

解决有复杂约束的最短路由问题的算法*

王宏** 王晟 李乐民

(电子科技大学 宽带光纤传输与通信系统技术国家重点实验室 成都 610054)

【摘要】介绍了有复杂约束的最短路由问题，应用K路由算法中的偏离算法解决该类问题。分析了在偏离算法的应用中影响算法效率的因素，提出了一种提高算法效率的改进方案，仿真结果表明改进方案是有效的。

关键词 K路由算法；偏离算法；最短路由；复杂约束；效率

中图分类号 TP311 文献标识码 A

Algorithm for Shortest Path Problem with Complex Constraints

Wang Hong Wang Sheng Li Lemin

(State Key Laboratory of Broadband Optical Fiber Transmission and Communication Networks, UEST of China Chengdu 610054)

Abstract The shortest path problem with complex constraints is described, and the application of deviation algorithm, a kind of k shortest paths algorithms, to this problem is introduced. The factors that affect the efficiency of the algorithm are analyzed, and an enhancement to the algorithm is proposed to improve the efficiency. Simulation results show that this enhancement is very efficient.

Key words K shortest paths algorithm; deviation algorithm; shortest path; complex constraints; efficiency

1 问题的提出

最短路由问题是网络技术研究中的一个重要问题，它分为无约束的最短路由和有约束的最短路由问题^[1]。无约束的最短路由问题已有大量的研究，如Dijkstra算法^[2]。因为网络技术中的很多问题可以被抽象为一定约束条件下的最短路由问题，所以有约束的最短路由问题也被广泛的研究。

1.1 有复杂约束的最短路由问题

文献[1, 2]给出了无约束的最短路由问题的准确描述，有约束的最短路由问题可视为一种约束条件下的最优化问题。其描述如下^[3]：

$$\begin{cases} \min f(x_1, x_2, \dots, x_n) \\ s.t. \quad g_i(x_1, x_2, \dots, x_n) \leq 0 \quad i=1, 2, \dots, m \\ \quad \quad h_j(x_1, x_2, \dots, x_n) = 0 \quad j=1, 2, \dots, k \end{cases}$$

式中 $\min f(x_1, x_2, \dots, x_n)$ 为优化目标， $g_i(x_1, x_2, \dots, x_n) \leq 0$ 为不等式约束条件， $h_j(x_1, x_2, \dots, x_n) = 0$ 为等式约束条件。最短路由问题的优化目标为

$$\min_{p \in P} c(p) = \min_{p \in P} \sum_{e_{ij} \in p} c(e_{ij})$$

2002年11月25日收稿

* 国家自然科学基金，编号：60002004

** 男 25岁 在职硕士生 主要从事光网络设计及宽带网络方面的研究

式中 P 为网络中一个原宿节点对之间所有路径的集合, p 为一条路径, e_{ij} 为网络中节点 i 到节点 j 的边, $c(\cdot)$ 为权值。采用普通最优化方法求解, 需要把约束条件 g_i 和 h_j 表示成边 e_{ij} 的函数。然而在很多实际问题中, 有的约束条件不能直接表示为边的函数, 称为复杂约束。求解复杂约束条件下的最短路径即为有复杂约束的最短路由问题。

1.2 实际问题中的复杂约束

在为求解最短路径而构造的抽象图上, 路由必须满足的约束通常反映在边权值的设定方法上。然而在一些问题中, 链路的某些属性比较复杂, 难以用边权值来表示, 如一个非马尔可夫过程的状态转移图中, 从一个状态到另一个状态的转移速率, 不一定是恒定的, 而是与怎样到达第一个状态有关。一些通信网络中的路由问题需要同时考虑链路资源分配问题, 如光网络中的(路由与波长分配)RWA问题^[4], 存在空闲波长的一条链路并不一定可用, 因为它与整条路由使用的波长有关。这种约束条件不能直接表示为抽象图上边的函数, 由于约束是在链路上, 称为复杂链路约束。此外, 通信网络的运营商或用户, 可能对某业务的路由作出一些指定, 如必须经过或不经过某链路也是一种复杂链路约束。

通常在抽象图上, 节点不具有属性, 它分为两种情况: 1) 图上的所有节点对路由的影响是相同的, 不需要设置属性加以区分; 2) 节点的信息可以转化为用边的信息来表达, 如网络中节点的失效可以用连接该节点的所有链路都失效来表示。然而实际情况并不都如此, 以光网络为例: 配置了不同设备的节点, 对信号的衰减作用是不同的, 在路由算法中, 需要对一些节点加以区别, 得出类似于“一条路由最多只能两次经过具波长变换能力的节点”的约束。通信网络的运营商或用户可能指定某些业务的路由必须经过或不经过某个或某类节点。这种关于节点的约束也不能直接表示为边的函数, 称为复杂节点约束。

1.3 两类算法

求解有约束的最短路由问题的算法, 可以分为两类。一类是针对某个或某类具体问题中的具体约束, 对无约束的最短路由算法进行扩展, 使之能处理该约束。该类算法针对性强、效率高, 但能解决的问题比较单一, 通用性不好。如果需要处理的约束很复杂, 扩展可能难以完成。如, 对于光网络的RWA问题, 一种常见的办法是分层图法^[4], 即按所有可使用的波长将网络分为若干个波长平面, 每一平面上只能使用一个波长, 然后依次在每个波长平面上寻找路由, 采用首次命中(first-fit)的方式找出可用路由, 把每个平面上的最短路由进行比较可以得到真正最短路由。但这种分层图法不适用解决用户指定必经链路的问题。

K 路由算法(K shortest paths algorithm)是一种用于解决有约束的最短路由问题通用的方法^[1, 5-7]。它不需要针对约束的具体情况来计算路由, 因而可用来解决有复杂约束的问题。

2 偏离算法应用于有复杂约束的最短路由问题

2.1 偏离算法步骤

在多种无环 K 路由算法中, 文献[5, 6]提出的一种偏离算法(deviation algorithm)具有较高的效率。其主要步骤如下:

- 1) 找出网络中所有节点到宿节点的最短路径, 即以宿节点为根的最短路径生成树。
- 2) 最短路径生成树中源节点到宿节点的路径 P 就是所求的 k 条路径中的第一条(最短的一条); 将 P 放入候选路径集合 X 。
- 3) 对网络中所有链路进行权值变化。
- 4) 对 P 逐点进行摆动, 得到一些新的从源节点到宿节点的路径, 放入候选路径集合 X 。
- 5) 选取 X 里的所有未考查过的候选路径中最短的一条作为新的 P , 若 P 无环, 则 P 就是下一条所求路径。
- 6) 反复执行4)和5), 直到找够所需的 k 条路径, 或已穷尽 X 。

2.2 算法思想

采用 K 路由算法解决有复杂约束的最短路由问题的基本思想是: 用 K 路由算法按路径长度由短到长的顺序依次给出源宿节点间的所有路径, 对所求出的每条路径按约束的要求进行筛选, 第一条满足要求的路径就是所求的最短路径。

将偏离算法的步骤5)略做修改, 就可以作为求解有复杂约束的最短路由问题的算法。由于它是在 K 路由

算法基础上进行扩展, 具有计算出由短到长的前 k 条满足复杂约束的路径的能力, 而不局限于只计算出最短路径。修改后的步骤5)为: 选取 X 里的所有未考查过的候选路径中最短的一条作为新的 P , 若 P 无环且满足约束的要求, 则 P 就是下一条所求路径。

当 $k=1$ 时, 即为求解满足复杂约束的最短路径。

2.3 适用范围

对不能规范描述的复杂约束, 算法采用排除法, 即通过否定不满足约束要求的路径来找到满足要求的路径。因此, 适用这种算法处理的复杂约束, 必须满足以下条件: 给定一条路径, 能容易判断该路径是否满足约束的要求。在很多实际问题中, 判断已知路径是否满足约束的要求, 比根据约束计算可用路径要简单得多。

3 对算法效率的改进与仿真结果

准确分析偏离算法的时间效率比较困难。该算法可分为三个主要模块: 最短路径生成树模块、权值变化模块、摆动模块。其中, 计算最短路径生成树一般可使用Dijkstra算法, 对节点数为 n , 链路数为 m 的图, Dijkstra算法的复杂度为 $O(n^2)$, 权值变化部分复杂度为 $O(m)$ 。最后的摆动部分的复杂度则较难度量, 但可以用摆动次数, 即候选路由由集合 X 的大小来反映, 而集合的大小不仅与图的规模有关, 还与图的拓扑有关, 当用于处理复杂约束时, 它还与网络的具体状态有关。

为了测试算法的实际效果与时间效率, 本文采用c++程序进行仿真。仿真方案为模拟通信网络中的路由与资源分配问题, 简要描述如下:

- 1) 拓扑: 采用 20×20 的2维mesh网络, 并划分为几个子网;
- 2) 链路权值: 表示链路长度, 随机设定为1、2、3中的某个值;
- 3) 链路状态: 表示该链路上的空闲资源;
- 4) 目标: 求出一个源宿节点对之间的由最短到最长的前50条可用路径, 不足50条的找遍所有可用路径为止;
- 5) 链路约束: 在同一子网内, 组成这一段路径的所有链路只能使用相同的资源, 在不同子网内可以使用不同的资源。

大量仿真表明, 算法的效率不稳定。当全网所有链路的空闲资源分布的统计特征固定, 但分布具体情况不同时, 算法效率有较大起伏。这与理论分析中的算法效率与“网络的具体状态有很大关系”是一致的。但算法效率还可以提高。如图1所示: 链路 a 、 b 组成了当前待摆动路由 P 上的一段, 且属于同一子网。设摆动顺序是从左到右, 链路 a 、 b 没有相同的空闲资源, 即 P 不可用。若 $a-c$ 所示的是可行的路由, 但相对长度较大, 则从集合 X 中取出的路由 P' 不是从节点 I 摆动得到的 $a-c$ 所示的路由, 而是从节点 J 或 J 以后的节点摆动得到的路由。这样 $a-b$ 仍在 P' 上, 故 P' 不满足链路约束。然后以 P' 为基础进行下一轮摆动, 基于同样的原因, 再次从集合 X 中取出的路由 P'' 可能仍然含有 $a-b$ 段。这样在找到正确的 $a-c$ 所示路由之前需进行大量无效的摆动, 集合 X 也变得较大, 效率大大降低。

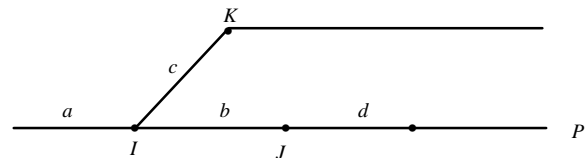


图1 禁止点对效率的影响

针对这一问题, 对原有算法做了如下改进: 当从集合 X 取出的路由 P 含有 $a-b$ 段时, 则将节点 J 标记为禁止点, P 上的禁止点与禁止点之后的点都不再进行摆动, 这样就减少了无效摆动。算法的流程图如图2所示。

不论问题的约束是链路约束还是节点约束, 只要在判断给定路由是否满足约束要求时, 否定路由的一段就可以否定整条路由, 通过设置禁止点就可减少无效摆动。

表1是算法改进前后时间效率的仿真结果比较。从表中可以看出: 在采用设置禁止点改进后, 候选路由由集合 X 的大小有了明显的下降, 算法的用时也随之减少。在表1所示的条件下, 超过50%的测试中, 改进后的候选集合大小比改进前的减小了一个数量级; 改进前的仿真用时超过120 s的比例为35%, 而改进后仅为5%。

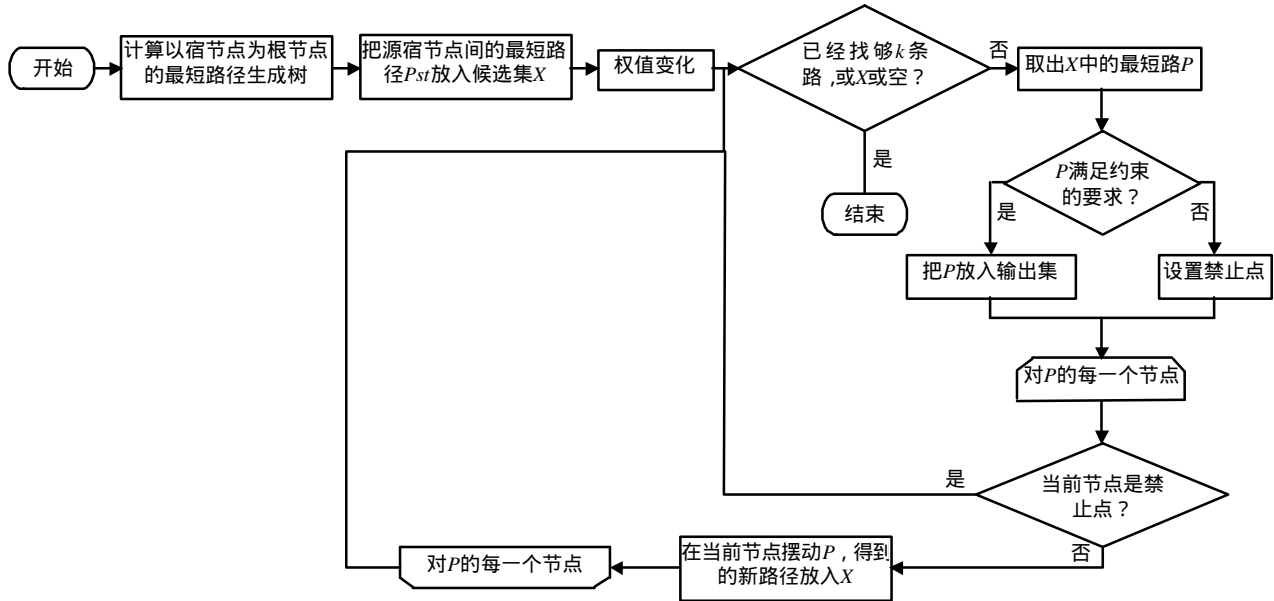


图2 改进后的算法流程图

表1 改进前后效率比较

空闲资源号为1的链路数 / 空闲资源号为2的链路数 = 0.98 / 0.02

仿真次数	改进前用时/s	改进前候选集合大小	改进后用时/s	改进后候选集合大小
1	3	5 531	1	1 498
2	7	17 263	2	2 192
3	>120	>100 000	22	24 208
4	1	1 337	1	936
5	1	3 260	1	967
6	5	9 528	1	1 996
7	1	963	1	963
8	>120	>100 000	4	6 960
9	64	65 535	4	3 042
10	1	2 900	1	1 254

空闲资源号为1的链路数 / 空闲资源号为2的链路数 = 0.93 / 0.07

仿真次数	改进前用时/s	改进前候选集合大小	改进后用时/s	改进后候选集合大小
1	51	54 933	13	12 837
2	>120	>100 000	>120	>100 000
3	>120	>100 000	1	1 190
4	>120	>100 000	41	26 139
5	1	1 092	1	880
6	>120	>100 000	6	8 339

表1(续) 改进前后效率比较

仿真次数	改进前用时/s	改进前候选集合大小	改进后用时/s	改进后候选集合大小
7	2	3 222	2	1 128
8	3	7 471	3	4 024
9	21	31 610	5	4 614
10	>120	>100 000	26	21 074

注:表中的每一行是一次仿真的数据。每次仿真使用完全相同的网络状态,分别测试并记录改进前和改进后的算法的效率。表中“>120”指用时超过120 s,“>100 000”指候选集合大小超过100 000。

4 结 论

总结了有约束的最短路由问题的两种方法:针对具体约束扩展无约束最短路由算法和应用 k 路由算法。通过比较,指出了第一种方法有的通用性不好,肯定了第二种方法在解决有复杂约束的问题时的优势,并分析了它的适用范围,指出其只有在能快速判断给定路由是否满足约束时才具有较好效果。

偏离算法作为一种高效的 k 路由算法得到了较广泛的应用。本文具体阐述了怎样应用偏离算法解决有复杂约束的最短路由问题,分析了应用于该类问题时,影响算法效率的因素,提出了一种对算法效率的改进方案。给出了仿真结果,通过比较改进前后的算法效率,说明了改进是有效的。

参 考 文 献

- [1] Ernesto Q, Vieira M, Marta M B, *et al.* The k shortest paths problem[EB/OL]. Research Report, CISUC, http://www.mat.uc.pt/~eqvm/cientificos/investigacao/r_papers.html, 1998-06
- [2] Sara B, Allen V G. Computer algorithm: introduction to design and analysis (3rd Ed.)[M]. 影印版,北京:高等教育出版社,2001
- [3] 傅英定,成孝予,唐应辉. 最优化理论与方法[M]. 成都:电子科技大学出版社,1996
- [4] 徐世中,李乐民,王 晟. 多光纤波分复用网动态路由和波长分配算法[J]. 电子学报,2000,28(7): 23-27
- [5] Ernesto Q, Vieira M, José L E, *et al.* A new shortest paths ranking algorithm[J]. *Investigação Operacional*, 2000, 20(1): 47-62
- [6] Ernesto Q, Vieira M, Marta M B, *et al.* Deviation algorithms for ranking shortest paths[J]. *International Journal of Foundations of Computer Science*, 1999, 10 (3): 247-261
- [7] Eppstein D. Finding the k shortest paths[J]. *SIAM J. Computing*, 1998, 28(2): 652-673

编辑 漆 蓉