蚁群算法的理论及其应用

姜长元

(南京师范大学数学与计算机科学学院，江苏 南京 210009)

摘 要：本文介绍了一种崭新的求解复杂优化问题的启发式算法—蚁群算法。该方法通过模拟蚁群搜索食物的过程，达到求解此类问题的目的。它具有智能搜索、全局优化、稳健性强、分布式计算、易与其它方法结合等优点。该算法用于解决组合优化问题，如 TSP，QAP，JSP 等效果较好。

关键词：蚁群算法；模拟进化算法；组合优化；旅行商问题

1 引言

研究群居性昆虫行为的科学家发现，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，但它们却可以解决许多复杂的问题。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，ACA）是由意大利学者 M.Dorigo，V.Maniezzo，A.Colorni 等人在 20 世纪 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。

蚁群算法不仅能够智能搜索、全局优化，而且具有稳健性（鲁棒性）、正反馈、分布式计算、易与其它算法结合等特点。利用正反馈原理，可以加快进化过程；分布式计算使该算法易于并行实现，个体之间不断进行信息交流和传递，有利于找到较好的解，不容易陷入局部最优；该算法易与多种启发式算法结合，可改善算法的性能；由于鲁棒性强，故在基本蚁群算法模型的基础上进行修改，便可用于其它问题。因此，蚁群算法的问世为诸多领域解决复杂优化问题提供了有力的工具。

M.Dorigo 等人将蚁群算法先后应用于旅行商问题（TSP），资源二次分配问题（Quadratic Assignment Problem，QAP）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。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。

2 蚁群算法的基本原理

蚁群系统本来是生物学家为更好揭示昆虫的交互作用而提出的一种昆虫自组织模式。尽管建立这种模式的初衷是为了帮助人们去理解这类昆虫的复杂行为，蚂蚁也不可能从这些解释中获益，但是数学及计算机方面的专家和工程师却把这种超越生物本身的模型转化成了一项有用的优化和控制算法—蚁群算法，也称蚁群系统(Ant Colony System，ACS)。

蚁群优化(Ant Colony Optimization，ACO)是该系统的核心内容，其原理可大致描述如下：蚂蚁属于群居昆虫，个体行为极其简单，而群体行为却相当复杂。相互协作的一群蚂蚁很容易找到从蚁巢到食物源的最短路径，而单个蚂蚁则不能。此外，蚂蚁还能够适应环境的变化，例如在蚁群的运动路线上突然出现障碍物时，它们能够很快地重新找到最优路径。人们通过大量的研究发现，蚂蚁个体之间是通过在其所经过的路上留下一种可称之为“信息素”(pheromone)的物质来进行信息传递的。随后的蚂蚁遇到信息素时，不仅能检测出该物质的存在以及量的多少，而且可根据信息素的浓度来指导自己对前进方向的选择。同时，该物质随着时间的推移会逐渐挥发掉，于是路径的长短及该路径上通过的蚂蚁的多少就对残余信息素的强度产生影响，反过来信息素的强弱又指导着其它蚂蚁的行动方向。因此，某一路径上走过的蚂蚁越多，则后来者选择该路径的概率就越大。这就构成了蚂蚁群体行为表现出的一种信息正反馈现象。蚂蚁个体之间就是通过这种信息交流达到最快捷搜索到食物源的目的。图 1 能更具体地说明蚁群系统的原理。

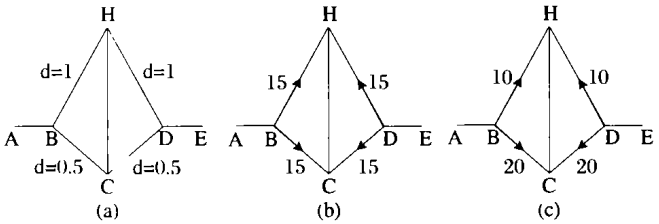

图 1 蚁群优化系统示意图

图中设 A 是蚁巢，E 是食物源，H、C 为障碍物，距离为 d。由于障碍物的存在，由 A 外出觅食或由 E 返回巢穴的蚂蚁只能经由 H 或 C 到达目的地。假设蚂蚁以“1 单位长度 / 单位时间”的速度往返于 A 和 E，每经过一个单位时间各有 30 只蚂蚁离开 A 和 E 到达 B 和 D(图 1a)。初始时，各有 30 只蚂蚁在 B 和 D 点遇到障碍物，开始选择路径。由于此时路径上无信息素，蚂蚁便以相同的概率随机地走两条路中的任意一条，因而 15 只选往 C，15 只选往 H(图 1b)。经过一个单位时间以后，路径 BCD 被 30 只蚂蚁爬过，而路径 BHD 上则只被 15 只蚂蚁爬过（因 BCD 距离为 1 而 BHD 距离为 2），BCD 上的信息量是 BHD 上信息量的两倍。此时，又有 30 只蚂蚁离开 B 和 D，于是各 20 只选择往 C 方向，而另外各 10 只则选往 H(图 1c)。这样，更多的信息素量被留在更短的路径 BCD 上。随着时间的推移和上述过程的重复，短路径上的信息量便以更快的速度增长，于是会有越来越多的蚂蚁选择这条短路径，以致最终完全选择这条短路径。

由上述可见，蚁群算法的核心有三条。第一，选择机制：信息素越多的路径，被选中的概率越大；第二，信息素更新机制：路径越短，迹增加越快；第三，协作机制：个体之间通过信息素进行交流。

3 蚁群算法的模型和流程

蚁群算法首先成功应用于 TSP 问题，下面简单介绍其基本算法。

3.1 TSP 问题的描述

给定 n 个城市的集合{0,1,2,…,n-1}及城市之间环游的花费 $C_{ij}(0\leq i\leq n-1,0\leq j\leq n-1,i\neq j)$。TSP 问题是指找到一条经过每个城市一次且回到起点的最小花费的环游。若将每个顶点看成是图上的节点，花费 C_{ij} 为连接顶点 V_i、V_j 边上的权，则 TSP 问题就是在一个具有 n 个节点的完全图上找到一条花费最小的 Hamilton 回路。

3.2 蚁群算法的描述

给定一个有 n 个城市的 TSP 问题，人工蚂蚁的数量为 m。每个人工蚂蚁的行为符合下列规律：根据路径上的激素浓度，以相应的概率来选取下一步路径；不再选取自己本次循环已经走过的路径为下一步路径。用一个数据结构(tabulist)来控制这一点；当完成了一次循环后，根据整个路径长度来释放相应浓度的信息素，并更新走过的路径上的信息素浓度。

现用 $\tau_{ij}(t)$表示在 t 时刻，边(i,j)上的信息素浓度。经过 n 个时刻，当蚂蚁完成了一次循环之后，相应边上的信息素浓度必须进行更新处理，模仿人类记忆的特点，对旧的信息进行削弱，同时，必须将最新的蚂蚁访问路径的信息加入到 τ_{ij}，得到：

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

其中 ρ 为一个取值范围在 0 到 1 之间的常数系数，表示残留信息的保留部分，(1-ρ)表示信息素的挥发程度，

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

其中$\Delta\tau_{ij}^{k}$是第 k 个蚂蚁在时间 t 到 t+n 之间，在边(i,j)上增加的信息素改变量。它的值由以下公式确定：

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k} & \text{若第 k 只蚂蚁在本次循环中经过 ij}\\ 0 & \text{否则}\end{cases}$$

其中 Q 是一个常量，用来表示蚂蚁完成一次完整的路径搜索后，所释放的信息素总量；L_k 是第 k 个蚂蚁的路径总花费，它等于第 k 个蚂蚁经过的各段路径上所需的花费 C_{ij} 的总和。如果蚂蚁的路径总花费越高，那么其在单位路径上所释放的信息素浓度就越低。很显然，蚂蚁不会在其没有经历过的路径上释放信息素。

$P_{ij}^{k}(t)$表示在 t 时刻蚂蚁 k 由位置 i 转移到位置 j 的概率：

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{k\in allowed_k}[[\tau_{ik}(t)]^{\alpha}\cdot[\eta_{ik}]^{\beta}]} & j\in allowed_k\\ 0\end{cases}$$

其中 $\eta_{ij}=1/C_{ij}$，C_{ij} 为经过路径(i,j)所需的花费。α 和 β 两个参数，分别用来控制信息素和路径长度的相对重要程度。$allowed_k$ 是第 k 个蚂蚁下一步可以选择的城市的集合。$allowed_k=\{0,1,2,\cdots,n-1\}-tabu_k$。与实际蚁群不同，$tabu_k$,(k=1,2,…,m)人工蚁群系统具有记忆功能，用以记录蚂蚁 k 当前所走过的城市，集合 $tabu_k$ 随着进化过程作动态调整。

3.3 蚁群算法的流程

在初始化的时候，m 个蚂蚁被放置在不同的城市上，赋予每条边上的信息素浓度为 $\tau_{ij}(0)=C$(C 为常数)，即各路径上和信息量相等。每个蚂蚁的 tabulist 的第一个元素赋值为它所在的城市。当蚂蚁们完成了一次完整的寻径过程后，计算 τ_{ij}^{k}，并且更新每条边上的信息素浓度。然后开始新一轮的循环。当循环的次数达到事先定义好的 NC_{MAX} 时或者所有的蚂蚁都选择了同一种路径方式时，整个程序终止。

Ant-cycle system 模型程序的伪代码如下：

```
Step 1:
  初始化:
  Set t=0,NC=0,每条边上的 τij(0)=0,并且△τij=0,放置 m 个蚂蚁
    到 n 个城市上
Step 2:
  令 s=1, (s 是 tabulist 的下标)
  For k=1 to m do
    把第 k 个蚂蚁的初始城市号码放置到 tabuk(s)中
Step 3:
  Repeat until tabulist is full
  Set  s=s+1
  For  k=1 to m do
    根据概率Pij^k来选择下一步应该到达的城市,将第 k 个蚂蚁移到
      城市 j,并将 j 插入到 tabuk(s)中
Step 4:
  For  k=1 to m do
    计算第 k 个蚂蚁的总路径长度 Lk,更新找到的最短路径
  For  k=1 to m do
    更新边上的信息素浓度
Step 5:
  对每一条边计算 τij(t+n)
  Set  t=t+n
  Set  NC=NC+1
  Set  △τij=0
Step 6:
  If(NC<NCMAX) and (不是所有的蚂蚁选择同一条路径)
  then 清空所有的 tabulist
    Goto Step 2
  打印出最短路径
  终止整个程序
```

如果程序终止于 NC 次循环后，这个算法的复杂度为 $O(NC\cdot n^2\cdot m)$。实际上，第一步的复杂度为 $O(n^2+m)$，第二步的复杂度为 O(m)，第三步和第四步的复杂度为 $O(n^2\cdot m)$，第五步的复杂度为 $O(n^2)$，第六步的复杂度为 $O(n\cdot m)$。实验证明 m 一般取

值与 n 为同一数量级。因此整个算法的复杂度为 $O(NC\cdot n^3)$。

M.Dorigo 曾给出 3 种不同模型，分别称之为 ant-cycle system, ant-quantity system, ant-density system。它们的差别在于表达式 $\Delta\tau_{ij}^{k}$ 的不同（即更新信息素的方式不同）。

在 ant-quantity system 模型中，

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{d_{ij}} & \text{若第 k 只蚂蚁在时刻 t 和 t+1 经过 ij}\\ 0 & \text{否则}\end{cases}$$

在此算法中，以 d_{ij} 表示城市 i 到城市 j 的距离，残留信息浓度为 Q/d_{ij}，即残留信息浓度会因为城市距离的减小而增大，也就是说，蚂蚁倾向于选择下一步较短的路径。

在 ant-density system 模型中，

$$\Delta\tau_{ij}^{k}=\begin{cases}Q & \text{若第 k 只蚂蚁在时刻 t 和 t+1 之间经过 ij}\\ 0 & \text{否则}\end{cases}$$

在此算法中，从城市 i 到 j 的蚂蚁在路径上残留的信息浓度为一个与路径无关的常量 Q。

后两种算法与前一种算法的区别在于：后两种算法中每走一步（即从时间 t 到 t+1），都要更新残留信息素的浓度，而非等到所有蚂蚁完成对所有城市的访问之后。

通过实验表明，在这三种算法中：ant cycle 算法的效果最好，这是因为它用的是全局信息 Q/L_k；而其余两种算法用的是局部信息 Q/d_{ij} 和 Q。这种更新方法很好地保证了残留信息不至于无限累积，如果路径没有被选中，那么上面的残留信息会随时间的推移而逐渐减弱，这使算法能“忘记”不好的路径，即使路径经常被访问也不至于因为 τ_{ij} 的累积，而产生 $\tau_{ij}>>\eta_{ij}$ 使期望值的作用无法体现。

在以上算法 Q，α，β，ρ 中的最佳组合可以由实验确定。

4 蚁群算法应用

蚁群算法在解决很多组合问题上都取得比较理想的效果。其中有两个比较著名的组合问题，QAP 问题和 JSP 问题 (Job-Shop Scheduling Problem)作相应调整的蚁群算法可以比较好地解决这两个组合问题。另外，将蚁群算法对实际问题的解决也取得一定的进展，如大规模集成电路中的综合布线以及电信网络中的路由等方面的应用。

4.1 QAP 问题

QAP 问题的目标函数可以用一个 n×n 的对称矩阵来描述。蚁群算法是基于它和 TSP 问题这方面的相似性来解决问题的。QAP 问题的目标函数矩阵 S 通过距离向量 D 和流向量 F 的组合组成，$S_{ih}=d_i\cdot f_h$。蚂蚁根据可见度信息 η_{ih} 来选择下一个节点，其中 $\eta_{ih}=1/S_{ih}$。矩阵 S 的元素值用作启发因子。

4.2 JSP 问题

JSP 问题可以用一个加权图描述。每条边的权值用参数对 $\{\tau_{kl},\eta_{kl}\}$ 表示。信息 τ_{kl} 和可见度 η_{kl} 是通过最长进程时间或者最短完成时间等要求决定。蚂蚁遍历节点的顺序就是相应的解决答案。在解决 10×10 和 10×15 的 JSP 问题中，蚂蚁算法的解与最优解的误差在 10%之内。这是一个相当不错的结果。

4.3 大规模集成电路综合布线

大规模集成电路中的综合布线可以采用蚁群算法的思想来进行。在布线过程中，各个引脚对蚂蚁的引力可根据引力函数来计算。各个线网 Agent 根据启发策略，象蚁群一样在开关盒网格上爬行，所经之处便布上一条金属线，历经一个线网的所有引脚之后，线网便布通了。给定一个开关盒布线问题，问题的计算量是固定不变的，主要由算法的迭代次数决定，而迭代次数由 Agents 的智能和开关盒问题本身的性质确定。蚁群算法本身的并行性，使之比较适合于解决布线问题。

4.4 电信网络路由

电信网络中的路由是通过路由表进行的。在每个节点的路由表中，对每个目的节点都列出了与该节点相连的节点，当有数据包到达时，通过查询路由表可知道下一个将要到达的节点。首先对路由表中的信息素强度进行初始化。在节点 x，以节点 i 为目的地址，邻节点为 J 处的信息素强度为 $\tau_{ih}=1/d_{ih}$，d_{ih} 为从 x 经节点 J 到节点 i 路径的最小费用值。然后周期性地释放蚂蚁来进行路由，并修改相应的信息素的值。仿真结果表明，无论呼叫是均匀分布还是集中分布，利用蚁群算法所得呼叫拒绝率和平均路径长度均小于最小负载法结果；在呼叫符合集中分布时，蚁群算法所得呼叫拒绝率低于最短路径法。

5 结论

目前，除了业界已得到公认的遗传算法、模拟退火法、禁忌搜索法、人工神经网络等热门进化类方法，新加入这个行列的蚁群算法开始崭露头角，为复杂困难的系统优化问题提供了新的具有竞争力的求解算法。蚁群算法思想在启发式方法范畴内已逐渐成为一个独立的分支，在有关国际会议上多次作为专题加以讨论。

这种由欧洲学者提出并加以改进的新颖系统思想，正在受到越来越多的人的注意和研究，应用范围也开始遍及许多领域。例如交通网络中的最佳路径选择问题，电信网络中的流量负载分配问题等等都可以应用该算法来解决。

但是，蚁群算法还不像其它的启发式算法那样已形成系统的分析方法和具有坚实的数学基础。参数的选择更多的是依靠实验和经验，没有定理来确定。而且它的计算时间偏长，国内外的有关研究仍停留在实验探索阶段，但从当前的应用效果来看，这种模仿自然生物的新型系统寻优思想无疑具有十分光明的前景，更多深入细致的工作还有待于进一步展开。

参考文献：

[1] 吴斌，史忠植．一种基于蚁群算法的 TSP 问题分段求解算法．计算机学报，2001.4(12):1~6

[2] 黎锁平，张秀媛，杨海波．人工蚁群算法理论及其在经典 TSP 问题中的实现．交通运输系统工程与信息，2002.2(1):54~57

[3] 温文波，杜维．蚁群算法概述．石油化工自动化，2002.(1):19~22

[4] 马良，项培军．蚂蚁算法在组合优化中的应用．管理科学学报，2001.4(2):32~36

[5] 吴庆洪，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．1999.36(10).