ant\ goes\ from\ i\ and\ j\\ 0, & otherwise\end{cases} \tag{4}$$

ρ 表示信息素的持久率。

3.4 算法步骤

根据以上的讨论，可行解优先的蚁群算法描述如下：

①初始化参数：蚂蚁数 m、NC(循环次数)、$\tau_{ij}(i,j=1,2,\cdots,n)$ 初始值相同，Q、ρ，每个节点的需求量分别为 Wk（k=1,2,…，n 为节点数），每只蚂蚁的初始载重量 Wm 为车辆的载重量，而且车辆的载重量大于等于任何一个节点的需求量，即 Wm>=Wn。

②将 m 只蚂蚁都放置在配送中心，每只蚂蚁的载重量均为 Wm，每只蚂蚁按照公式（1）以累积概率选择下一个城市 k，当完成第一次选择，便对蚂蚁的载重量进行更新，$W_m \Leftarrow W_m - W_k$，同时更新禁忌表，$tabu_m^n \Leftarrow k$。

③在后续的城市选择时，首先判断是否满足条件：Wm>=Wn，若不满足，则转步骤（2），即蚂蚁回到配送中心，重新装满货物；否则，在满足条件的节点中，按照公式（1）以累积概率继续选择下一个城市 k，同样更新载重量和禁忌表。

④当 $k\in tabu_k^n$，即第 k 只蚂蚁完成所有节点的配送任务，计算第 k 只蚂蚁完成本次配送成本，找出配送路径，与 2-opt 法进行局部优化，保存优化后的配送路线。

⑤当 $k\in tabu_m^n$，即 m 只蚂蚁全部完成配送任务，对 m 只蚂蚁完成本次配送成本进行比较，找出较优配送路径，按公式(2)、(3)更新信息素。

⑥NC=NC+1，若 NC 小于规定的循环次数且无退化行为（即找到的都是相同解），则转步骤（2）；否则，输出最好的解，算法结束。

4　可行解优先的蚁群算法

以参考文献[6]中 243 页的算例进行分析比较：设有 19 个客户随机分布于弧段长为 10km 的正方形区域内，配送中心位于区域正中央,其坐标为（0，0）；各客户的需求由计算机随机产生，车辆载重量为 9 吨，

实验基础数据如表 1 所示。

表 1　实验基础数据[6]

客户编号	0	1	2	3	4	5	6	7	8	9
横坐标(km)	0	0	0	2	3	3	4	4	1	1
纵坐标(km)	0	1	3	2	3	1	0	1	2	1
配送量(t)	0	1.5	1.8	2.0	0.8	1.5	1.0	2.5	3.0	1.7
客户编号	10	11	12	13	14	15	16	17	18	19
横坐标(km)	1	3	−3	2	1	2	2	1	−3	−1
纵坐标(km)	3	4	0	0	−3	−1	1	−4	2	−1
配送量(t)	0.6	0.2	2.4	1.9	2.0	0.7	0.5	2.2	3.1	0.1

用自适应蚁群算法进行测试，其参数设置为：$M=60$，$NC_{\max}=50$，$\tau_{ij}=10$，$\alpha=1$，$\beta=1$，$\rho=0.15$，$Q_1=10$，$Q_2=50$，$Q_3=100$。运行 10 次，其结果如表 2 所示。

对于同样的算例，用可行解优先的蚁群算法进行测试，其参数设置为：$M=15$，$NC_{\max}=100$，$\tau_{ij}=10$，$\alpha=1$，$\beta=5$，$\rho=0.9$，$Q_1=10$。运行 10 次，其结果如表 2 所示。

表 2 两种算法结果比较

算法及解的情况	自适应蚁群算法	可行解优先蚁群算法
最小配送距离	41.86	42.32
最大配送距离	47.37	42.36
车辆数目	4	4
平均搜索终解代数	32.3	34.6
平均计算时间	2.14	0.22

由表 2 可见，采用可行解优先的蚁群算法得到了更高的求解质量，在 10 次求解中均得到了最好解：42.32km（与自适应蚁群算法所求的 41.86km 的差异是由于计算的误差所造成），对应的四条配送路径如图 1 所示，分别为：

路线 1：0→18→0；

路线 2：0→1→17→14→8→0；

路线 3：0→12→6→7→4→3→19→0；

路线 4：0→2→10→11→16→13→5→15→9→0。

与自适应蚁群算法的配送路径完全相同，在 10 次求解中，平均求解时间为 0.22 秒，具有更短的搜索时间；平均首次搜到终结代数为 34.6 代，自适应蚁群算法的平均首次搜到终结代数为 32.3 代，在平均首次搜到终结代数接近的情况下，可行解优先的蚁群算法的平均求解时间仅为 0.22 秒，约为自适应蚁群算法平均求解时间的十分之一，主要是简化了求解过程，表现出更高的求解效率。

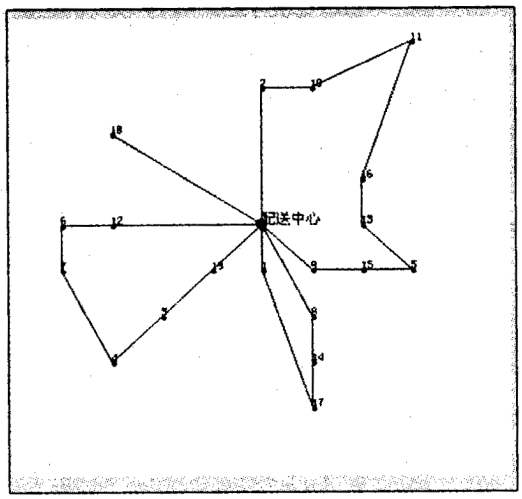

图 1　19 个客户的配送线路图

为了对可行解优先的蚁群算法做进一步的验证，选用 51 个城市的 TSP 问题 eil51 为例进行实验，以城市 1 为配送中心，其余 50 个城市的需求量都为 1 吨，车辆的载重量分别为 25 吨、17 吨和 10 吨，迭代 1000 次，其余参数与上例相同，实验结果分别为：

车辆载重量为 25 吨：需要 2 辆车，路径长度为：447.40，配送路线见图 2。

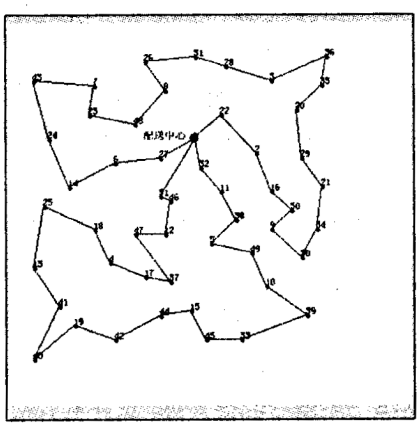

图 2　车辆载重量为 25 吨的配送路径分布图

车辆载重量为 17 吨：需要 3 辆车，路径长度为：472.23，配送路线见图 3。

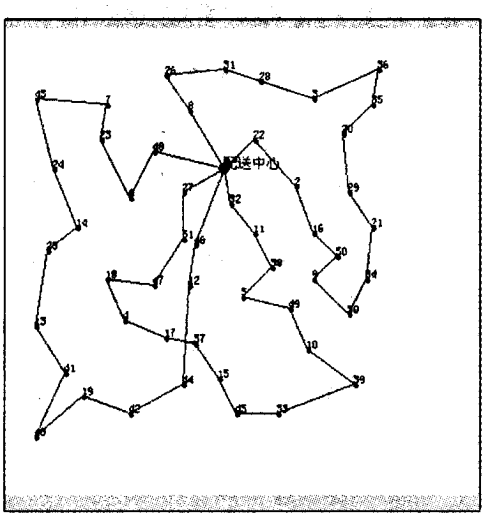

图 3 车辆载重量为 17 吨的配送路径分布图

车辆载重量为 10 吨：需要 5 辆车，路径长度为：570.99，配送路线见图 4。

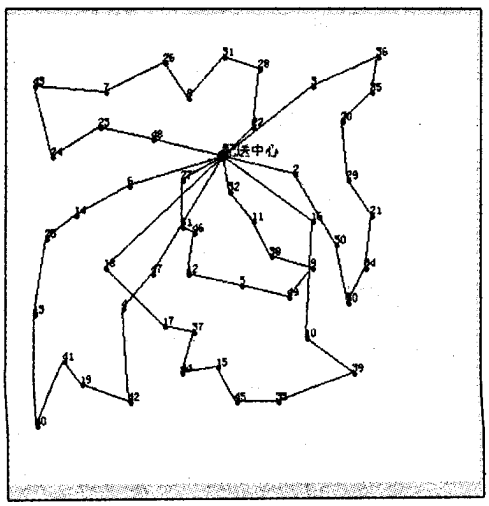

图 4 车辆载重量为 10 吨的配送路径分布图

从配送路径分布图可以看出，配送路线以配送中心为起点向四周成扇形分布。

5 结束语

在求解 VRP 问题时，借鉴蚁群算法对 K-TSP 问题的求解思路，提出了可行解优先的蚁群算法，实验结果表明，可行解优先的蚁群算法是行之有效的，较好地解决了无可行解的问题，与自适应蚁群算法相比具有更高的求解质量，更短的搜索时间以及更高的求解效率。

参考文献

1 Bullnheimer B, Hartl R F, Strauss C. An improved ant system algorithm for the vehicle routing problem. Technical Report POM-10/97, Institute of Management Science, University of Vienna, Austria,1997,Ann. Oper. Res. 89,1999.

2 Bullnheimer B, Hartl R F, Strauss C. Applying the ant system to the vehicle routing problem.Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic Publishers, Boston, MA, 1999:285–296.

3 Gambardella LM,TailllardĚ, Agazzi G. MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows.New Ideas in Optimization, McGraw-Hill, London, UK, 1999：63–76.

4 刘志硕,申金升,柴跃廷.基于自适应蚁群算法的车辆路径问题研究.控制与决策,2005,20(5):562–566.

5 黄席樾,胡小兵.蚁群算法在 K-TSP 问题中的应用.计算机仿真,2004, 21(12):162–164.

6 段海宾.蚁群算法原理及其应用.北京:科学出版社,2005:238–244.