

基于背离路径的 Kth 最短路径实用搜索算法

傅俊伟¹, 李兴明¹, 陈捷²

(1. 电子科技大学 宽带光纤传输与通信网技术教育部重点实验室, 四川 成都 610054;

2. 中兴通讯股份有限公司, 广东 深圳 518057)

摘要:基于背离路径的概念,设计 Kth 最短路径实用搜索算法。通过对第 $K-1$ 最短路径求背离路径,求得第 K 最短路径。算法时间复杂度限制在 $O(e \times n^2)$,其中 e 为图的总边数, n 为图的顶点数。在实时应用中,文中的算法有很好的应用前景。该算法已经成功应用到一个传输网络规划系统的动态 RWA 问题中。

关键词:WDM 光网络; Kth 最短路径; 背离路径

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2009)02-0120-03

A Practical Algorithm for Finding the Shortest Kth Path Based on Deviation Path

FU Jun-wei¹, LI Xing-ming¹, CHEN Jie²

(1. Ministry of Education Key Lab. of Broadband Optical Fiber Transmission and Communication Systems,

University of Electronic Science and Technology of China, Chengdu 610054, China;

2. ZTE Corporation, Shenzhen 518057, China)

Abstract: Based on the concept of deviation path, a practical algorithm was designed for finding the shortest Kth path. By deviating path from the $(K-1)$ th shortest path, this algorithm can get the shortest Kth path. The time complexity of this algorithm is limited by $O(e \times n^2)$, while e is the total number edges of a topology. This algorithm has good prospects in the real-time application.

Key words: WDM optical networks; the shortest Kth path; deviation path

0 引言

随着通信网业务量的爆炸性增长以及高性能的 WDM 光网络设备的出现,波分复用技术成为下一代骨干网络的核心技术。光网络要提供我们所要求的物理性能,就必须对现有可用资源进行高效利用和优化。其中,路由与波长分配(RWA)问题是最优化网络性能的核心问题之一。

RWA 问题分为静态的和动态的。静态的业务是指节点间的所有连接请求事先知道,此时 RWA 问题解决的是从全局优化的角度为所有连接请求建立光路,这种计算可以是离线的。而对于动态 RWA 问题,光路需求随机、顺序到达网络,当一条光路持续一段时间后又被拆除,要为每一条作实时 RWA 计算。当网

络规模较大时,为了减小复杂性,常将 RWA 问题分为选路和波长分配两步进行^[1]。通常选路采用前 K 条最短路径作为备用路由集,然后在备用路由集上根据某种规则选取一条备用路由进行波长分配^[2,3]。

国内外学者对于波长分配算法已经有很多的研究^[1-3],而对于如何得到最短路由集,却不见有很好的算法。传统的求取两节点间的前 K 条最短路径算法是遍历图中所有可能路径,将所有路径排序取前 K 条路径。通常,文献中采用的是离线计算得到的最短路由集,这样会浪费大量存储空间,而且,当网络拓扑变化时,路由就会出错,因此需要有一种实时算法计算最短路由集。

文献[4]利用在第 $K-1$ 最短路径所在子图的基础上删边派生子图,然后在派生子图上求最短路径的方法求第 K 最短路径(具体派生子图方式见文献[4])。第 $K-1$ 最短路径所在子图并非只有一个,如果这种子图有 m 个,那么求第 K 最短路径的复杂度就是 $O(m \times n^3)$,如果一个图的节点很多, m 将很大。考虑到实时性问题,该算法不能解决动态 RWA 问题。

收稿日期: 2008-05-30

基金项目: 中兴通讯研究基金项目(GT-2005-N004)

作者简介: 傅俊伟(1985-),男,硕士研究生,主要研究方向为宽带网络技术、网络规划与优化;李兴明,教授,博士生导师,主要研究方向为网络管理、光纤通信、光交换与自动光网络。

文献[5]利用背离路径的概念,通过使用 2nd 最短路径搜索算法,可以成功求得 3rd 最短路径,但是文中在求第 K 最短路径($K > 3$)时,求得的不是实际意义上的第 K 最短路径。

文中基于背离路径的概念,重新设计算法,将求第 K 最短路径的复杂度限制在 $O(e \times n^2)$,其中 e 为图的总边数。

1 问题定义及一些用语的说明

文中要解决的问题是,在一个给定权值的图中,求取源目的节点间的前 K 条最短路径作为备用路由集。文中基于无向图,链路权值以跳数计。实际应用中也定义其他的权值,由于最短路径算法基于 Dijkstra 算法,链路权值必须为非负数。下面定义一些用语。

1.1 背离路径

假设 v_1 为源节点, v_n 为目的节点。 (v_1, v_n) 间的两条路径 $v_1 v_2 v_3 v_4 \cdots v_n$ 和 $v_1 v_2' v_3' v_4' \cdots v_n$, 如果 $v_1 v_2 \cdots v_i$ 和 $v_1 v_2' \cdots v_i'$ 节点相同,在 v_{i+1} 处开始不同,则称路径 $v_1 v_2' v_3' v_4' \cdots v_n$ 由 $v_1 v_2 v_3 v_4 \cdots v_n$ 在 v_{i+1} 处背离得到, v_{i+1} 为被背离节点, v_{i+1}' 为背离节点。可见, v_{i+1}' 必须为 v_i 的邻接点,且与 $v_1, v_2, \dots, v_{i-1}, v_{i+1}$ 不同。

定义 1 背离路径:路径 $v_1 v_2 v_3 v_4 \cdots v_n$ 以 v_{i+1} 为被背离节点,以 v_{i+1}' 为背离节点的所有背离路径中有一条最短,它由 $v_1 v_2 \cdots v_i +$ 链路 $v_i v_{i+1}' + v_{i+1}'$ 到 v_n 的最短路径得到。

文中,以 v_{i+1}' 为背离节点的背离路径专指这条最短路径,而非以 v_{i+1}' 为背离节点的其它路径。在求 v_{i+1}' 到 v_n 的最短路径时,应删除与节点 $v_1 v_2 \cdots v_i$ 有关的链路,再利用 Dijkstra 算法求最短路径,以避免拼接而成的背离路径产生环路。

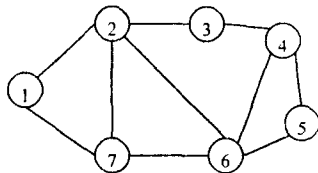


图 1 背离路径图示

例如,在图 1 中,链路权值以跳数计,节点 1 到节点 5 的最短路径为 $1 \rightarrow 2 \rightarrow 6 \rightarrow 5$,该路径的节点 2、6、5 均可以是背离节点。其中,以 2 为背离节点,可得背离路径 $1 \rightarrow 7 \rightarrow 6 \rightarrow 5$ (7 为背离节点, $1 \rightarrow 7 \rightarrow 6 \rightarrow 4 \rightarrow 5$ 也是以 2 为背离节点、7 为背离节点的背离路径,但是文中专指前面这条短的路径为以 7 为背离节点的背离路径);以 6 为背离节点,可得背离路径 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ (3 为背离节点)和 $1 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 5$ (7 为背离节

点);以 5 为背离节点,可得背离路径 $1 \rightarrow 2 \rightarrow 6 \rightarrow 4 \rightarrow 5$ (4 为背离节点)。

1.2 路径前缀

两节点间的路径除第 1 最短路径外,其它任何一条路径必是某条路径的背离路径,对应有一个背离节点。定义一条路径的源节点到这条路径的背离节点的这段路径为路径前缀。两节点间的路径除第 1 最短路径外都有路径前缀。定义 devNode 为一条背离路径背离节点的编号。对于第 1 最短路径,devNode 可设为 1。

例如,在 1.1 的举例中,路径 $1 \rightarrow 2 \rightarrow 6 \rightarrow 5$ 没有路径前缀(devNode = 1);路径 $1 \rightarrow 7 \rightarrow 6 \rightarrow 5$ 、 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ 、 $1 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 5$ 、 $1 \rightarrow 2 \rightarrow 6 \rightarrow 4 \rightarrow 5$ 的路径前缀分别是 $1 \rightarrow 7$ (devNode = 2)、 $1 \rightarrow 2 \rightarrow 3$ (devNode = 3)、 $1 \rightarrow 2 \rightarrow 7$ (devNode = 3)、 $1 \rightarrow 2 \rightarrow 6 \rightarrow 4$ (devNode = 4)。

1.3 背离路径集 G1 的构造方法及性质

假设源目的节点 (v_1, v_n) 间的第 1 最短路径 $v_1 v_2 v_3 v_4 \cdots v_n$ 。背离路径集 G1 是指由以下步骤构造得到的集合。

(1) G1 为空;

(2) $v_1 v_2 v_3 v_4 \cdots v_n$ 以 v_i ($2 \leq i \leq n$) 为被背离节点派生背离路径(背离路径定义见 1.1),所有背离路径放入 G1 中,并记下每条背离路径的 devNode;

(3) G1 中的任何一条路径以其第 devNode 个节点后的节点为被背离节点派生背离路径,在 G1 中删除该路径,该路径派生得到的背离路径放入 G1 中;

(4) 重复步骤(3)。

背离路径集 G1 具有如下性质:

a) G1 中所有路径的路径前缀都不同。

b) G1 中不会有一条路径的路径前缀是另一条路径的路径前缀的子集(即不会有一条路径的路径前缀是另一条路径的路径前缀的一部分)。

下面对上述性质进行证明。

证明:步骤(1)中 G1 没有路径,性质 a、b 显然成立。由于路径前缀由得到背离路径的原路径的源节点到被背离节点前一节点的一段路径加上背离节点组成。所以路径前缀不会是原路径的一部分。对于步骤(2),原路径为第 1 最短路径,不同被背离节点派生的得到的派生路径,由于源节点到被背离节点前一节点的一段路径不同,路径前缀肯定不同,加上背离节点,性质 b 也成立。对于相同被背离节点派生得到的背离路径,源节点到被背离节点前一节点的一段路径相同,但是背离节点不同,所以 a、b 成立。对于步骤 3,设 G1 中的任何一条路径以其第 devNode 个节点后的节点为被背离节点派生背离路径的路径集合 G2,类似步骤

2, G_2 也满足性质 a, b。由于步骤 2 得到的 G_1 具有 a, b 性质, 而 G_2 中路径的前缀有相同部分(即选中的派生的路径的路径前缀)。由于选中的派生的路径已经从 G_1 删除, 所以将 G_2 加入 G_1 后, 新的 G_1 仍满足性质 a, b。步骤 4 只是步骤 3 的重复, 不会改变 G_1 的性质。得证。

由以上分析可知, 这种构造方法得到背离路径集 G_1 里的所用路径均不相同, 并且任何一条路径派生的背离路径都不会比原路径短。

2 算法实现

K th 最短路径的搜索算法是下述流程的步骤 3 在求得第 K 最短路径时终止的算法。

通过以下流程可以得到源目的节点(v_1, v_n)间的所有路径。

(1) 用 Dijkstra 算法求原图源目的节点(v_1, v_n)间的第 1 最短路径 $path(1) = v_1 v_2 v_3 v_4 \cdots v_n$, $devNode = 1$; 如果 $path(1)$ 不存在算法终止。 G_1 初始化为空。 $v_1 v_2 v_3 v_4 \cdots v_n$ 以 $v_i (2 \leq i \leq n)$ 为被背离节点派生背离路径, 所有背离路径放入 G_1 中, 并记下每条背离路径的 $devNode$, 如果 G_1 为空算法终止。

(2) 将 G_1 中的路径排序, 最短的那条为原图的第 2 最短路径。第 2 最短路径以其第 $devNode$ 个节点后的节点为被背离节点派生背离路径, 记下每条背离路径的 $devNode$, 将产生的背离路径放入 G_1 中, 在 G_1 中删除第 2 最短路径。如果 G_1 为空算法终止。

(3) 假设已经求得第 $K-1$ 最短路径 $path(K-1)$, 其背离节点编号为 $devNode$, 此时路径集 G_1 。那么, 第 K 最短路径是 G_1 中最短的那条。第 K 最短路径以其第 $devNode$ 个节点后的节点为被背离节点派生背离路径, 记下每条背离路径的 $devNode$, 将产生的背离路径放入 G_1 中, 在 G_1 中删除第 K 最短路径。

(4) 如果 G_1 不为空, 那么重复步骤(3), 可以求得源目的节点(v_1, v_n)间的所有路径。

下面证明上述流程的正确性。

证明: 步骤(2)是由于第 2 最短路径只可能是与第 1 最短路径前 i 个节点相同(包括节点顺序相同, $1 \leq i \leq n-1$), 它刚好对应于第 1 最短路径以 $v_i (2 \leq i \leq n)$ 为被背离节点派生背离路径的情况。故步骤(2)成立。

对于步骤(3), 由步骤 2, $K=2$ 时已经成立, 当 $K \geq 3$ 时, 由于 $path(K)$ 必须是某条路径 $path(i)$ 的背离路径。

(a) 如果 $path(i)$ 比 $path(K-1)$ 短, 那么按照步骤 3 的派生背离路径的方式, $path(i)$ 肯定已经派生过, 由

于派生的所有背离路径在 G_1 中, 所以 $path(K)$ 在 G_1 中。

(b) 如果 $path(i)$ 比 $path(K-1)$ 长, 那么按照步骤 3 的派生方式 $path(i)$ 没有被派生过, 由 1.3 的分析, 任何一条路径派生的背离路径都不会比原路径短, 那么 $path(K)$ 最多为 $path(i)$, 而不会是 $path(i)$ 的背离路径。

(c) 如果 $path(i)$ 和 $path(K-1)$ 等长, 如果 $path(i)$ 已经派生过, 由于派生的所有背离路径在 G_1 中, 所以 $path(K)$ 在 G_1 中。如果 $path(i)$ 没有派生过, 那么 $path(K)$ 最多为 $path(i)$, 而不会是 $path(i)$ 的背离路径。所以第 K 最短路径在 G_1 中, 且为 G_1 中最短的那条。

3 算法复杂度分析

1.3 中已经证明, G_1 中的路径不会相同, 所以算法步骤(3)执行一次将得到一条路径。由于第 K 最短路径是第 $K-1$ 最短路径以 $devNode$ 节点后的节点为被背离节点派生背离路径后的 G_1 中最短的那条, 而求一条背离路径需要一次 Dijkstra 算法, 背离路径的条数与第 $K-1$ 最短路径在 $devNode$ 节点后的节点的邻接点个数有关, 第 $K-1$ 最短路径的路径前缀越长, $devNode$ 节点后的节点数越少, 需要执行 Dijkstra 算法的次数就越少。设 n 为图的总节点数, 那么 Dijkstra 算法的时间复杂度为 $O(n^2)$ 。显然, 背离路径的条数不会超过图的总边数 e , 在实际计算中, 远达不到。所以, 求第 K 最短路径的时间复杂度限制在 $O(e \times n^2)$ 。

4 实例分析

图 2 是 NSFNET 网络, 利用文中算法与文献[4]的算法对图 2 的拓扑图进行求解所有路径, 两者的结果相同, 但是文中算法在时间上优势非常明显。求得图 2 中所有节点间的路径(共 7780 条路径)文中算法不到 1 秒时间。而文献[4]的算法, 仅仅求节点对 (1, 16) 间所有路径(共 96 条路径)的时间就超过了 30 秒。

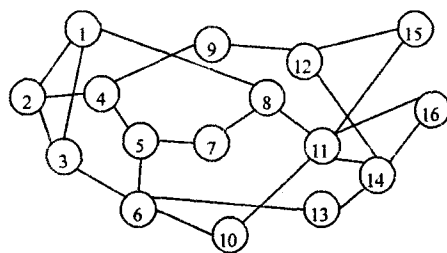


图 2 NSFNET 网络

全率)和精度(查准率)均可达到近 100%。经过分析,清华大学出版社由于其图书发布页面以列表形式显示。

为了抽取其详细数据,利用二次抽取完成数据抽取,在抽取过程中一些图书信息定位不同造成一些图书信息未能抽取出来。但从整体实验结果来看,此 Web 包装器的设计的可行性,体现了其高效性,可以应用于 Wrapper/Mediator 方法的 Web 数据集成^[9]。

4 结束语

设计的基于预定义模式的 Web 包装器考虑到了 Wrapper 的维护问题,当 Web 页面发生变动时,只需修复抽取规则(Config.xml)即可,而不用改动、重新编写程序,便于 Wrapper 的维护和源程序的重用。基于此 Web 包装器的设计与实现,已经实现了 Wrapper/Mediator 方法的新书信息集成查询系统(如图 3 所示),通过统一的查询接口,提供新书查询检索服务。

下一步的研究工作是利用此 Web 包装器的设计思想,在不损失其召回率和精度的情况下,设计并实现针对某个领域的通用 Web 包装器,利用 Wrapper/Mediator 方法开发基于 Web Service 的 Web 数据集成系统^[10]。

参考文献:

- [1] 孟小峰. Web 信息集成技术研究[J]. 计算机应用与软件, 2003,20(11):32-36.
- [2] Raposo J, Pan A, Alvarez M, et al. Automatically maintaining wrappers for semi-structured web sources[J]. Data & Knowledge Engineering, 2007,61:331-358.
- [3] Knoblock C A, Lerman K, Minton S, et al. Accurately and Reliably Extracting Data from the Web: A Machine Learning Approach[J]. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2000,23(4):33-41.
- [4] 刘 迁,焦 慧,贾惠波. 信息抽取技术的发展现状及构建方法的研究[J]. 计算机应用研究, 2007,24(7):6-9.
- [5] 王敬普,林亚平,周顺先,等. 基于包装器模型的文本信息抽取[J]. 计算机应用, 2007,27(3):655-658.
- [6] 杜冬梅,许彩欣,苏 健. 浅谈正则表达式在 web 系统中的应用[J]. 计算机系统应用, 2007(8):87-90.
- [7] Holzner S. XML 完全探索[M]. 北京:中国青年出版社, 2001:7-10.
- [8] 贺令亚,柳佳刚. 基于 Web 的包装器技术的现状与发展[J]. 电脑开发与应用, 2007,20(6):27-29.
- [9] 师雪霖,牛振东,宋瀚涛. 基于中介器/包装器的联合数字图书馆集成信息检索机制[J]. 计算机应用, 2005,25(3):703-705.
- [10] 张素智,李宏伟,李树凯. 基于 Web 服务的数据集成[J]. 郑州轻工业学院学报, 2005,20(4):34-37.

(上接第 122 页)

表 1 文中算法与文献[4]的算法结果对比

长度	文中算法	文献[4]算法
3	1→8→11→16	1→8→11→16
4	1→8→11→14→16	1→8→11→14→16
5	1→3→6→10→11→16	1→3→6→13→14→16
5	1→3→6→13→14→16	1→3→6→10→11→16
6	1→3→6→10→11→14→16	1→2→4→9→12→14→16
6	1→3→6→13→14→11→16	1→2→3→6→10→11→16
6	1→2→3→6→10→11→16	1→3→6→13→14→11→16
6	1→8→11→15→12→14→16	1→2→3→6→13→14→16
6	1→2→3→6→13→14→16	1→3→6→10→11→14→16
6	1→2→4→9→12→14→16	1→8→11→15→12→14→16

表 1 列出了两者求解节点 1 到节点 16 的前 10 条最短路径的结果。两者结果相同,不同的只是求得相同长度的路径的顺序不同。

文中算法的优势在于 G1 中路径的唯一性,以至于 G1 中的最短路径一次派生背离路径,就会得到下一条最短路径。而 G1 中的最短路径派生背离路径时,如果其路径前缀很长,那么派生的次数明显减少,时间复杂度将远低于 $O(e \times n^2)$,从而加速了搜索速度。文中算法已经成功应用到中兴通讯基金资助项目

“传输网络规划系统 NetNumen™TOP”中,取得了很好的效果。

5 结束语

基于背离路径设计的算法,有效地解决了前 Kth 最短路径的搜索问题,其时间复杂度为 $O(e \times n^2)$ 。在实时应用中,文中的算法有很好的应用前景。

参考文献:

- [1] 李乐民. WDM 光传送网的选路和波长分配算法[J]. 中兴通讯技术, 2001,6:4-7.
- [2] Mokhtar A, Azizoglu M. Adaptive wavelength routing in all-optical networks[J]. IEEE/ACM Trans. on Networking, 1998,6(2):197-206.
- [3] Alanyali M, Ayanoglu E. Provisioning algorithms for WDM optical networks[J]. IEEE/ACM Trans. on Networking, 1999,7(5):767-778.
- [4] 柴登峰,张登荣. 前 N 条最短路径问题的算法及应用[J]. 浙江大学学报, 2002,36(5):531-534.
- [5] 王明中,谢剑英,陈应麟. 一种新的 Kth 最短路径搜索算法[J]. 计算机工程与应用, 2004(30):49-50.