一种基于蚁群算法的TSP问题分段求解算法

吴斌 史忠植[1]

(中科院计算技术研究所智能信息处理开放实验室 北京 100080)

摘要： 群居性昆虫行为的研究为计算机科学家提供了设计分布式控制和优化算法的有力方法。对以蚁群算法为代表的群集智能研究已经逐渐成为一个研究热点。本文首先在蚁群算法的基础上提出了相遇算法，提高了蚁群算法蚂蚁一次周游的质量，然后将相遇算法与采用并行策略的分段算法相结合，提出一种基于蚁群算法的 TSP 问题分段求解算法。实验结果表明该算法有较好的有效性。

关键字： 蚁群算法 组合优化 旅行商问题 并行策略 群集智能

AN ANT COLONY ALGORITHM BASED PARTITION ALGORITHM FOR TSP

Wu bin Shi zhongzhi

The key lab. of Intelligence Information Processing,Institute of Computing Technology ,CAS

Abstract As a novel computational approach ,swarm intelligence systems and ant algorithms have become a hot research domain. This paper proposes a meeting algorithm and a partition algorithm for TSP based on typical ant algorithms. The meeting algorithm improves the ant touring quality, it provides good initial touring results for local optimization on the condition of low iteration times. Combining with a simple parallelization strategy –the partition algorithm, this paper get some good experiment results on Traveling Salesman Problems .

Key words ant colony system, combinational optimization , TSP, parallelization strategy, swarm intelligence

1 引言

研究群居性昆虫行为特性的科学家发现，昆虫在群落一级上的合作基本上是自组织的，在许多场合中尽管这些合作可能很简单，但是它们却可以解决复杂的问题。这种由群居性生物产生出来的一种集体行为，即产生的群集智能引起了包括计算机科学家在内众多研究人员的兴趣。蚁群算法就是利用群集智能解决组合优化问题的典型例子。蚁群算法（ant colony algorithm）是由意大利学者 M. Dorigo, V Maniezzo 等人在 90 年代初首先提出来的。它是继模拟退火算法、遗传算法、禁忌搜索（Tabu Search）算法、人工神经网络算法等元启发式搜索算法以后的又一种应用于组合优化问题的启发式搜索算法[1,2]。Marco Dorigo 等人将蚁群算法先后应用于 TSP 问题、资源二次分配问题（Quadratic Assignment problem）等经典优化问题，得到了较好的效果。蚁群算法在动态环境下也表现出高度的灵活性和健壮性，如其在电信路由控制方面的应用被认为是目前较好的算法之一[3]。此外，蚁群任务分工、打扫蚁巢分类蚁卵等行为也启发了相应的协作和聚类算法[3]。目前，国内也有一些学者对蚁群算法及其在电信领域的应用开展了研究[6,7]。

蚁群算法虽然成功应用于多个经典组合优化问题，但是它也存在一些可以改进的地方。其中一点是计算时间较长。针对这一点，本文从两个方面对算法进行了修改。首先提高蚂蚁

[1] 吴斌，博士生，主要研究方向人工智能、智能信息处理；史忠植，博士生导师，主要研究方向人工智能。

每一次周游的质量，其次将减少循环次数与并行策略相结合。为了提高蚂蚁周游质量、本文在文献[3]采用 2-opt 局部优化的基础上，在一次周游中采用两只蚂蚁分头寻找，途中碰头合并路径信息的方法。实验表明这一方法可以在一定程度上改善蚂蚁一次周游的质量。关于第二个方面，本文采用了加速和效率都很高[8]的区域分裂并行策略，同样从实验结果可看出，在减少循环次数以后，分段的并行策略也能达到较好的效果。

本文选用了通用的 TSPLIB* (ftp:ivr.uni-heidelberg.de/pub/tsplib)中的两个实例（ST70 和 KroB150）和文献[5]中提到的中国 CHC144 城市实例共三个实例。其中 ST70 实例实验得到的最短路径为 677.1096，TSPLIB 提供的最佳路径运算结果为 678.59（但提供的最短路径值为 675）；KroB150 实例实验得到的最短路径为 26127.35，TSPLIB 提供的最佳路径值为 26130；CHC144 实例实验得到的最短路径值与文献[5]中通过并行遗传算法得到的结果相同，都为 30354.3。路径图见实验结果部分。

本文在蚁群算法的基础上，将相遇算法与采用并行策略的分段算法相结合，提出了一种基于蚁群算法的 TSP 问题分段求解算法。在循环次数较小的条件下，获得了较好的实验结果。在一定程度上改善了算法的有效性，并提高了算法效率。文章组织如下：首先介绍基本的蚁群算法（AS）[2]和改进型蚁群算法（MMAS）[4]，然后分别说明相遇算法和采用区域分裂策略的分段算法，最后给出实验结果和分析结论。

2　蚁群算法 AS 和 MMAS

用于寻找最短路径的蚁群算法来源于蚂蚁寻食的行为。蚁群寻找食物时会派出一些蚂蚁分头在四周游荡，如果一只蚂蚁找到食物，它就返回巢中通知同伴并沿途留下“信息素”(外激素 pheromone)作为蚁群前往食物所在地的标记。信息素会逐渐挥发，如果两只蚂蚁同时找到同一食物，又采取不同路线回到巢中，那么比较绕弯的一条路上信息素的气味会比较淡，蚁群将倾向于沿另一条更近的路线前往食物所在地。蚁群算法设计虚拟的“蚂蚁”，让它们摸索不同路线，并留下会随时间逐渐消失的虚拟“信息素”。根据“信息素较浓的路线更近”的原则，即可选择出最佳路线。

蚁群算法首先成功应用于 TSP 问题（旅行商问题）。下面简单介绍其基本算法。已知一组城市 n，TSP 问题可表述为寻找一条访问每一个城市且仅访问一次的最短长度闭环路径。设 d_{ij} 为城市 i 到 j 之间路径长度；在欧式 TSP 中，d_{ij} 为 i 到 j 之间的欧式距离（即 $d_{ij}=[(x_i-x_j)^2+(y_i-yj)^2]^{1/2}$）。TSP 的实例是已知一个图 G（N，E），N 是一组城市，E 是一组城市间的边（在欧式 TSP 中的一个全连通图）。本文选用的实例是对称 TSP 问题，即 $d_{ij}=d_{ji}$。

基本的蚁群算法 AS 可以简单表述如下：在 0 时刻进行初始化过程，蚂蚁放置在不同的城市，每一条边都有一个初始外激素强度值 $\tau_{ij}(0)$。每一只蚂蚁的禁忌表的第一个元素置为它的开始城市。然后，每一个蚂蚁从城市 i 移动到城市 j，依据两个期望措施的概率函数选择移动城市（包括参数 α 和 β，见公式（4））。在 n 次循环后，所有蚂蚁都完成了一次周游，它们的禁忌表将满；在这时，计算每一个蚂蚁 k 的 L_k，$\Delta\tau_{ij}^k$ 依据公式（3）更新。而且，由蚂蚁找到的最短路径（即 $\min_k L_k, k=1,\ldots,m$）将被保存，所有禁忌表将置空。这一过程重复直到周游计数器达到最大（用户定义）周游数 NC_{MAX}，或者所有蚂蚁都走同一路线。后一种情况被称为停滞状态。如果算法在 NC 次循环后结束，蚂蚁算法的复杂度为

$o(NC \cdot n^2 \cdot m)$

信息素更新公式：

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \qquad (1)$$

其中 ρ 是一个参数，（1 - ρ）表示在时刻 t 和 t+n 之间外激素的蒸发，

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad (2)$$

$\Delta\tau^{k}_{ij}$ 是单位长度上在时刻 t 和 t+n 之间第 K 个蚂蚁在边 e(i,j)留下的外激素的数量；

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{如果在时刻}t\text{和}t+n\text{之间第}k\text{个蚂蚁使用边}e(i,j) \\ 0 & \text{其它} \end{cases} \qquad (3)$$

Q 是一个常数，L_k 是第 K 个蚂蚁周游的路程长度。

第 k 个蚂蚁从城市 i 到城市 j 的跃迁概率为：

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum\limits_{k \in allowed} [[\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}]^{\beta}} & j \in allowed_k \\ 0 & otherwise \end{cases} \qquad (4)$$

其中 $allowed_k$={N-$tabu_k$}，α 和 β 都是控制外激素与可见度的相对重要性的参数。跃迁概率是可见度和 t 时刻外激素强度的权衡。

文献[4]提到的 MMAS 算法在 AS 算法的基础上作了三点改进：首先初始 $\tau_{ij}(t) = c$ 设为最大值 MAXTAO；其次一圈中只有路径最短的蚂蚁才修改 $\tau_{ij}(t)$；最后将 $\tau_{ij}(t)$ 限定在 MAXTAO 和 MINTAO 之间。从实验结果可知，MMAS 算法在防止算法过早停滞以及有效性方面对 AS 算法有较大改进。另外，文献[2]、[4]中在蚁群算法的基础上，还采用了 2-opt 局部优化方法提高算法效率，

3 相遇算法

相遇算法的基本思路是将一只蚂蚁的一次周游分由两只蚂蚁分头进行，在路径中间碰头合成一次周游路径。此外，由于在一次循环中，蚂蚁进行多次周游，最短的一次周游影响路径信息素，因此，在一次周游中，选择一个城市后，即计算当前路径长度，如果长度超过本次循环已得到的最小值即结束此次周游，这样可以进一步减短计算时间。

相遇算法 MMMAS 表示为：

1、初始化参数：α，β，ρ，Ncmax, tabu[k], taomax, taomin, m;

2、NC 加 1;

3、一组蚂蚁开始相遇循环 k++;

4、两只蚂蚁选择同一起点城市 tabu[k][0]；

5、第一只蚂蚁以公式（4）计算的跃迁概率选择城市 j1：j1∈{0,1…n-1}-tabu[k]；

6、第二只蚂蚁以公式（4）计算的跃迁概率选择城市 j2：j2∈{0,1…n-1}-tabu[k]-j1；

7、当前路径长度大于本次 m 组相遇循环已知最短路径，结束此次两只蚂蚁周游；

8、禁忌表未满转至第 5 步；

9、2-opt 局部优化路径；（可选）

10、 如果 k<m 转至第 3 步；

11、 计算本次 m 组蚂蚁相遇循环的最短路径，置空禁忌表；

12、 用公式（1）（3）更新路径信息数（最短路径增加，其它衰减）；

13、 NC 小于 Ncmax 且未进入停滞状态，转至第 2 步；

14、 输出最优解。

从算法描述可以看出，一次周游的两只蚂蚁共用一个禁忌表，这样保证两只蚂蚁不会选择重复的城市。与仅用一只蚂蚁的算法 MMAS 相比，算法的复杂度没有改变，区别在于 MMMAS 第一个城市有两次选择下一城市的机会，而 MMAS 只有一次选择机会。实验结果也表明，算法 MMMAS 的结果较 MMAS 有一定程度的提高。

4 分段算法

由于蚁群算法在算法运算前期收敛较快，而且城市数对所有 TSP 算法包括蚁群算法的性能有较大的影响，采用分段算法的目的是吸收算法前期快速收敛的特点，在得到一定质量的结果后，减少城市数，分段优化，提高算法运行效率。

分段算法 Section_MMMAS 具体描述如下：

```
Section_MMMAS()
{ //参数初始化：
    Ccycle=n/8;  //循环因子
  Ncmax=50;
  Section_length=n/2;//分段长度；
    counter=0;
 //循环分段开始；
 while(counter<=n/Ccycle)
  {
    start=counter*Ccycle;  //分段起点
   MMMAS(start, Section_length);
    if(分段结果优于原结果) 更换路径；
   }
  输出最优结果；
}
```

算法中 MMMAS（start,Section_length）由上一部分描述的 MMMAS（）稍作修改即可。算法中循环因子 Ccycle 与分段长度 section_length 决定算法的时间复杂度，设分段因子 CSL=n/section_length，算法复杂度为 $o(NC\cdot(n/CSL)^2\cdot m\cdot n/Ccycle)$，相比 MMAS 的复杂度可以看出，虽然算法 Section_MMMAS 的复杂度增加了一个 n 的乘积，但是由于一次

周游城市数减小，所以其中不仅 NC 可较大地减小，同时 m 值也可相应减小，而且合理选取 CSL 和 Ccycle 的数值也可控制复杂度过大。

5 实验结果

(一) 实验一：相遇算法与基本蚁群算法的比较

实例 1 ST70 $\alpha=1,\beta=5,\rho=0.5$,Ncmax=50, m=n=70

MMAS	681.9	695.1	681.22	699.6	688.13	697.35	682.13
	685.5	699.8	692.7	690.4	701.4	693.1	702.2
MMMAS	677.1096	688.2	678.6	677.1096	677.19	677.1096	685.9
	687.7	677.1096	686.4	694.5	678.6	688.6	678.5

表 1 ST70 实例 MMAS+2opt 与 MMMAS+2opt 算法结果

已知最优值为 677.1096, 可以看出 MMAS 在连续 14 次运算中，最好为 681.2,最差为 702.2，偏差 $\sqrt{\frac{\sum_{n}(X_t-Best)^2}{n}}$ 为 16.7，平均值为 692.18；而 MMMAS 最好为 677.1096,最差为 694.5,偏差为 7.7，平均值为 682.33。

实验一说明相遇算法在循环次数较小的条件下，在最好值、最差值偏差、偏差和平均值四方面都优于 MMAS，同时由于相遇算法没有从根本上改变 MMAS 算法，所以从数值差值也可看出相遇算法仅在一定程度上改善了蚁群算法 MMAS 的性能。

(二) 实验二：证明分段算法有效性

实例 1 ST70

表 3 ST70 实例分段算法结果

$\alpha=1,\beta=5,\rho=0.5$,Ncmax=50, m=n=70 Ccycle=n/8,CSL=2.

MMMAS 初始值	684.24	677.19	677.19	677.1096	685.82	677.1096	678.62
Section_MMMAS	677.1096	677.1096	677.1096	677.1096	685.82	677.1096	677.1096
MMMAS 初始值	690.28	678.42	684.24	688.21	677.19	677.1096	681.21
Section_MMMAS	677.19	677.1096	677.1096	685.31	677.1096	677.1096	677.1096
MMMAS 初始值	684.24	690.95	684.17	685.91	677.1096	678.62	
Section_MMMAS	677.1096	683.06	678.12	685.82	677.1096	677.1096	

已知最优值为 677.1096，可以看出 MMMAS 在连续 20 次运算中，最好为 677.1096,最差为 690.95，偏差为 6.68，平均值为 681.74；而 Section_MMMAS 最好为 677.1096,最差为 685.82、偏差为 3.57 ，平均值为 678.74。经过分段优化，20 个值中有 14 个值到达最优值，14 个值 的初始值都不大于 684.24，结果说明分段优化的效果明显，也可看出初始值直接影响分段效果。

实例 2 CHC144 分段算法所得最优路径图 result=30354.3（并行遗传算法得到的最优值也为 30354.3）。实例 3 kroB150 分段算法所得最优路径图 result=26127(TSPLIB 提供最佳值为 26130)。

实验二表明分段算法得到了三个实例的最优值，其中实例 1 和 3 超过已知最优值，实例 2 与并行遗传算法得到的最优值相同。这些证明分段算法不仅能进行局部优化，而且在搜索最优解上是有效的。同时也说明蚁群算法在解决组合优化问题方面较同类算法有一定的优势。

图 1 CHC144 最优路径图

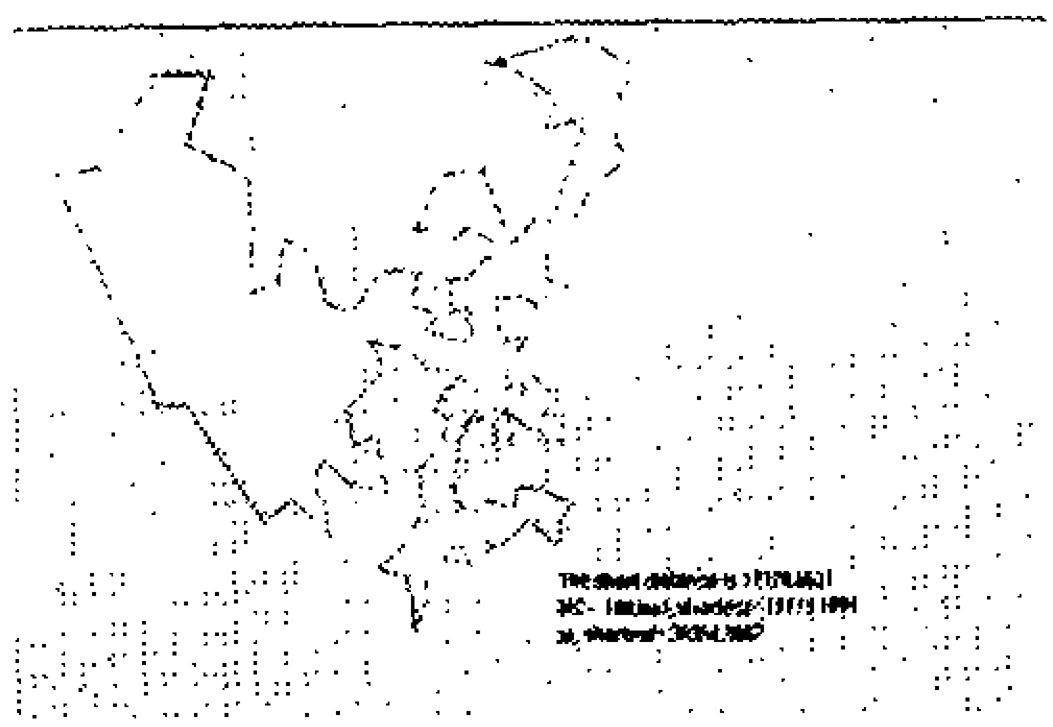
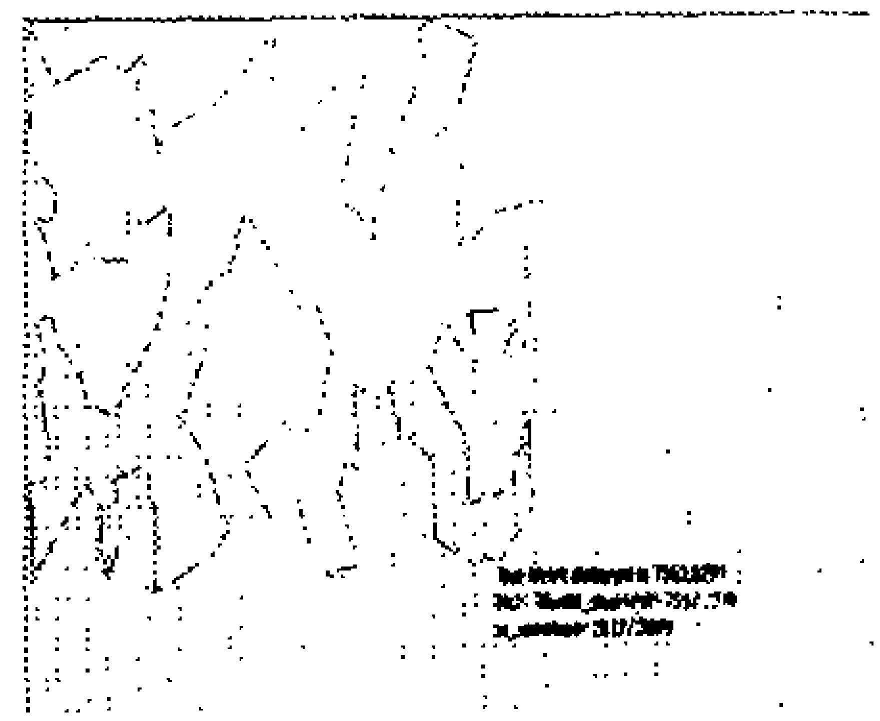


图 2 kroB150 最优路径图



6 结论

实验说明相遇算法在一定程度上改善了一次周游的质量，为分段算法在较小循环次数的条件下，提供了较好的初始值。而由分段算法得到三个实例的最优值可看出基于蚁群算法的分段优化算法在有效性方面可以到达甚至超过同类组合优化算法的水平。同时也说明群集智能在解决组合优化问题方面不仅是可行的，而且还具有一定程度的优势。但是，蚁群算法研究的时间并不长，多数成果都只是基于大量实验的数据分析，还没有坚实数学基础，其中各种参数的选取也比较复杂，所以从算法的理论方面还有许多需要解决的问题。此外，蚁群算法在其它组合优化问题上的应用以及群集智能在其它研究领域的应用也都是值得进一步研究的问题。

参考文献

1. A.Colorni,M.Dorigo, Heuristics from nature for hard combinatorial optimization problems,International Transactions in Operational Research, 3,1,pag.1-21.
2. Dorigo M. & L.M. Gambardella (1997). **Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem.** *IEEE Transactions on Evolutionary Computation,* 1(1):53-66.
3. E.Bonabeau, M.Dorigo,G.Theraulaz, Inspiration for optimization from social insect behaviour, Nature,vol 406,6 July 2000.
4. T.Stutzl and H.H.Hoos. The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. In T.Baeck,Z.Minchalewicaz,and X.Yao, editors,Proceeding of the IEEE International Conference on Evolutionary Computation(ICEC'97), page 309-314,1997.
5. *Rui Jiang, K.Y. Szeto, Yupin Luo, Dongcheng Hu* A Path-Splitting Scheme Based Distributed Parallel Genetic Algorithm for Large Traveling Salesman Problems, Proceeding of WCC2000-IIP2000.
6. 张纪会，徐心和，具有变异特征的蚁群算法，计算机研究与发展，1999，(10)
7. 张素兵，石国英，刘泽民，周正，基于蚂蚁算法的 QoS 路由调度算法，电路与系统学报，2000，(3)
8. 康立山，谢云，尤矢勇，罗祖华， 非数值并行算法 第一册 模拟退火算法， 科学出版社，1997；