第 34 卷 第 5 期 计 算 机 工 程 2008 年 3 月
Vol.34 No.5 Computer Engineering March 2008
·博士论文· 文章编号：1000—3428(2008)05—0005—03 文献标识码：A 中图分类号：TP311.5

基于蚁群算法的 XML 概率查询策略与算法优化

刘 波[1]，杨路明[1]，雷刚跃[2]
(1. 中南大学信息学院，长沙 410083；2. 湖南信息职业技术学院，长沙 410200)

摘 要：针对 XML 数据半结构化的特点及概率查询理论，结合蚁群算法，提出添加杂交算子和更新信息素的方法，该方法不仅能动态选择数据查询方向，而且能避免无效查询，扩大数据查询范围，提高收缩效率。模拟测试证明了该方法能优化 XML 查询。
关键词：蚁群算法；概率查询；信息素；杂交算子

Strategy of XML Probabilistic Query and Optimization Algorithm Based on Antcolony Algorithm

LIU Bo[1], YANG Lu-ming[1], LEI Gang-yue[2]
(1. College of Information Science and Engineering, Central-south University, Changsha 410083;
2. Hunan College of Information, Changsha 410200)

【Abstract】 Based on XML semi-structured characteristic and probabilistic query theory and antcolony algorithm, this paper gives a method to research for XML query which adds crossover operator and renews pheromone. It can not only choose data-query direction dynamic, but also avoid useless query. As a result, query range is widened and shrinkage efficiency is enhanced. At last, it is made sure by corresponding simulant test that the method can optimize XML query.

【Key words】 antcolony algorithm; probabilistic query; pheromone; crossover operator

XML 通过语义规则描述一定数目的元素，有许多优点，如数据与显示分离、可在应用程序之间传递数据等。由于其半结构化的特征，相对一般数据库查询，XML 查询会涉及大量的结构连接操作，为了选择适当的结构连接，需要对结构连接的顺序、大小进行估计，因此 XML 查询成为国内外研究的热点，特别是利用概率查询更是一个研究新课题。

1 问题描述

设$N=\{N_1,N_2,\cdots,N_n\}$表示XML相关结点的集合，$E=\{E_1,E_2,\cdots,E_m\}$表示XML相关结点表的集合，$P=\{P_1,P_2,\cdots,P_m\}$表示每条边上相应概率的集合，如图 1 所示。本文将对如下问题进行讨论：(1)对于表中要查询的结点，如何经过根结点就能快速定位到要查找的结点；(2)对于多次查询的结点，如何设计其查询路径；(3)对于不断变化的XML树结构，如何维护其边值概率。

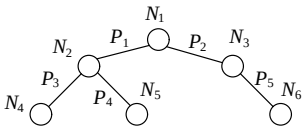

图 1 XML 树结构

2 相关工作

经过多年的发展，针对XML的特征提出了许多查询算法，如XPath查询最小化、X-RESTORE的XML视图查询等，特别是利用概率查询，有助于减小XML查询的搜索空间、循环比较次数，从而提高查询的效率。在文献[1]中，根据树的半结构化特点及DTD语法规则，利用条件概率对XML概率查询进行统计分析，并提出概率关系式$A=(N,\sum,R,\rho,P)$,其中，N表示数据状态集合；$\sum$表示标志集；R表示操作规则；ρ表示对应关系；P表示概率值，并给出了相应的计算公式。文献[2]提出了对XML数据进行动态处理的方法，并提出了相应处理模型，特别是利用DTD结构和图的操作定义条件概率查询、分析半结构化的XML处理过程对本文的启示较大。文献[3]针对XML查询的结构连接顺序操作提出了好的选择策略，特别是带剪枝的动态规划法，对本文利用蚁群算法进行概率查询的启示更大。文献[4]则针对XML树的特点，提出模糊树的概率查询与更新方法，对XML概率查询更新有很好的借鉴作用。当然利用蚁群算法[5-6]能够很好地解决路径查询问题，它同样也适合XML数据查询。

蚁群算法[5]是近年来诞生的随机优化方法，主要是通过蚂蚁群体之间的信息传递而达到寻优的目的。其优点在于：(1)具有正反馈机制，通过信息素的不断更新达到查询最优路径；(2)是通用型随机优化方法，它不是对实际蚂蚁的一种简单模拟，而是更多地融入了人的智能；(3)是一种全局优化的方法，不仅可求解单目标优化问题，而且可求解多目标优化问题。因此本文结合XML概率问题，利用蚁群算法求精确解，期望能在优化性能和时间性能方面获得双赢。

3 概率查询与算法优化

3.1 形式化描述

始终假设存在结点字符集 N 和结点类型集$\sum$,并且约定：

基金项目：湖南省教育厅科研基金资助项目“遗传算法参数设计”(05c671)；中南大学大学生创新创业启航行动基金资助重点项目“非一致性数据库的一致性查询技术研究”(ZB018)
作者简介：刘 波(1969－)，男，博士研究生，主研方向：软件工程与数据库技术；杨路明，教授、博士生导师；雷刚跃，讲师、硕士
收稿日期：2007-03-15 **E-mail：**ltbo99@yahoo.com.cn

如果没有特殊声明，出现的 $a,b,c,d \in \Sigma$，并且它们一般不相等，r, s, u, v，$\upsilon_i \in N$，Σ ρ 表示对应关系集合，p 表示相应边上的查询概率。

定义 1 给定树模式查询Q，记Q的最小化树为$Q_{\min}$，$Q_{\min}$满足$Q_{\min} \equiv Q$，且不存在Q'，满足$Q' \equiv Q$，使$|Q'|<|Q_{\min}|$，其中$|Q|$表示Q的大小或规模，即树中结点数目。

因为大部分 XML 文档树的结构都有一定的特点，与关系数据类似，所以 XML 数据也受到一些完整性约束的限制。

定义 2 令$t=(Q_t, \sum t, R_t, \rho_t, P_t)$为一棵树，$a, b \in \Sigma$，如果每个类型为$a$的结点都有一个类型为$b$的孩子结点，则称树$t$满足孩子约束，记为$a \to b$；若结点$a$都有一个类型为$b$的后裔结点，则称树$t$满足遗传约束，记为$a \Rightarrow b$；若$a,c \in v$，$B \subseteq \sum$，若结点$a$有$B$里面的每个结点类型的孩子结点时，则$c$也是$a$的孩子，称树$t$满足兄弟约束，记为$a:B \Downarrow c$。

为了考虑完整性约束IC，定义Q的最小树$Q_{\min}$为：满足$Q_{\min} \equiv_{IC} Q$，且不存在Q'，满足$Q' \equiv_{IC} Q$，使$|Q'|<|Q_{\min}|$，其中$\equiv_{IC}$表示满足约束IC条件下的等价。

定义 3 给一个查询模式$Q=(V_Q, E_Q, P_Q)$，一个状态树$S=(V_S, E_S, P_S)$，满足如下条件：

(1)$V_Q = \{\upsilon \mid \upsilon \in N_S\}$;

(2) $\forall N_S, N'_S \in V_S \Rightarrow N_S \cap N'_S = \phi$;

(3) $\bigcup_{N_S \in V_S} N_S = V_Q$;

(4)$E_S \subseteq E_Q$;

(5) $\forall N_S \in V_S, \forall \mu, \upsilon \in N_S \Rightarrow (\mu, \upsilon) \notin E_S$。

其中，$P_k : E_k \to [0,1]$; $P_k(i,j) = P(j|k,i)$, $(i,j) \in E_k$, $k=1,2,\cdots, m$; $\sum_{(i,j) \in E_k} P_k(i,j)=1$; $(i,j) \in E_k$, $k=1,2,\cdots, m$。

定义 4 基本蚁群算法是通过模拟生物界中的蚂蚁搜索食物的过程，即通过人工蚂蚁之间的信息交流与相互协作最终找到从蚁穴到食物源的最短路径。

3.2 概率查询与算法优化

3.2.1 算法描述

(1)根据蚁群算法，结合 XML 概率查询的特点，进行算法形式化描述。

设有n个数据结点集$C=(1,2,\cdots,n)$，任意 2 个相连边$E(i,j)$之间的概率为p_{ij}，距离为d_{ij}，求一条至多经过每个结点仅一次、且能找到目标结点的路径 $\pi=(\pi(1), \pi(2), \cdots, \pi(n))$，使得 $\sum_{i=1}^{n-1} p_{\pi(i)\pi(i+1)}$ 最大，而比较的次数最小，即 $\sum_{i=1}^{n-1} \mathrm{d}_{\pi(i)\pi(i+1)}$ 最小。其中，$b_i(t)$表示t时刻位于结点i的蚂蚁的个数；$m=\sum_{i=1}^{n} b_i(t)$ 为蚂蚁的总数；$\tau_{ij}(t)$表示t时刻边ij上的信息素量，$\tau_{ij}(0)=\tau_0$，(τ_0为常数)。随着时间的推移，新的信息素不断加进来，旧的信息素逐渐挥发。用 ρ 表示信息挥发系数，$\rho \in [0,1)$，则用 1-ρ 表示信息素的残留因子。当所有蚂蚁完成一次循环后，各条边上的信息素按下式调整：

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

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

其中，$\Delta\tau_{ij}$ 表示本次循环中路径上的信息素增量，初始时刻，$\Delta\tau_{ij}=0$；$\Delta\tau_{ij}^{k}$ 表示第 k 只蚂蚁在循环过程中释放在边 ij 上的信息素。

$$\Delta\tau_{ij}(t)=\begin{cases}\dfrac{Q}{L_k} & \text{若第}k\text{只蚂蚁在本次循环中经过边}ij\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

式中，Q为常数，表示信息素强度；P_k表示本次循环第k次蚂蚁所形成的边的权值。蚂蚁在循环时，向哪个结点转移由转移概率p_{ij}^{k}决定。

$$p_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\partial}\eta_{ij}^{\beta}}{\sum_{s \in allowed_k}\tau_{js}^{\alpha}\eta_{js}^{\beta}} & j \in allowed_k\\ 0 & \text{otherwise}\end{cases} \tag{4}$$

其中，$allowed_k=\{\text{C-tabu}_k\}$表示蚂蚁$k$当前能选择的结点集合；$tabu_k$为禁忌表，它记录蚂蚁$k$已路过的结点和不能走的结点，用来说明人工蚂蚁的记忆性；η_{ij}表示某种启发信息，在优化问题中，$\eta_{ij}=d_{ij}^{-1}$，α, β体现了信息素和启发信息对蚂蚁决策的影响程度。

XML概率查询中蚁群算法基本的运行过程是这样的：m只蚂蚁同时从某个结点出发，根据式(3)选择下一次经过的结点，已去过或不能经过的结点放入$tabu_k$中，一次循环完成后，由式(1)~式(3)更新每条边上的信息素，反复重复上述过程，直到终止条件成立为止。

3.2.2 信息素更新

用蚁群算法求解XML查询优化选择问题，如图 2 所示，图中椭圆表示任务T_i对应的备选资源集，实心圆表示备选路径对应结点，需要解决的问题是从各组备选资源集中选项出一个资源结点，使得所有选择路径付出的代价最小，得到的结果最优化，如果将每个资源结点看成是蚂蚁爬过的一个结点，则问题的求解过程就可以转化为如何按照$T_1, T_2, \cdots, T_n$的顺序选择一条路径，使得蚂蚁爬过这一条路径所走过的总概率最大，而比较的次数最小。

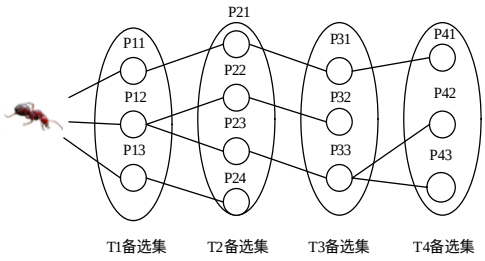

图 2 群算法的 XML 概率结点资源优化选择

为了使所求解的问题适合蚁群算法求解的需要，还需经过适当的转换，由于应用蚁群算法求解问题的关键是确定信息素的变化，因此采用以下 2 式更新信息素：

$$\Delta\tau_p^k = Q_1 / L_p^k \tag{5}$$

$$\Delta\tau_g^k = Q_2 / L_g^k \tag{6}$$

在式(5)中，$\Delta\tau_p^k$为局部信息素更新，表示蚂蚁k每爬过一段路程，每选择一个资源结点时，就在该段路程留下局部信息素；Q_1为常数，表示信息素的强度；L_p^k为该路程的长度。设i,j分别为该路程的起点与终点，用式(7)确定L_p^k：

$$L_p^k(i,j)=\begin{cases}C_j + trc_{ij} & (i,j) \in A\\ C_j & \text{otherwise}\end{cases} \tag{7}$$

其中，C_j为资源结点j的代价成本；trc_{ij}为资源结点i,j存在次序约束时的操作成本；A为任务次序约束有向图中的有向边的集合。式(6)中，$\Delta\tau_g^k$为全局信息素更新，表示信息素的强度；L_g^k为该路径的总长度(比较次数)；用$allowed_k(t)$表示蚂蚁k第t次允许爬行的资源结点；用数组$tabu_k$记录蚂蚁k已经爬

过的资源结点，如$allowed_k(0)=\{1,2,3\}$, $allowed_k(1)=\{4,5,6,7\}$, $tabu_k(0)=0$，则L_g^k可由以下公式计算：

$$L_g^k=\sum_{t=0}^{n-1}L_p(tabu_k(t),tabu_k(t+1)) \tag{8}$$

3.2.3 杂交算子

在运算过程中，蚁群的转移是由各条路径上留下的信息量的强度和资源结点的相关成本来引导的，蚁群运动的路径总是趋近于信息量最强的路径，因此信息量最强的路径与所需要的最优路径比较接近，然而信息量最强的路径不是所需要的最优路径的情况依然存在，这是由于在人工蚁群系统中，各路径上的初始信息量是相同的，蚁群创建的第 1 条路径所用到的信息就主要是结点之间的概率信息，这样蚁群在所经过的路径上留下的信息就不一定能反映出最优路径的方向，特别是蚁群中个数较少或所计算的路径的组合较多时，更不能保证蚁群创建的第 1 路径能引导蚁群走向全局最优路径。

为增强蚁群算法的寻优能力，特引入杂交算子，当新的解产生时，就对这些新的解进行杂交，以产生更好的解，从而对所求的解起增强作用，由于杂交算子能对现有解进行重组，因此很可能会发现更好的解，这也是遗传算法具有很好的搜索能力的原因。

当 m 只蚂蚁爬完了整个路径后，通过式(8)可以分别计算他们爬行路径的长度，在进行杂交算子之前需要选择父体蚂蚁，采用遗传算法中的轮盘赌的方式选择，第 k 只蚂蚁被选项中的概率为

$$P_k=L_g^k/\sum_{k=1}^{m}L_g^k \tag{9}$$

杂交的过程为先随机选择 2 个杂交点，然后以邻近正交的原则交换 2 个父体，这是由 XML 树型特征决定的，如以下 2 个父个体：

A=2　5　9

B=2　6　10　12

由于蚂蚁爬行路径上各备选资源集中资源结点编号各不相同，因此杂交只需直接交换就行了。

A=2　6　10　12

B=2　5　9

3.2.4 算法实现步骤

带杂交算子的蚁群算法求解 XML 概率选择问题的具体实现步骤如下：

(1)参数初始化，令迭代次数N_c=0，设置最大迭代次数N_{cmax}；对任务次序约束图的每条有向边用距离初始化，对结点的概率进行初始化；对图 2 有向边赋信息素初始值$\tau_{ij}(0)=\tau_0$，$\Delta\tau_{ij}(0)=0$。

(2)将 m 只蚂蚁放置在初始结点 0 上，把初始结点 0 放入禁忌表 $tabu$ 中，并设置 k=1 可访问结点集 $allowed$。

(3)t=0，蚂蚁 k 开始爬行：

1)蚂蚁k根据状态转移概率公式式(4)计算的概率选择下一个访问的资源结点j并前进，$j\in allowed_k(t)$, $t\leftarrow t$+1。

2)将蚂蚁移动到的新结点号加入$tabu_k$，并根据式(5)计算$\Delta\tau_p^k$，用式(10)执行局部更新。

$$\tau(tabu_k(t\text{-}1),tabu_k(t))=(1-\rho)\tau(tabu_k(t\text{-}1),tabu_k(t))+\Delta\tau_p^k \tag{10}$$

3)如果 t<n(n 为所需寻求的资源结点个数)，则转 1)，否则据根式(6)计算$\Delta\tau_g^k$，用式(11)执行全局更新，其中$i\in$[0,n-1]，i 取整数。

$$\tau(tabu_k(i),tabu_k(i+1))=(1-\rho)\tau(tabu_k(i),tabu_k(i+1))+\Delta\tau_g^k \tag{11}$$

4)如果 k<m，则 $k\leftarrow k$+1，转步骤(3)。

(4)根据式(9)计算每只蚂蚁被选择的概率，按照轮盘赌选择方式从蚁群中选择出 2 个父体，用随机生成 2 个杂交点按前述方法进行杂交。

(5)计算杂交生成后路径的路径长度L_1，L_2，如果$L_1<\min_{1\le k\le m}(L_g^k)$，则按式(11)对$L_1$或$L_2$对应路径执行信息素更新，否则不执行更新。

(6)$N_c\leftarrow N_c$+1，若$N_c<N_{cmax}$，则转步骤(2)，否则结束。

4 仿真测试

实验环境如下：win2000 专业SP4 版，CPU：C-M 1.73 GHz，内存 1 GB，硬盘 40 GB, 5 400 转，程序开发软件VS2005 C#。并以文献[3]中的带剪枝的动态规则作比较，在仿真计算时，各项参数的取值 α =1, β =0.5, Q_1=10, Q_2=12, ρ =0.2，分别查询 3 个相同的结点，按前面提出的带杂交算子的蚁群算法进行计算，运行结果如表 1 所示。

表 1　仿真结果比较

次数	剪枝	蚁群
m=5	10	10
m =10	12	12
m =50	65	51
m =100	126	87

由表 1 可见，随着数据量越大，迭代次数越多，这种概率查询收缩的速度就越快。

5 结束语

本文结合蚁群算法，提出了对 XML 进行概率查询的改进算法，通过正交算子求最优解和全局(或局部)次优解路径上的信息素强度以及相应操作的优化手段相结合改进 XPath 查询方法，从而有效地克服了树查询模式中重复比较的缺陷，使得 XPath 查询方法的性能有显著的提高。随着其他算法研究的不断引入，XML 概率查询算法将会获得更广泛的应用。

参考文献

[1] Carrasco R C, Rico-Juan J R. A Similarity Between Probabilistic Tree Languages: Application to XML Document Families[J]. Pattern Recognition, 2003, 36(9): 2197-2199.

[2] Cannataro M, Pugliese A. An XML-based Architecture for Adaptive Web Hypermedia Systems Using a Probabilistic User Model[C]// Proceedings of the IEEE IDEAS Conference. Washton D. C.: IEEE Computer Society, 2000: 257-265.

[3] Wu Yuqing, Patel J M, Jagadish H V. Structural Join Order Selection for XML Query Optimization[C]//Proceedings of the 19th IEEE ICDE International Conference on Data Engineer. Los Alamitos: IEEE Computer Society, 2003: 443-454.

[4] Abiteboul S, Senellart P. Querying and Updating Probabilistic Information in XML[C]//Proceedings of the 10th Int'l Conf. on Extending Database Technology. Munich: Springer-Verlag, 2006: 1059-1068.

[5] Colorni A, Dorigo M, Maniezzo V, et al. Ant System for Job-shop Scheduling[J]. Belgian Journal of Operations Research and Statistic Computing Science,1994, 34(1): 39-53.

[6] Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computing, 1997, 1(1): 53-56.