北京郵電大學

题目：量子蚁群算法在 TSP 的应用

学　　院：　　计算机学院

班　　级：　　2018211302

小　　组　　成　　员：

谯森　　学　号:2018210895

杨成栋　学　号:2018210913

熊宇　　学　号:2018210074

陈万鑫　学　号:2018210614

量子蚁群算法在 TSP 的应用

摘要 本文探究了量子蚁群算法的基本原理和在具体问题中的实际应用。基本原理:将传统的蚁群算法引入量子算法中的量子的态矢量和量子旋转门来分别表示和更新信息素，使算法具有更好的种群多样性和全局寻优能力。实际应用:将量子蚁群算法用于解决 TSP 问题(旅行推销员问题)中,并和传统的蚁群算法进行对比。

关键词 TSP 问题 蚁群算法 量子蚁群算法 量子计算

1 引言

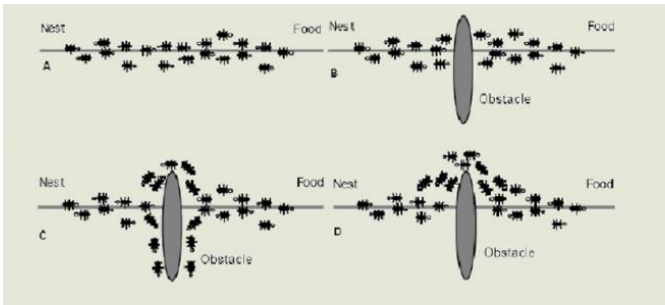
量子蚁群算法(quantum ant colony algorithm, QA-CA)是 20 世纪 90 年代后期新兴的一个研究领域,它以量子计算的一些概念和理论为基础,用量子位编码来表示信息素,通过量子门的旋转来完成信息素更新。与传统的蚁群算法相比，量子蚁群算法具有更好的种群多样性、更快的收敛速度和全局寻优能力。

本文介绍了蚁群算法的基本原理和蚁群算法改进为量子蚁群算法的方法,分别使用这两种算法解决 TSP 问题(旅行推销员问题),并且比较两种算法的效率与复杂度。

2 经典蚁群算法

蚁群算法(ant colony optimization，ACO)是一种模拟进化算法，它由 Marco Dorigo 于 1992 年在他的博士论文中提出，其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。ACO 本质上是一种启发式的全局优化算法，属于进化算法，具有分布计算、信息正反馈和启发式搜索的特征。

算法思想是模拟蚁群的信息素机制,蚁群的路径表示待优化的问题的可行解,蚁群的所有路径既是待优化问题的可行解空间。而根据信息素机制，路径较短的蚂蚁在一定时间内比路径较长的蚂蚁释放的信息素总量较多，而在信息素的影响下，会有更多的蚂蚁选择较短的路径。信息正反馈作用下，整个蚂蚁会在正反馈的作用下集中到最佳的路径上,此时对应的便是待优化问题的最优解。



图中可见蚁群一开始有两种路径（一条较长，一条较短）到达食物源，在一定时间内，较短的路径到达食物源的蚂蚁比较长的路径多，随着时间推移，较短路径积累的蚂蚁信息素逐渐增多,选择该路径的蚂蚁也越来越多，最后在信息正反馈的作用下，蚁群集中到最优即最短的路径上。

3 经典蚁群算法的量子化改进

3.1 量子编码改进

基本原理:量子比特(qubit)是一个充当信息存储

单元的物理介质的双态量子系统，是定义在二维复向量空间中的一个单位向量,该空间由一对特定的标准正交基{|0>，|1> }张成。因此，一个量子位不仅可以表示“0”或“1”两种状态，而且可以同时表示这两个状态之间的任意叠加态。即一个量子位可能处于|0>或|1>，或者处于两者之间的中间态，即|0>和|1>的不同叠加。因此一个量子位的状态可表示为:

|Ψ>=|α>+|β>,其中|α|^2,|β|^2 分别表示量子态处于“0”和“1”的概率。

ξ=arctan(β/α)，则第 i (i =1,2,.., m)个量子位的相位为ξi=arctan (βi/αi)。用符号 d 表示α和β的乘积，即 d=α*β。其中 d 的正负值代表此量子位的相位ξ在平面坐标中所处的象限。如果 d 的值为正,则表示ξ处于第一、三象限，否则处于第二、四象限。于是有 m 个量子位的个体 j 的概率幅可表示为:

$$pj=\left(\frac{\alpha 1}{\beta 1}\left|\frac{\alpha 2}{\beta 2}\right|\ldots\left|\frac{\alpha m}{\beta 2}\right.\right)$$

改进方法：在量子蚁群算法中，用量子比特来表示信息素，设种群大小为 n，其信息素用量子位表示为 P=(p1,p2⋯ pn),其中 pj(j=1,2,3⋯n)如下图所示。

$$pj=\left(\frac{\alpha 1}{\beta 1}\left|\frac{\alpha 2}{\beta 2}\right|\ldots\left|\frac{\alpha m}{\beta 2}\right.\right)$$

3.2 量子门改进

在常规量子算法中选用量子旋转门来进行更新操作。量子旋转门的调整操作如下:

$$\begin{pmatrix}\alpha\mathrm{i}\\ \beta\mathrm{i}\end{pmatrix}=\begin{pmatrix}\cos(\theta i) & -\sin(\theta i)\\ \sin(\theta i) & \cos(\theta i)\end{pmatrix}\begin{pmatrix}\alpha i\\ \beta i\end{pmatrix}$$

其中 i=1,2, ..m， $[\alpha\mathrm{i},\beta\mathrm{i}]^\wedge T$ 为第 i 个量子位的概率幅， $\theta\mathrm{i}$ 为第 i 个量子位的旋转角度,其大小和方向根据一个事先确定的调整策略而定。本文提出一种新的策略，其大小和方向不须要查表，而是通过一个公式自动调整量子旋转门的旋转角度和方向。公式如下:

$$\theta\mathrm{i}=\Delta\theta * f(\alpha i,\beta i)$$

其中 $\Delta\theta$ 是一个关键参数。如果选择比较大的 $\Delta\theta$,算法容易收敛到局部优点;反之, $\Delta\theta$ 较小又会使算法收敛很慢，甚至会处于停滞状态。

本文 $\Delta\theta$ 将定义为一个与迭代次数有关的变量， $\Delta\theta$ =5* exp(-t/tmax),其中 t 为迭代次数，tmax 是一个根据优化问题复杂性而定义的一个常数，为最大迭代次数。函数 $\mathrm{f}(\alpha i,\beta i)$ 的作用是使算法朝着较优解的方向搜索。

表1 函数 $f(\alpha_i, \beta_i)$ 的查询表

$d_{best}>0$	$d_{now}>0$	$f(\alpha_i, \beta_i)$	
		$\lvert\xi_{best}\rvert>\lvert\xi_{now}\rvert$	$\lvert\xi_{best}\rvert<\lvert\xi_{now}\rvert$
True	True	+1	-1
True	False	+1	+1
False	True	-1	-1
False	False	-1	+1

4 回顾 TSP 问题

旅行商问题（Travelling salesman problem, TSP）是这样一个问题：给定一系列城市和每对城市之间的距离，求解访问每一座城市一次并回到起始城市的最短回路，它是组合优化中的一个 NP 难问题。我们可以将其抽象理解为：对于一个图 G=(V,E)，每条边 e∈E 上有非负权值 W(e)，我们要去求解 G 的一个 Hamilton 圈 C，使得 C 的总权值 W(C)=ΣW(e)最小。

我们知道，随着图中结点的数量不断增加，解空间将呈现指数级增长，通过穷举法根本无法正确有效地求解，所以使用组合优化方法去求解 TSP 问题就显得十分必要。

目前求解 TSP 的优化算法有很多，比如模拟退火算法、神经网络、紧急搜索法、遗传算法等。这些算法在拥有自身优点的同时，也或多或少存在一些缺陷，比如求解结点规模仍有限制、求解时间过长等。为了解决上述问题，本文在使用蚁群算法求解 TSP 问题的基础上提出量子蚁群算法求解 TSP。

5 蚁群算法在 TSP 问题中的应用

5.1 经典蚁群算法在 TSP 问题中的应用

在旅行商问题中，设 m 为蚂蚁总数，$\tau_{ij}(t)$ 表示 t 时刻在边 ij 上的信息素的强度。随着时间的推移，先前的信息素会挥发，而新的信息素被填入，ρ 为旧的信息素保留程度，$1-\rho$ 为旧的信息素消逝程度。因此，当 m 只蚂蚁都完成一次循环后，各个边的信息素都需要发生一次调整，调整策略参照下面公式：

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

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

$\Delta\tau_{ij}^{k}$ 表示第 k 只蚂蚁留在路径 ij 上的信息素，为此次循环中边 ij 上的信息素增量。

$$\Delta\tau_{ij}^{k}=\begin{cases} Q/L_k,\ when\ ant\ k\ \ pass\ edge\ ij \\ 0,\ else \end{cases}$$

$$L_k=\sum_{i=1,j=2}^{m}d_{ij}$$

L_k 为本次循环后，第 k 只蚂蚁的路径长度，Q 为常数。

在循环过程中，由转移概率决定转移方向。这里我们用 ρ_{ij}^{k} 表示第 k 只蚂蚁从城市 i 转移到城市 j 的概率，那么：

$$\rho_{ij}^{k}=\begin{cases} (\tau_{ij}^{\alpha}\eta_{ij}^{\beta})/\sum\limits_{s\in allowed_k}\tau_{is}^{\alpha}\eta_{is}^{\beta},\ j\in allowed_k \\ 0,\ else \end{cases}$$

η_{ij} 为边(i,j)的能见程度，该能见程度作为蚁群算法的启发式信息。

经典蚁群算法求解 TSP 问题的一般步骤为：

Step1：将迭代次数赋初值为 0，τ_{ij} 赋初值为 C（C 为常数），一般设置为 0，将 m 只蚂蚁随机放在 n 个顶点上；

Step2：首先将各个蚂蚁的出发点放入解集中，然后根据前面给出的 ρ_{ij}^{k} 的计算公式来求解转移概率，并按照转移概率转移到结点 j，然后将结点 j 同样放入解集中；

Step3：当各个蚂蚁完成本次循环后，计算各个蚂蚁的路径长度 L_k，并按照信息素更新公式更新信息素；

Step4：迭代次数加一；

Step5：如果迭代次数小于总迭代次数，就转移到 Step2；

Step6：根据各个路径上的信息素强度，得出最优解，求解完毕。

5.2 量子蚁群算法在 TSP 中的应用

对任意的 TSP 采用量子蚁群算法的步骤：

① 初始化：设有 n 个个体的种群 P(t)=($p_1^t,p_2^t,\dots,p_n^t$),其中 p_i^t(i=1,2…,n)为种群第 t 次迭代的第 i 个个体,表示如下

$$p_i^t=\begin{bmatrix}\alpha_1^t & \alpha_2^t & \cdots & \alpha_m^t\\ \beta_1^t & \beta_2^t & \cdots & \beta_m^t\end{bmatrix}$$

其中 m 为量子比特的数目，算法开始的时候，所有的α_j^t、β_j^t(j=1,2…,m)都取$1/\sqrt{2}$,迭代次数 t 初始为 0。

②由 P(t)中的概率幅值的取值情况可以构造出 R(t)={ $r_1^t,r_2^t,\dots,r_n^t$},其中 r_i^t(i=1,2…,n)是一个长度为 m 的二进制串而其中的每一位由 p_i^t(i=1,2…,n)中的 $\left|\alpha_j^t\right|^2$、$\left|\beta_j^t\right|^2$(j=1,2…,m)的值所确定。设随机数 k(k ∈[0,1])，当$\left|\alpha_j^t\right|^2$>w 时，$r_j^t=0$，不然，$r_j^t=1$。(j=1,2…,m)。

③把这 n 个蚂蚁随机置于这 k 个城市的其中一个上，作为起始点。

④每只蚂蚁重复地借助状态转移原则（即伪随机概率转移），由式子(6) 建立一条路径，并且在建立路径时蚂蚁每状态转移一次，便释放信息素，即使得边越短的则认为这条边属于最优路径的可能性则越大，就是让相应的信息素量子比特概率幅越倾向于|1>，直到所有蚂蚁都完成了路径的建立。

$$\beta_i(a,b)=$$

$$\begin{cases} \frac{d_{(a,b)}-E_a}{L_a} * 0.5+0.5 & (a \neq b) \wedge (L_a \neq 0) \\ 0 & a=b \\ 0.5 & (a \neq b) \wedge (L_a = 0) \end{cases}$$

$$\alpha_i(a,b) = \sqrt{1-(\beta_i(a,b))^2}$$

$$E_a = \frac{1}{k}\sum_{j=1}^{k} d_{(a,j)}$$

$$L_a = \max\{\left|d_{(a,j)} - E_a\right|, j=1,2,\ldots,k\}$$

其中$d_{(a,b)}$表示城市 a 与 b 之间的距离

状态转移规则为

$$b = \begin{cases} arg\max\limits_{b \in U}\{[\tau(a,b)].[\eta(a,b)]^{\gamma}\} & q \leq q_0 \\ arg\max\limits_{b \in U} p(a,b) & otherwise \end{cases}$$

上式确定了量子蚁群算法中蚂蚁 n 状态转移由城市 a 转移至 b，其中:

$\tau(a,b)$是边(a,b)上的信息素浓度,其值由 α 和 β 确定。

$\eta(a,b)$=1/ $d_{(a,b)}$,其作为边(a,b)的自启发量，而$d_{(a,b)}$是 a 和 b 两个城市的距离。

γ 是自启发量的权重。

定义 q(q∈[0,1])为均匀分布的随机变量，决定了选择下一个城市的概率。再定义q_0($q_0 \in [0,1]$)是一个给定的常数，q_0越大则蚂蚁选择下一个城市的概率也就越小，我们可以通过对参数q_0的调整从而去调节算法对于新路径的探索度，进而决定了算法集中搜索至今最优路径附近的路径还是其他路径。即蚂蚁选择当前的最优可能移动的方式概率为q_0，该最优移动方式通过信息素的积累和启发式的信息值求出。那么剩下的概率(1-q_0)决定了蚂蚁有偏向性地探索其他路径。

U 是从 a 点出发的可行集。

$$p(a,b) = \begin{cases} \frac{\tau(a,b).\eta(a,b)^{\gamma}}{\sum_{s \in U}\tau(a,b).\eta(a,b)^{\gamma}} & b \in U \\ 0 & otherwise \end{cases}$$

p(a,b)是蚂蚁 i(i=1,2,⋯,n)自城市 a 选择城市 b 的概率

⑤记录本次迭代产生的最优解。

⑥通过应用量子旋转门规则对刚走过的路径进行信息素量的更新

⑦判断是否满足算法的终止条件。如果满足了最大迭代次数 t_{max}=2000(此处设为 2000)又或者最优解停滞不再发生改变则终止。否则回到步骤③算法继续。

5.3 量子蚁群算法和经典蚁群算法在 TSP 上的对比

在城市数目和蚂蚁数目相同情况下，取 γ =2，$t_{max} = 2000, q_0 = 0.8$情况下对两种 TSP 测试结果比较如下

表 1 采用经典蚁群算法和量子蚁群算法对 UlysseS16 与 Chn144 问题的求解平均迭代次数比较

TSP	平均迭代次数	
	量子蚁群算法	经典蚁群算法
Ulysses16	29	338
Chn144	1095	1930

由此可以看出量子蚁群算法求解速度大大提高，所需迭代次数大大减少。

6 结论

量子算法是一种新兴的模拟进化算法，虽然研究的时间不长，缺乏坚实的理论基础，但它具有并行计算和强鲁棒性等优点，使它的应用前景十分广阔。本文算法在经典蚁群算法的基础上，加入量子算法的思想，完成了一种改进的算法——量子蚁群算法，大大地加速了蚁群算法的收敛速度。通过使用量子蚁群算法解决 TSP 问题，并与经典蚁群算法进行时间和迭代次数的比较，证明了改进后算法是可行并且有效的。

致谢 向对该文有帮助的各位人士表示谢意.

参考文献

1 李盼池,宋考平,杨二龙等.基于相位编码的量子蚁群算法[J].系统工程理论与实践,2011 ,31(8) :1565 - 1570.

2 李絮,刘争艳,谭拂晓等.求解 TSP 的新量子蚁群算法[J].计算机工程与应用,2011 ,47(32) :42 - 44 ,86.

3 赵俊生,李跃光,张远平等.一种改进的量子蚁群算法及其应用[J].计算机应用与软件, 2010,27(7):133 -135,216.

4 张广帅，张煜东，吉根林.蚁群算法求解 TSP 综述[J].南京师范大学学报:工程技术版, 2014, 14(4):39-44.

5 宋辉，戴葵,王志英.量子算法模拟系统现状[J].计算机科学，2000, 27 (9) : 1-3.

6 郭平,鄢文晋基于 TSP 问题的蚁群算法综述[J].计算机科学, 2007，34(10):181-184.