

求解旅行商问题的混合蚂蚁算法

陈文兰¹, 戴树贵^{1,2}

(1. 滁州学院 数学系, 安徽 滁州 239000;

2. 华东师范大学 计算机科学技术系, 上海 200062)

摘要:旅行商问题是一个经典的 NP 问题, 文中给出了一个有效的求解旅行商问题的混合蚂蚁算法。算法设计了初始信息素量设置方案和信息素的更新方法, 限制了蚂蚁转移的目标城市数, 并使用 2-Opt 方法对路径进行优化。数据实验表明, 该算法是有效的。

关键词:旅行商问题; 蚂蚁算法; 2-Opt; 局部优化

中图分类号: TP18; TP301.6

文献标识码: A

文章编号: 1673-629X(2007)07-0110-04

A Hybrid Ant Colony Algorithm for Solving Traveling Salesman Problem

CHEN Wen-lan¹, DAI Shu-gui^{1,2}

(1. Department of Mathematics, Chuzhou University, Chuzhou 239000, China;

2. Department of Computer Science and Technology, East China University, Shanghai 200062, China)

Abstract: An efficient hybrid ant colony algorithm is proposed for the classical NP-hard problem—Traveling Salesman Problem (TSP). Initial pheromone setting method and pheromone updating method are designed. The number of destination cities is limited for the ant's transfer, and the routes are optimized by 2-Opt method when all ants have found effective route. Data experiments show that the algorithm is efficient.

Key words: traveling salesman problem; ant colony algorithm; 2-Opt method; local optimization

0 引言

旅行商问题 (Traveling Salesman Problem, TSP) 是一个经典的组合优化问题, 在交通运输、电路板线路设计以及物流配送等领域有着广泛的应用。由于该问题的可行解是所有顶点的全排列, 随着顶点数的增加, 会产生组合爆炸, 它是一个 NP 完全问题。早期的研究者使用精确算法求解该问题, 常用的方法包括: 分枝定界法^[1]、线性规划法^[2]和动态规划法^[3]等。但是, 随着问题规模的增大, 精确算法将变得无能为力, 因此, 在后来的研究中, 国内外学者重点使用近似算法或启发式算法, 主要有遗传算法、蚁群算法、模拟退火算法和禁忌搜索算法^[4-6]等。文中首先介绍 TSP 的模型, 然后构造了一个求解 TSP 的蚂蚁算法, 最后通过数据实验讨论了算法的有效性和参数设置对算法性能的影响。

1 TSP 的数学模型

经典 TSP 可以描述为: 有 N 个城市, 一个旅行商从某个城市出发, 遍历所有城市后回到出发点, 求一条经过所有城市仅一次的最短遍历路径。从图论的角度来看, 该问题实质是在一个带权完全无向图中, 找一个权值最小的 Hamilton 回路。其精确描述为: $G = (V, E)$ 为一个带权图, $V = \{1, 2, \dots, N\}$ 为顶点集, $E = \{e_{ij} = (i, j) \mid i, j \in V, i \neq j\}$ 为边集。 $d_{ij} (i, j \in V, i \neq j)$ 为顶点 i 到顶点 j 的距离, 其中 $d_{ij} > 0$ 且 $d_{ij} \neq \infty$, 同时 $d_{ij} = d_{ji} (i, j \in V)$, 则经典 TSP (Classical TSP) 的数学模型为:

$$\min F = \sum_{i \neq j} d_{ij} \times x_{ij} \quad (1)$$

$$\text{s. t. } x_{ij} = \begin{cases} 1, & \text{边 } e_{ij} \text{ 在最优路径上} \\ 0, & \text{边 } e_{ij} \text{ 不在最优路径上} \end{cases} \quad (2)$$

$$\sum_{i \neq j} x_{ij} = 1, i \in V \quad (3)$$

$$\sum_{i \neq j} x_{ij} = 1, j \in V \quad (4)$$

$$\sum_{i, j \in S} x_{ij} = |S|, S \text{ 为 } G \text{ 的子图} \quad (5)$$

其中 $|S|$ 为图 S 的顶点数。式(1)为目标函数, 求经过

收稿日期: 2006-10-14

基金项目: 安徽高校省级自然科学基金(2006KJ253B, 2006KJ085B)

作者简介: 陈文兰(1972-), 女, 浙江绍兴人, 硕士, 讲师, 研究方向为软件与算法; 戴树贵, 博士研究生, 副教授, 研究方向为软件与算法、高级物流。

所有顶点的回路的最小距离。式(2)~(4)限定回路上每个顶点仅有一条入边和一条出边。式(5)限定在回路中不出现子回路。

2 求解 TSP 蚂蚁算法

蚂蚁算法是根据蚂蚁寻找食物的群体行为,由意大利学者 Colomi 和 Dorigo 等^[7]于 1991 年提出的。蚂蚁在运动过程中会释放一种特殊的分泌物——信息素,以此来寻找路径,当一只蚂蚁碰到一个还没有走过的路口时,会随机挑选一条路径前进,同时释放出与路径长度有关的信息素。蚂蚁走的路径越长,则释放的信息素量越小。后来的蚂蚁再碰到这个路口时,会以较大概率选择信息素浓度较高的路径前进,这个便形成了一个正反馈,最优路径上的信息素浓度越来越大,而其它路径上的信息素浓度却会随着时间的流逝而衰减,最终整个蚁群会找到最优路径。

2.1 基本蚂蚁算法

求解 N 个城市的 TSP 的基本蚂蚁算法的实现步骤如下,其中 C 为所有城市的集合:

(1) 参数初始化。令时间 $t = 0$ 和循环次数 $N_c = 0$,设置最大循环次数 $N_{c_{\max}}$,设置信息素总量 Q ,将 M 只蚂蚁分别置于 N 个城市上,令有向图上每条边 (i, j) 的初始化信息素 $\tau_{ij}(t) = \text{const}$ (常数),且初始时刻 $\Delta\tau_{ij}(0) = 0$,置所有蚂蚁的禁忌表 $\text{tabu}_k (k = 1, 2, \dots, M)$ 为空。

(2) 循环次数 $N_c \leftarrow N_c + 1$ 。

(3) 顶点计数器 $s = 0$ 。

(4) $s \leftarrow s + 1$ 。

(5) 蚂蚁的禁忌表索引号 $k = 0$ 。

(6) 蚂蚁数目 $k \leftarrow k + 1$ 。

(7) 蚂蚁 k 根据状态转移概率公式计算的概率选择城市 j 并转移到 $j, j \in C - \text{tabu}_k$ 。

(8) 修改禁忌表指针,即 $\text{tabu}_k \leftarrow \text{tabu}_k \cup \{j\}$ 。

(9) 若 $k < M$,则转(6),否则转(10)。

(10) 若 $s < N$,转(4),否则转(11)。

(11) 更新每条路径上的信息素浓度。

(12) 若满足结束条件,即 $N_c \geq N_{c_{\max}}$,则循环结束并输出程序计算结果,否则清空禁忌表并转(2)。

2.2 基本蚂蚁算法的缺陷

在 2.1 中给出了求解 TSP 的基本蚂蚁算法,基本蚂蚁算法主要存在以下缺陷:

(1) 当一只蚂蚁在一个城市选择下一个转移目标时,需要考查所有该蚂蚁未走过的城市。但实际上 TSP 的最优路径上,通常从一个城市只会转移到与其距离较近的城市。

(2) 在所有蚂蚁完成一次遍历后,更新路径上的信息素时,通常采用以下三种方式:① 蚂蚁所经过路径的每条边上的信息素加上信息素总量 Q/L_k (路径长度);② 蚂蚁所经过路径上的每条边上的信息素加上 Q/d_{ij} (边的长度);③ 每条边上信息素均加上 Q 。其中:① 只考虑了路径长度,没考虑到每条边在路径中的贡献大小,实质上,较小的边一般对求得最优路径具有较大的贡献。② 考虑了每条边的贡献,却没有考虑路径长度。③ 既没有考虑每条边的贡献也没有考虑路径长度。

(3) 由于每只蚂蚁从一个城市向下一个城市转移时,均按概率选择下一个城市,蚂蚁算法收敛度较慢,而易于陷入局部最优。

(4) 信息素更新时,考虑所有蚂蚁走过的路径,但实质上,有些蚂蚁走过的路径与最优路径相距甚远。

2.3 混合蚂蚁算法

针对基本蚂蚁算法的缺陷,笔者对基本蚂蚁算法作了改进,给出了一个高效的混合蚂蚁算法。对基本蚂蚁算法的改进主要通过以下几种方法:(1)初始化时,每条边上的信息素量的设置;(2)蚂蚁的转移策略设计;(3)信息素更新方法;(4)引入局部优化策略,优化蚂蚁走过的路径。

2.3.1 预处理

为了限制一个蚂蚁从一个城市转移到下一个城市只在一定数量的较近的城市中选择,定义以下两个数据结构:

集合 $\text{ptabu}_i (i = 1, 2, \dots, N)$:存放距城市 i 最近的前 K 个城市编号;

数组 $\text{Sumtabu}[N]$: $\text{Sumtabu}[i]$ 存放距城市 i 到与其最近的前 K 个城市的距离之和;

通过对每个城市 i 的邻接城市按距离排序,选择前 K 个城市置于 ptabu_i 中,然后根据 ptabu_i 计算 $\text{Sumtabu}[i]$ 。

2.3.2 初始边上的信息素量的设置

通常情况下,蚂蚁算法在初始化时,对每条边赋予等量的信息素,以使蚂蚁在初始时可以完全随机地选择路径。为了使蚂蚁在开始时就有目的地选择较近的路径,本算法在初始化时,就对每个顶点的邻接点按照边的长度设置不等量的信息素,即:

$$\tau_{ij}(0) = d_{ij} * Q / \text{Sumtabu}[i] * K / N \quad (6)$$

2.3.3 蚂蚁的转移策略

当蚂蚁 k 由城市 i 向另一个城市 j 转移时,基本蚂蚁算法考查所有不在禁忌表 tabu_k 中的城市,增加了算法的执行时间并延缓了算法的收敛速度。本算法中,蚂蚁 k 在城市 i 选择下一个城市时,只有满足下面条件的

城市 j 才会被考虑: $j \in \text{ptabu}_i$ 且 $j \in C - \text{tabu}_k$, 即只在距城市 i 最近的 K 个城市中选择。若对所有 $j \in \text{ptabu}_i$ 均有 $j \in \text{tabu}_k$, 算法将无从作出选择, 为此, 随机选择一个 $j \in C - \text{tabu}_k$ 的城市 j , 并将蚂蚁转移到该城市。 K 的大小会影响到算法的性能, K 值小则算法运行时间短, 但有可能求不出最优解; 而 K 值大时, 又会使运行时间较长。因此, 只有选择适当的 K 值, 才能使算法既具有较好的精度, 又能在一定程度上缩短运行时间。转移策略如下:

若 $\text{ptabu}_i \cup C - \text{tabu}_k$ 非空, 则在 t 时刻蚂蚁 k 由城市 i 转移到城市 j 的状态转移概率为:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{allowed}_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, \text{ if } j \in \text{allowed}_k \quad (7)$$

其中: $\text{allowed}_k = \text{ptabu}_i \cup C - \text{tabu}_k$; $\tau_{ij}(t)$ 为 t 时刻边 (i, j) 上的信息素; $\eta_{ij}(t) (= P/d_{ij}, P$ 为一常量) 为启发函数; α 为信息启发式因子, 表示轨迹的相对重要性, 即蚂蚁在运行过程中所积累的信息素量在蚂蚁运动时所起的作用; β 为期望启发式因子, 表示能见度的相对重要性, 反映了蚂蚁在运行过程中启发信息在选择路径中的受重视程度。

若 $\text{ptabu}_i \cup C - \text{tabu}_k$ 为空, 则随机选择一个城市 $j (j \in C - \text{tabu}_k)$ 作为转移目标。

2.3.4 信息素的更新

基本蚂蚁算法采用的信息素更新方法, 或者只考虑路径长度而不考虑每条边对减小路径长度所做的贡献, 或者不考虑路径长度, 另外, 对所有蚂蚁所走过的路径均更新信息素, 这些更新方法均不利于信息素向最优路径上集中。本算法中, 只对部分较短路径更新信息素。更新方法为: 对所有蚂蚁走过的路径按长度排序, 选择前 F 个较短路径更新路径上的信息素, 置 $\Delta\tau_{ij}(t) = Q/(S_k * d_{ij})$ (S_k 为第 k 只蚂蚁所找到的路径长度), $\tau_{ij}(t + N) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t)$ 。另外, 信息素的过分集中会使算法易于收敛于局部最优, 为此, 限制每条边上的信息素量于 $[\text{DOWN}, \text{UP}]$, 在更新信息素时, 若边上的信息素小于 DOWN , 则置为 DOWN , 若大于 UP , 则置为 UP 。

2.3.5 局部优化策略

局部优化分为两个部分: (1) 在每只蚂蚁找到可行解 R 之后, 使用 $2 - \text{opt}$ 方法优化路径, 即在该路径上随便选择两个位置 k_1 和 $k_2 (|k_2 - k_1| > 1)$, 令 $t_1 = R_k, t_2 = R_{(k+1)\%N}, t_3 = R_{k_1}, t_4 = R_{(k_1+1)\%N}$ ($\%$ 为取模运算), 若 $d_{t_1 t_3} + d_{t_2 t_4} < d_{t_1 t_2} + d_{t_3 t_4}$, 则将 $R_{(k+1)\%N}$ 和 R_{k_1} 之间的路径反向。(2) 算法运行结束后, 在算法所找

到的最好路径中, 如果该路径不是最优路径, 其中的部分路径也是和最优路径相同的, 而只是有部分路径和最优路径不同, 这时反复使用 $2 - \text{opt}$ 优化方法对最好路径进行优化, 直到不能优化为止。这样, 可以很好地提高算法性能。

3 数据实验

笔者运用 TSP 的标准测试数据 eil51 (已知该问题的最优解为 426) 对算法进行了测试, 并与基本蚂蚁算法作了性能比较。初始参数设置为 $\alpha = 1, \beta = 5, P = 40, K = 8, F = M/2, Q = 1000, [\text{DOWN}, \text{UP}] = [0.001, 10]$, 最大运行代数为 2000。当蚂蚁数 $M = 15$ 时, 对 ρ 从 0.1 到 0.9, 算法各运行 100 次, 运行结果如表 1 所示。

表 1 改进蚂蚁算法的运行结果

次数 ρ	解	426	427	428	429	平均值
0.1		85	14	1	0	426.16
0.2		81	15	4	0	426.23
0.3		72	24	2	2	426.34
0.4		74	20	6	0	426.32
0.5		74	25	1	0	426.27
0.6		71	27	1	1	426.32
0.7		74	21	2	3	426.34
0.8		65	25	6	4	426.49
0.9		46	34	11	9	426.84
		642	205	34	19	426.37

从表 1 可以看出, 改进的蚂蚁算法在不同的 ρ 值下共进行的 900 次运行中, 求最优解的次数为 642 次, 占总运行次数的 71.3%, 最差解仅比最优解多 0.7%, 平均解仅比最优解多 0.087%。另外从表 1 可以看出, ρ 值在 0.1 ~ 0.9 之间变化, 对求得最优解的次数有一定的影响, 但对最差解和平均解的影响并不明显。

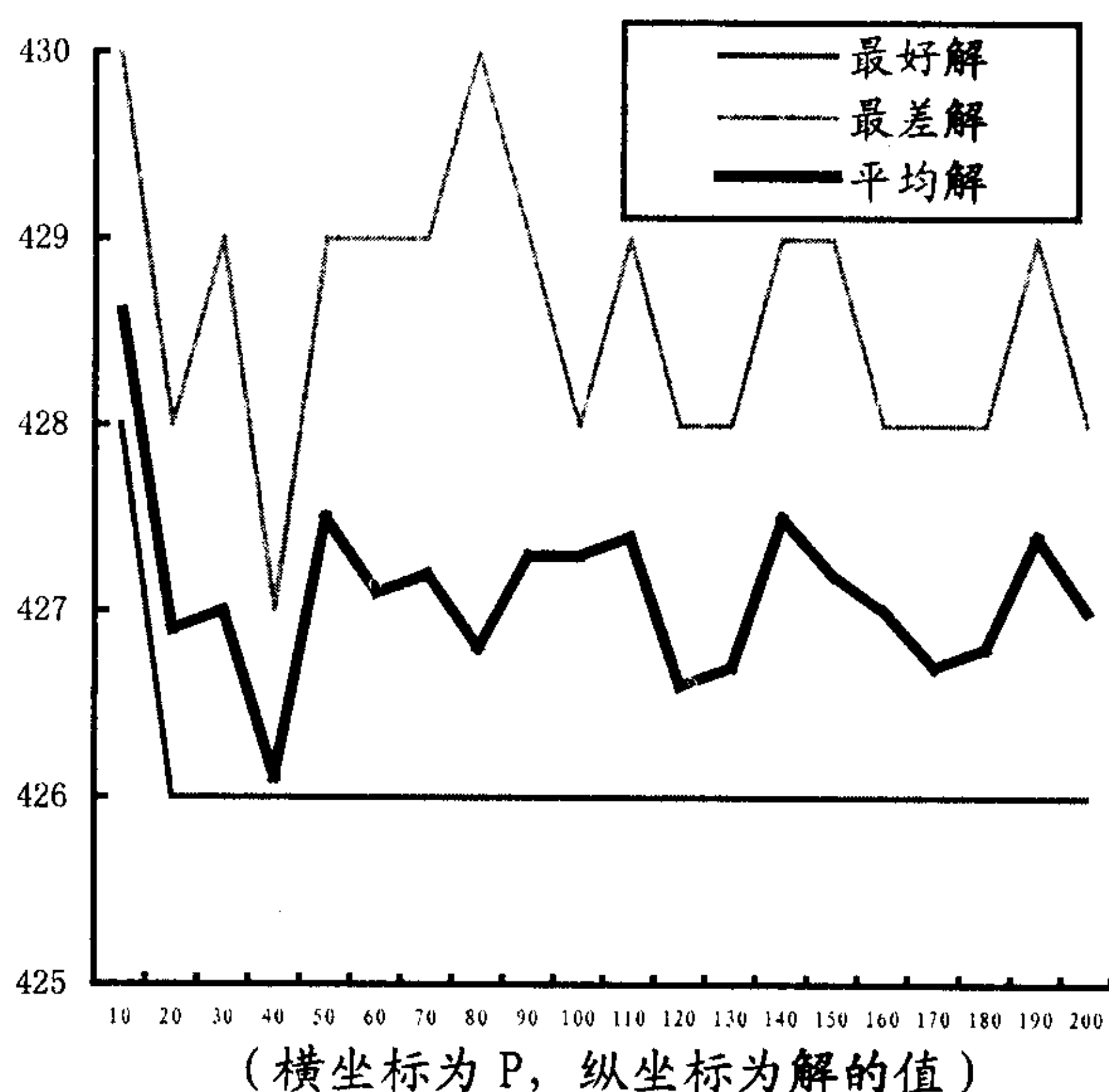
在初始化时对每条边赋予相同的信息素量 0.1, $\rho = 0.5$ 时, 对该测试数据, 基本蚂蚁算法的运行结果如表 2 所示。

表 2 基本蚂蚁算法的测试结果

最好解	偏差	最差解	偏差	平均解	偏差
444	4.2%	466	9.4%	456.3	7.1%

从表 1 和表 2 的对比来看, 文中的算法大大地提高了蚂蚁算法的求解性能。

$\eta_{ij}(t)$ 值无论是对基本蚂蚁算法还是对文中的算法的性能都有明显的影响, 其值的大小明显影响算法的求解精度。文中算法中 $\eta_{ij}(t) = P/d_{ij}$, 在 $\rho = 0.5$, 其它参数相同的情况下, 对 P 从 10 到 200, 以 10 为间隔, 对文中的算法各进行 10 次运行, 运行结果如图 1 所示 (图中, 横坐标为 P , 纵坐标为解的值)。

图 1 η 对算法求解精度的影响

4 结束语

蚂蚁算法是一个有效地解决组合优化问题的通用算法框架。文中讨论了基本蚂蚁算法的缺陷,给出了一个求解旅行商问题的混合蚂蚁算法。算法将蚂蚁的转移目标限制在距一个城市较近的 K 个城市中,设计了初始信息素的设置方案和信息素的更新方法,并引入了局部优化策略优化蚂蚁所找到的路径。数据实验

表明算法是有效的。

参考文献:

- [1] Carpaneto G, Toth P. Some New Branching and Bounding Criteria for the Asymmetric Traveling Salesman Problem[J]. Management Science, 1980, 26: 736 - 743.
- [2] Dantzig G B. Solution of a Large Scale Traveling Salesman Problem[J]. Operations Research, 1954, 2: 393 - 410.
- [3] Bellman R. Dynamic Programming Treatment of the Traveling Salesman Problem[J]. J ACM, 1962, 9: 61 - 63.
- [4] Grefenstette J J, Gopal R, Rosmaita B, et al. Genetic Algorithms for Traveling Salesman Problem[C]//In: Proceedings of an International Conference on Genetic Algorithms and Their Applications. Pittsburgh, USA: Pittsburgh PA Carnegie - Mellon University, 1985: 160 - 168.
- [5] Gutin G, Yeo A. Polynomial approximation algorithms for the TSP and the QAP with a factorial domination number[J]. Discrete Applied Mathematics, 2002, 119: 107 - 116.
- [6] Tsai Cheng - Fa, Tsai Chun - Wei, Tseng Ching - Chang. A new hybrid heuristic approach for solving large traveling salesman problem[J]. Information Sciences, 2004, 166: 67 - 81.
- [7] Colomni A, Dorigo M. Distributed optimization by ant colonies [C]//Proceedings of the 1st European Conference on Artificial Life. Paris, France: Elsevier, 1991: 134 - 142.

(上接第 109 页)

按照随机网络生成算法和分布式拓扑生成算法两种类型的节点分布形态,对网络的组网性能参数进行评估。在战场环境下,实时数据业务占据相对较大比重,所以采用端到端时延作为网络性能评估的一个重要参数。

由仿真结果从图 3 可以看出,与均匀分布模型相比,两种分布模型的时延较小,同时对于分布式网络,由节点之间呈现出“集群”动态性,端到端时延比随机网络模型的大。DSR 路由协议增加了路由缓存机制(Route Cache),所以在集群度较高的网络中,路由“缓存”命中率高。

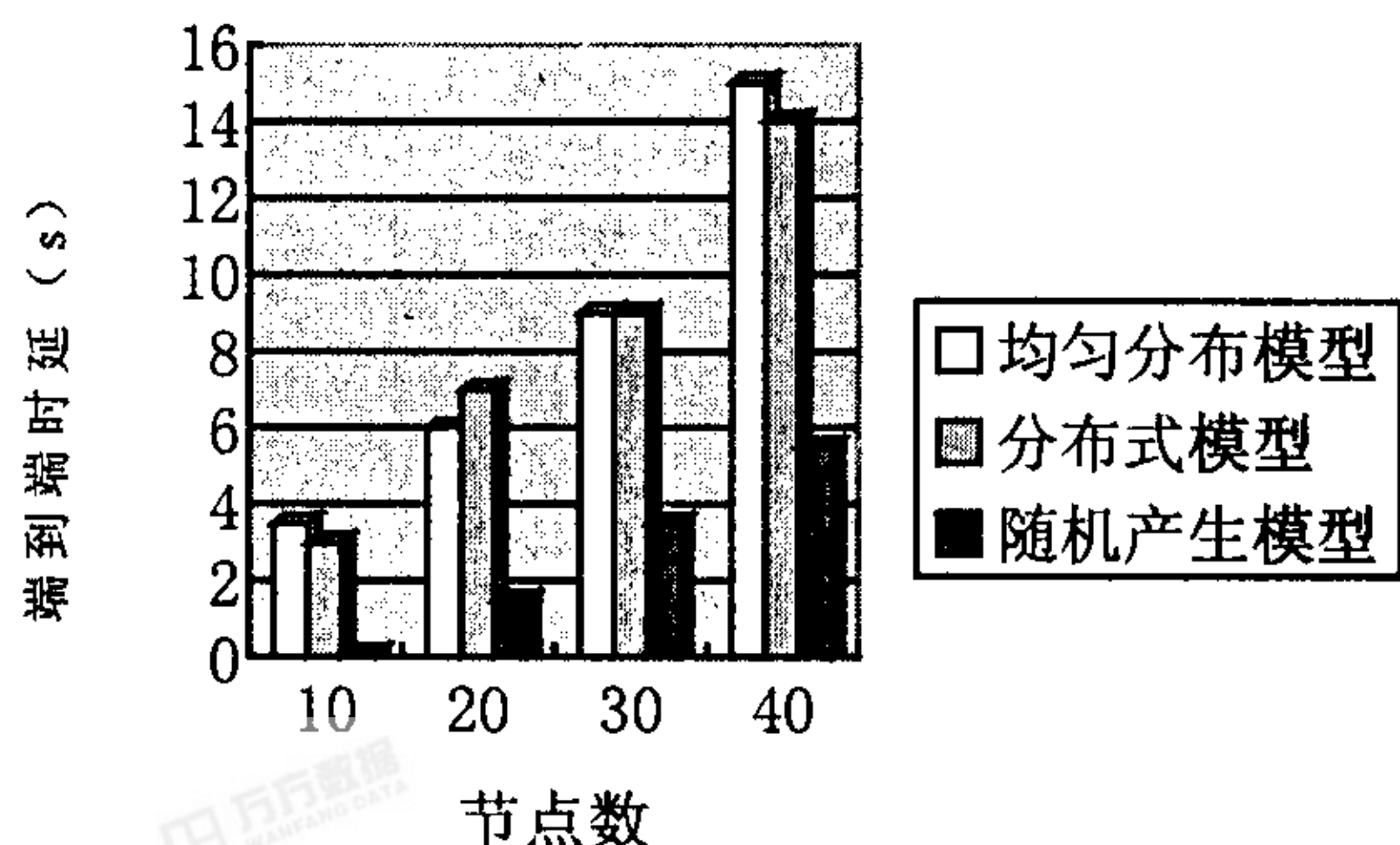


图 3 端到端时延结果比较

战术互连网络拓扑模型的研究与仿真,为战术互

联网仿真平台的设计打下基础。战术互联网性能仿真必须建立在实际网络环境的基础上。文中的两种网络拓扑模型能够按照战术互联网应用实际,生成符合网络特征的拓扑结构。战术互联网组网性能的评估与拓扑形态关系密切。在下一步的工作中,重点是在拓扑仿真模型的基础上,构建战术互联网移动模型,根据网络特性对 MANET 提出的通用路由协议进行改进,并建立完善的组网性能评估体系。

参考文献:

- [1] 竺南直,朱德成. 指挥自动化系统工程[M]. 北京:电子工业出版社,2003.
- [2] Waxman B M. Routing of multiple connections[J]. IEEE Trans on Communication, 1988, 6(9): 1671 - 1622.
- [3] 梁 荣. 基于遗传算法的 QoS 组播路由算法[D]. 上海:华东师范大学,2004: 53 - 54.
- [4] Salama H F, Reeves D S, Viniotis Y. Evaluation of multicast routing algorithms for real - time communication on high - speed networks[J]. IEEE Journal on Selected Areas in Communication, 1997, 15(3): 332 - 345.
- [5] 杨盘龙. 分层分布式 Ad Hoc 网络拓扑模型的研究[J]. 系统仿真学报, 2005, 17(6): 1405 - 1407.