一种基于时延约束的最优路径求解算法

张 涛 柳重堪 张 军

(北京航空航天大学 电子信息工程学院, 北京 100083)

摘 要: 作为 QoS 路由和流量工程的关键技术之一,基于时延约束的最优路径问题一直没有得到有效的解决. 针对现有的算法很难得到最优解和计算复杂度过大等问题,提出了一种基于时延约束的最优路径求解(DCOP)算法,该算法通过减少算法的搜索空间来有效地降低算法的计算复杂度,可得到最优的无环解. 算法采用自适应参数设计,提高了对网络规模和复杂业务变化的适应性. 仿真表明该算法比同类算法计算复杂性降低了近一个数量级,且算法具有自适应能力,设计简单,易于工程实现.

关 键 词: 时延受限; 路由算法; 服务质量; 流量工程

中图分类号: TP 301.6

文献标识码: A **文 章 编 号**: 1001-5965(2006)02-0209-05

Algorithm for delay constraints optimal path calculation

Zhang Tao Liu Zhongkan Zhang Jun

(School of Electronics and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: As one of the most challenging problems in the QoS Routing and traffic engineering, the problem of delay constraints optimal path calculation has the non-polynomial(NP) complete complexity. The algorithm which is proposed in literature has some problem, such as not getting optimal path and being very hard complexity etc. So a delay constraints optimal path (DCOP) algorithm which can solve this problem well was proposed. By reduced the search region of the algorithm, the efficiency of algorithm can be improved effectively and an optimal loop-less path can be gotten. The self-adapting parameter design is adopted in this algorithm to improve the adaptability on the network scale and the changing of complicated service. Using extensive simulations on random graphs and random assigned link weights, the huge improvement in complexity of the new algorithm is tested. The test also indicate that the algorithm has more adaptability and more practicability.

Key words: delay constrained; routing algorithm; quality of service; traffic engineering

基于 QoS 约束的最优路径算法是实现多媒体通信服务质量和网络流量工程的关键技术之一. 其中基于时延约束的最优路径算法最具有现实意义,尤其是卫星网络这类节点间时延较大的网络.

时延受限最优路径属于单约束最优路径问题,一般情况下,该问题是一个 NP(Non-Polynomial)问题[1],为此人们设计了许多近似算法,如 C-N 算法[2]、T-K 算法[3]、BG 算法[1]等,这些算法都有较大的缺陷,首先算法一般不能得到最佳路径,只能得到次优解;其次算法的性能往往与具体参数密切相关,如有的求解算法需利用 k 路由算法,而 k 的选值将决定算法的效果,若 k 值较大,则难以保证算法的效率,反之,不能保证找到可行路径;另外,这些算法大多设计复杂,工程实现困难.

解决时延受限最优路径的另一个思路是扩展原有的单一量度最优算法(如 Dijkstra 算法),使其能针对多度量求解,如文献[4]给出的多标号算法和 EBFA 算法[5]等,这些算法可得到满足约束条件的最优解,且算法设计简单,易于实现,但

收稿日期: 2005-01-20

基金项目: 国家 863 资助项目(2003AA712022);国家自然科学基金资助项目(10377005)

作者简介: 张 涛(1973-),男,山东威海人,博士生,zhtao73@163.com.

其计算复杂度随网络节点数目的增加呈指数级增长,算法无法实用. 随后的许多学者对上述算法进行了改进,比较成功的有路径受限算法[6]和HMCOP算法[7],这些算法往往通过限制每个节点的最大路径数 X,以增大路由失败概率和降低路由性能为代价来降低运算复杂度,算法性能不够理想,同时,X 的选择决定算法的效率和性能,它的选择没有既定的标准,这将不利于算法的实现和推广.

本文的主要工作在于针对多标号算法和HMCOP算法的优缺点,提出可有效降低算法计算复杂度的改进方法,并给出相应的时延受限最优路径(DCOP)算法. 该算法具有以下优点:①算法可得到满足约束条件的最优解;②算法计算复杂度小,接近无约束的Dijkstra算法;③采用自适应动态调整技术,无需人为选择参数,算法适应性好.

1 问题描述以及符号说明

1.1 符号说明

$G(V,E)$:无向图,其中 $V=\{v_1,v_2,\cdots,v_n\}$ 是有限节点集,$E\subseteq V\times V$ 是有限边集;

$p(i,j)$:v_i、v_j 间的一条路,是路经过的节点集合;

$D(p(i,j))$:路 $p(i,j)$ 的时延值;

$C(p(i,j))$:路 $p(i,j)$ 的代价值,可表示费用或其他QoS参数;

$D_{\min}(s,t)$:s 到 t 的最短时延路的时延值;

$C_{\min}(s,t)$:s 到 t 的最小代价路的代价值;

$D_{\mathrm{cmin}}(s,t)$:s 到 t 的最小代价路的时延值;

$C_{\mathrm{dmin}}(s,t)$:s 到 t 的最短时延路的代价值;

D_r:路径的时延约束值;

Path _ Set:算法的数据结构,用于记录算法中的临时路;其操作包括Remove _ Min _ Path(移出Path _ Set中排序参数最小的路)和Inset _ Path(将一满足条件的路径按代价大小顺序插入Path _ Set中);

$V_p(y)$:算法的数据结构,用于记录已检查过的从源节点到 y 的有效路;其操作包括Inset _ Path _ Vp(将已检查过的路插入 $V_p(y)$)和Check _ Path _ Vp(检查某路 $p(s,y)$ 的时延和代价是否都大于或等于 $V_p(y)$ 中任一路的时延值和代价,若否,则返回FAIL).

1.2 问题描述

给定 $G(V,E)$,且对于任意边 $e(i,j)\in E$,存在非负函数 $D(e(i,j))$ 和 $C(e(i,j))$,则时延约束最优路径(DCOP)问题可以描述为:在图 G 中,寻找一条从源节点 s 到目的节点 t 的路径 $p(s,t)$,满足:$D(p(s,t))\leqslant D_r$,且 $\min(C(p(s,t)))$.

2 DCOP算法设计

2.1 算法设计思想

DCOP问题要求在当前网络状态下寻找满足时延要求的最优路径,该问题同时涉及时延和代价两个度量,使得Dijkstra算法无法直接使用,这是由于算法在搜索过程中每个节点需针对约束条件保留若干个可能的路径,导致问题的复杂度为NPC(NP Complete). 解决的思路是尽量减少算法需要搜索的可行路径,在这方面,多标号算法和HMCOP算法都采用仅保留满足约束条件的解来降低算法的搜索空间.

其中多标号算法的判断规则是:设当前路径为 p_{sx},则判断 p_{sx} 是否满足约束条件. 而HMCOP算法的判断规则为:设当前路径为 p_{sx},若经过路 p_{sx} 到目的节点 t 基于时延最小的路径为 p_{st},则判断 p_{st} 是否满足约束条件来决定是否保留当前路径. 相关理论分析如下.

命题1 对于图 G 中的任意节点 s、t 和 m,设路 $p(s,m)$ 的时延为 x,t 到 m 的最小时延为 y,若 $x+y>D_r$,则经过路 $p(s,m)$ 的所有路 $p(s,t)$ 都不满足条件 $D(p(s,t))\leqslant D_r$.

证明是显然的,略.

由命题1,HMCOP算法的约束条件为 $x+y>D_r$,多标号算法的约束条件为 $x+y>D_r+y$,所以HMCOP算法的可行解区间(见图1 *abgh*)大大小于多标号算法(见图1 *aijh*),其算法复杂度明显比多标号算法小.

由此可见,可行解区间的大小将直接影响算法的运行效率,例如HMCOP算法的效率就与 D_r 和 $D_{\min}$ 相对位置有关,当 D_r 逼近 $D_{\min}$ 时,可行解区间很小,算法效率很高,随着 D_r 与 $D_{\min}$ 差距拉大,算法效率下降很快(参见实验3). 所以可以通过进一步限制其大小来提高算法效率,根据这个思路,本文给出的DCOP算法作了下面两方面的

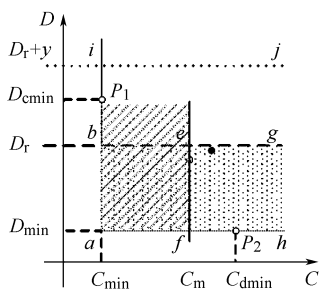


图1 三种算法可行解区间比较

改进.

1) 在解空间内找到路 P,使其 $D \leqslant D_r$,C 尽量小. 令 C 值为 C_m,并以 C_m 为约束条件来限制可行解区间的大小.

该改进的判断规则是:设当前路径为 p_{sx},若经过路 p_{sx} 到目的节点 t 基于代价 C 最小的路径为 p_{st},则判断 p_{st} 是否满足约束条件 C_m 来决定是否保留当前路径. 具体分析如下.

命题 2　对于图 G 中的任意节点 s、t 和 m,存在一条代价为 C_m 路 $P_m(s,t)$,设路 $p(s,m)$ 的代价为 x,t 到 m 的最小代价为 y,若 $x+y>C_m$,则经过路 $p(s,m)$ 的所有路 $p(s,t)$ 都不会比路 $P_m(s,t)$ 优.

证明是显然的,略.

由命题 1、命题 2 可以保证 DCOP 问题的全局最优值在区间 *abef* 内(如图 1 所示),这样,改进后的算法搜索空间将大大缩小,算法效率进一步提高. 这里 C_m 的获取将是算法实现的一个关键,对其具体的分析见 2.2 节.

2) 在算法运行过程中,采用动态调整 C_m 的方法,使可行解区间进一步优化. 这种动态调整使算法具有很好的自适应性,可以克服诸如 HMCOP 算法效率受 D_r 和 D_{min} 相对位置关系影响等缺点,算法适应性大大加强(参见第 3 节实验 3).

2.2　C_m 取值分析

由上节分析可知 C_m 大小将直接影响算法的效率,若令 C_0 为某 DCOP 问题的最优路径对应的代价值,则 C_m 值越接近 C_0,算法效率越高,所以寻找合适的 C_m 值将是一个关键问题,本算法在这方面采用首先取得一个可行值,再逐步动态调整其大小,以获得最优值的设计思路,具体是:

1) 对某一个 DCOP 问题,若问题有解,则一定有 $D_{min} \leqslant D_r$ 存在,所以可令 $C_m = C_{dmin}$;

2) 若 $D_{cmin} > D_r$,则可以通过修改最小代价路 p_1(见图 1)来获取比 C_{dmin} 更优的 C_m;

3) 在算法计算过程中,对每一个可行解 $p(s,y)$,计算其代价值与 $C_{dmin}(y,t)$ 之和,并动态调整 C_m 的值,使其逐渐向 C_0 收敛.

2.3　算法描述

DCOP 算法的主体模块包括以下三个步骤:①分别求 G 中任一节点 y 到目的节点 t 基于代价和时延的最优路;判断 $D_{min}(s,t)$ 是否大于约束条件,若大于则算法停止;判断 $D_{cmin}(s,t)$ 是否满足约束条件,若满足则算法停止;记录 $D_{min}(y,t)$、$C_{min}(y,t)$ 和 $C_{dmin}(y,t)$,以备主约束计算用. ②调用 C_m 求解模块,求得算法初始的代价限定条件 C_m. ③调用主约束计算模块,计算从源节点 s 到目的节点 t 的最优路径.

C_m 求解模块采用修改最小代价路的方法获得,模块从路的目的节点 t 开始,依次检查每一个节点. 具体伪码实现如下.

```
BEGIN
  C_m = C_dmin(s,t)
  y = t
  WHILE(y≠s) DO {
    y = NEXT_NOTE(p(s,t),y)
    IF D(p(s,y))+D_min(y,t) ≤ D_r
        AND C(p(s,y))+C_dmin(y,t) < C_m
        C_m = C(p(s,y))+C_dmin(y,t) }
END
```

主约束计算模块采用类似于 Dijkstra 算法的方法来进行求解. 该模块在计算中需维护一个临时路集合 Path_Set,将当前满足条件的路(可行解)加入该集合,具体实现如下:

设当前的最优路为 $p(s,x)$,对于 x 的邻接节点 y,则

第 1 步:检查 y 是否已在路 $p(s,x)$ 中,若在则表示路径 $p(s,y)$ 存在环路,该路无效;

第 2 步:计算 $p(s,y)$ 的时延值与 $D_{min}(y,t)$ 之和,若约束超限,则 $p(s,y)$ 不是可行解;

第 3 步:计算 $p(s,y)$ 的代价值与 $C_{min}(y,t)$ 之和,若约束超限,则 $p(s,y)$ 不是可行解;

第 4 步:动态调整 C_m,对于可行解 $p(s,y)$,计算其代价值与 $C_{dmin}(y,t)$ 之和,如小于 C_m,则替代之;

第 5 步:将 $p(s,y)$ 的时延和代价值与节点 y 已纪录的时延和代价值相比,若不都大于或等于,则将该路加入 Path_Set 中(Inset_Path(Path_Set, $p(s,y)$)),并将路 $p(s,y)$ 加入 $V_p(y)$ 中(Inset_Path_Vp($V_p(y)$,$p(s,y)$)).

模块确定当前最优路径的原则是:Path_Set 中代价最小的路,若同时有几条存在,则判断这几条路的时延与该路最外一点 y 到 t 的最小时延值之和,取最小的那条作为当前最优的路径.

模块停止的条件为:找到符合条件的解或当前 Path_Set 为空.

2.4　DCOP 算法性能分析

算法的计算复杂性主要体现在①和③,其中①需要用两次 Dijkstra 算法,其计算复杂度为 $O(|V|^2)$. ③的计算复杂度与算法搜索空间有关,

假设该空间的节点数目为 n,节点的最大邻居数目为 E_m,每个节点保留的最大路径数目为 L_m,则主约束计算模块最多进行 $(n\text{-}2)L_m$ 次主循环. 每次主循环需要最多 E_m 次副循环,每次副循环中最耗时的是算法第5步,需要最多 $(n\text{-}2)L_m$ 次比较,所以该模块的计算复杂度为 $O(E_m n^2 L_m^2)$,整个算法的计算复杂度为 $O(|V|^2+E_m n^2 L_m^2)$.

由上面的分析可知,主约束计算模块的计算复杂度主要与 n 和 L_m 有关,由于算法在保证最优解的同时有效地降低需要搜索的空间,此时的 n 和 L_m 一般很小,表1中给出了在节点数 $|V|$ 为100~1 000时不同规模的网络,本算法和HMCOP算法的主约束计算模块在最坏情况下所需要的最多比较次数,取得该数据的测试条件参见实验2,最终结果为得到的10 000个数据中的最大一个.

表1 两种算法主约束计算模块所需最多比较次数表 万次

$\|V\|$/个	100	200	300	400	600	800	900	1 000
HMCOP	9.2	116	75.6	237	487	1 008	1 194	1 203
DCOP	1.6	4.7	14.5	17.1	18.2	44.3	39.9	38.4

上表可看出在最坏的情况下,本算法主约束计算模块所需的最多比较次数约等于节点数的平方,与经典的 *Dijkstra* 算法计算复杂性大体相当. 而同样条件下的 *HMCOP* 算法计算复杂性要远大于 $O(|V|^2)$,本算法在计算复杂度方面比 *HMCOP* 算法改进了近一个数量级,详细仿真结果见下节.

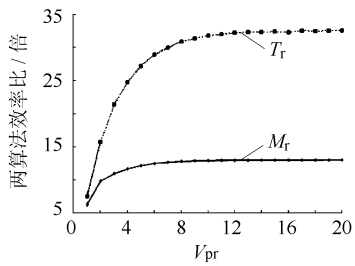
3 仿真分析

本节给出了 *DCOP* 在算法运行效率和性能方面的仿真测试,并对比同类型的已有算法 *HMCOP*. 其设计如下:每组实验测试10个不同的拓扑,拓扑采用 *Transit-Stub* 模型随机生成[8],每个网络中的链路代价在范围[0,1]内随机选取,时延在范围[0,20]内随机选取. 实验针对每个网络都产生1000次不同的请求,每次请求包括源宿节点和时延约束值,源宿节点是在所有节点中随机选取,而时延约束值则在[D_{min}, D_{cmin}]内随机取得,这样设计是考虑到在 $D_{min}>D_r$ 和 $D_{cmin}\leqslant D_r$ 时,两种算法都只执行了第1步,在耗时和性能方面没有区别. 每组实验针对不同的测量内容可得到10 000个数据,最终的结果取这些数据的均值. 仿真测试分为如下3组实验.

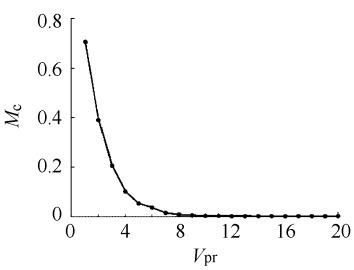
实验1 算法运行效率和性能比较

本组实验给出网络节点规模(节点数为500)相同时,两种算法在时间、空间复杂度和算法性能三方面的对比结果. 由于 *HMCOP* 算法在运行中每个节点保留最大路径数 V_{pr} 将直接影响算法效率和性能,实验分别给出 $V_{pr}=1,\cdots,20$ 的结果.

图2中 T_r 表示算法的耗时比,T_r=(*HMCOP* 主约束计算模块耗时)/(*DCOP* 主约束计算模块耗时),主要考虑算法主约束计算模块的耗时情况,其大小等于每次调用相应模块时,调用前后的计算机时间差. M_r 表示两种算法所需的储存空间比,主要考虑算法在计算过程中所需要的临时队列的大小,其计算表达式为:M_r=(*HMCOP* 临时队列大小)/(*DCOP Path _ Set*+V_p 大小). M_c 表示两种算法寻路性能比,比较两种算法得到的路径的代价,代价越小的得到的路径越优,其计算表达式为:(*DCOP* 得到代价小于 *HMCOP* 的次数)/10 000. 由图2可得 $V_{pr}=1$ 时,*HMCOP* 算法的效率接近 *DCOP* 算法(T_r=7.5),但此时 *DCOP* 算法有70.5%的解优于 *HMCOP* 算法,随着 V_{pr} 变大,存在两种相反的变化趋势:T_r 迅速上升(图2*a*)而 M_c 迅速下降(图2*b*),在 $V_{pr}=10$ 时 T_r 在32附近趋于平稳,而 $V_{pr}=8$ 后 M_c 趋于零. 结果表明 *DCOP* 算法在计算复杂度方面有明显改善,同时在空间复杂度方面也得到很好改善(M_r=10 左右).



a T_r、M_r 实验结果



b M_c 实验结果

图2 实验结果

实验 2　算法对网络规模变化的适应性比较

本组实验给出网络规模对两种算法影响情况的对比分析. 实验以节点数为 100 时每种算法的主约束计算模块耗时为基准,记为 B_t,分别给出节点数为 200 ~ 1 000 时每种算法的主约束计算模块耗时相对 B_t 的大小,即 T_s = (主约束计算模块耗时)/B_t. 实验中 *HMCOP* 的 V_{pr} 取值为 20.

由图 3 可得随着网络规模的变大,两种算法的 T_s 都呈上升趋势. 其中 *HMCOP* 算法的 T_s 变化剧烈,且呈发散状态,而 *DCOP* 算法的则变化缓慢,且趋于平稳. 由此可见随着网络节点数目的增加,*HMCOP* 算法运行效率将迅速下降,其算法对网络规模的适应性远不如 *DCOP* 算法.

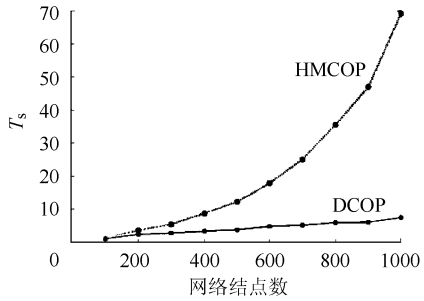


图 3　T_s 实验结果

实验 3　算法效率与约束条件"严格"程度的关系

HMCOP 算法的运行效率与 D_r 和 D_{min} 相对位置有关,称为算法约束条件"严格"程度,D_r 越逼近 D_{min} 则约束条件越"严格". 本组实验给出不同"严格"程度对两种算法影响情况的对比分析. 实验将"严格"程度分为 10 级,其取值 $C_i = D_{min} + (D_{cmin} - D_{min}) \times (2 \times i - 1)/20$, $(i = 1, \cdots, 10)$. 以 $i = 1$ 时 *DCOP* 的主约束计算模块耗时为基准,记为 B_t,分别给 $i = 1, \cdots, 10$ 时每种算法的主约束计算模块耗时相对 B_t 的大小,即 T_h = (主约束计算模块耗时)/B_t. 此时 *HMCOP* 中的 V_{pr} 取值为 10,网络节点数为 500.

由图 4 可得在约束条件非常"严格"($i = 1$)时,*HMCOP* 的效率接近 *DCOP*($T_h = 1.2$),但随着约束条件的"放宽",*HMCOP* 算法的效率迅速下降,而由于 *DCOP* 算法的自适应性,算法对约束条件的"严格"程度不敏感. 实际应用中约束条件是与业务相关的,这使得此时的 *HMCOP* 算法无法针对各种业务都取得很好的效果,算法实用性不如 *DCOP* 算法.

4　结 束 语

基于时延约束的最优路径求解算法在目前网络通信中具有十分重要的应用,但一直没能得到很好的解决. 本文采用自适应参数设计,有效减小搜索空间,给出一种 *DCOP* 算法,仿真表明该算法不仅在取得最优解、计算复杂度等关键指标上优于 *HMCOP* 算法,且具有适用于更大的网络规模、更复杂的业务类型、容易实现以及得到的最优路无环等优点. 该算法已经用 *C*++ 实现,可应用于各种通信网络,尤其适用于网络资源有限、节点计算能力弱、链路延迟较大的卫星网络和无线移动网络. *DCOP* 算法不仅可有效解决单约束最优路径问题,该算法也可扩展到求解多约束最优路问题上,这方面具体的研究将在另一篇文章中讨论.

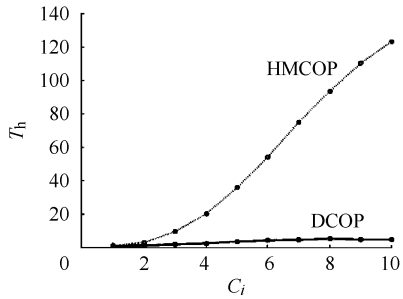


图 4　T_h 实验结果

参考文献 (**References**)

[1] Wang Z, Crowcroft J. Quality of service routing for supporting multimedia applications[J]. IEEE Journal on Selected Areas in Communications, 1996, 14(7):1219 ~ 1234

[2] Cheng S, Nahrstedt K. On finding multi-constrained paths[A]. In:IEEE ICC'98[C]. Atlanta: IEEE Communication Society, 1998. 874 ~ 879

[3] Korkmaz T, Krunz M. An efficient algorithm for finding a path subject to two additive constraints[J]. Computer Communications, 2002, 25(3):225 ~ 238

[4] Hussein F, Douglas S, Viniotis Y. Evaluation of multicast routing algorithm for real-time communication on high-speed networks[J]. IEEE Journal on Selected Areas in Communications, 1997, 15(3):332 ~ 345

[5] Widyono R. The design and evaluation of routing algorithms for real-time channels[R]. TR-94-024, 1994

[6] Yuan X, Liu X. Heuristic algorithms for multi-constrained quality of service routing[A]. In: Proceedings of the IEEE INFOCOM 2001[C]. Piscataway, NJ: IEEE Communication Society, 2001. 844 ~ 853

[7] Korkmaz T, Krunz M. Multi-constrained optimal path selection [A]. In: Proc of the IEEE INFOCOM 2001. Conference Proc [C]. Anchorage, Alaska: IEEE Communication Society, 2001. 834 ~ 843

[8] Calvert K I, Doar M B, Zegura E W. Modeling internet topology [J]. IEEE Communications Magazine, 1997,35(6): 160 ~ 163