

K 则最短路径算法效率与精度评估

高松 陆锋

(中国科学院地理科学与资源研究所 资源与环境信息系统国家重点实验室,北京 100101)

摘要 精度和效率是决定最短路径算法实用价值的重要依据。对于 K 则最短路径问题,各种理论严密算法和有损算法的实用性分析是目前研究的薄弱环节。理论严密算法的实际运行效率比较及其有损算法的精度损耗与效率提高幅度的量化一直未得到深入研究。针对这一问题,在对 K 则最短路径算法进行系统分类的基础上,分析了各种经典的理论严密算法和精度有损算法的特征与时间复杂度,结合实际城市路网数据对各种 K 则最短路径算法的运行效率和精度进行了测试和比较。结果显示,与有损算法相比,理论严密的 K 则最短路径算法普遍缺乏实用性,只有多重标号算法适合于某些要求精度无损的应用;而一些有损 K 则最短路径算法以较小的精度损失换取了较大幅度的效率提高,尤以双向搜索算法最具应用推广价值。

关键词 K 则最短路径算法 交通网络 效率 精度

中图法分类号: TP391, U491 文献标识码: A 文章编号: 1006-8961(2009)08-1677-07

The Kth Shortest Path Algorithms: Accuracy and Efficiency Evaluation

GAO Song, LU Feng

(State Key Laboratory of Resources and Environmental Information System, Institute of Geographical Science and Natural Resources Research, CAS, Beijing 100101)

Abstract Efficiency and accuracy are undoubtedly two exclusive indices for shortest path algorithms. The practical analysis on different algorithms solving the Kth shortest path problems has not invoked much research. The comparison on running efficiency between several theoretically rigorous algorithms and the amount of accuracy loss and efficiency improvement brought by lossy Kth shortest path algorithms have remained quantitatively under investigated, and the trade-off between them has not been analyzed in detail. In this paper, with a systematic classification of the most popular Kth shortest path algorithms, the author discussed the characteristics and time complexity of these algorithms, and analyzed and compared their efficiency and accuracy with a real roads network. The author argued that the theoretically rigorous Kth shortest path algorithms, generally lack practical value, only the multi-label setting algorithm can be applied for some applications requiring lossless accuracy. Some lossy Kth shortest path algorithms, greatly improve the efficiency with only little accuracy lost. The bidirectional-search algorithm was argued worth paying more attention in practical applications.

Keywords Kth shortest paths algorithms, roads network, efficiency, accuracy

1 引言

K 则最短路径问题是最短路径问题族中的重要问题之一^[1]。K 则最短路径旨在寻找起点和终点间

的多个备选优化路径,形成最短路径组,以最大程度满足用户对不同出行路径的选择需求。K 则最短路径问题最早由 Hoffman 和 Pavley 提出^[2],多年来一直受到业界的广泛重视,提出了各种各样的 K 则最短路径算法。根据算法的理论严密性,K 则最短路

基金项目:国家高技术研究发展计划(863)项目(2006AA12Z209,2007AA12Z241);中国科学院知识创新工程前沿基金项目(CXIOG-D04-02)

收稿日期:2007-12-12;改回日期:2008-05-19

第一作者简介:高松(1982~),男,中国科学院地理科学与资源研究所、资源与环境信息系统国家重点实验室、地图学与地理信息系统专业硕士研究生。主要从事交通地理信息系统技术研究。E-mail:songziwell@163.com

径算法可以分为严密 K 则最短路径算法^[2-8]和有损 K 则最短路径算法^[9-13]两类。前者具有严密的理论基础,所得到的 K 条备选路径是起终点间所有可能路径按权值递增排序后的前 K 条路径。而后者通过启发式策略等技术,通过牺牲有限的精度以换取效率的大幅度提高。根据路径限制条件,K 则最短路径算法还可以分为限定无环算法^[14-17]和自由路径算法^[18-24]。前者要求所有求得的最短路径都必须是简单路径,不能含有回路,而后者则允许路径中含有回路。在实际应用中,限定无环路径的 K 则最短路径应用更加普遍。

与单源最短路径问题相比,K 则最短路径问题在算法设计上更为复杂,对算法结果的认可度也存在较大差异。因此,到目前为止,并没有一种 K 则最短路径算法如单源最短路径算法族中的 Dijkstra 算法、Pallottino 算法一样得到业界共识。本文旨在对目前业界所提出的各种认可度较高的 K 则最短路径算法进行系统分类,通过理论分析和实践验证,对各种算法的复杂度和运行效率进行比较和评价,为算法的实用性研究提供参考。

2 K 则最短路径问题描述

K 则最短路径问题定义如下:给定赋权图 $G(V, E)$ 及其上两个顶点 v_i 和 v_j , r_{ij} 为 v_i 和 v_j 之间的一条路径,记其权值为 $d(r_{ij})$,由 v_i 和 v_j 之间的所有互不相同的路径组成的集合 $R(G, v_i, v_j)$ 称为

G 上 v_i 和 v_j 之间的路径集合,即 $R(G, v_i, v_j) = \{r \mid r \text{ 为 } G \text{ 上 } v_i, v_j \text{ 之间的所有路径}\}$ 。按路径权值大小将其排列,得到 $r_1, r_2, \dots, r_m, d(r_1) \leq d(r_2) \leq \dots \leq d(r_m)$,则称 r_1 为 G 上 v_i 和 v_j 间的第 1 最短路径, r_2 为第 2 最短路径,直至 r_m 为第 m 最短路径。求取 G 上 v_i 和 v_j 间的第 1 ~ K ($K \leq m$) 最短路径的问题称为 K 则最短路径问题^[8],如图 1 所示,从节点 A 到 E 的前 3 条最短路径为 $A \rightarrow D \rightarrow E, A \rightarrow B \rightarrow E$ 和 $A \rightarrow C \rightarrow E$ 。在实际应用中,路径权值除长度外,还可以是时间、费用或其他因素。

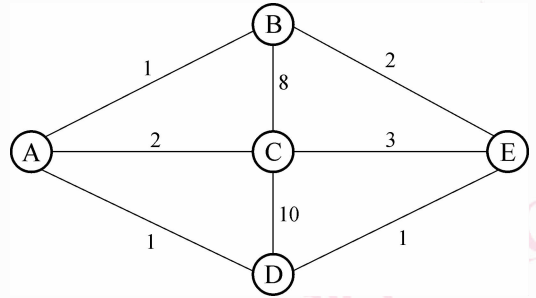


图 1 K 则最短路径问题示例

Fig. 1 Demonstration of the Kth shortestest path problem

3 K 则最短路径算法分类

到目前为止,业界所提出的各种 K 则最短路径算法,可按照图 2 所示分类体系进行分类。

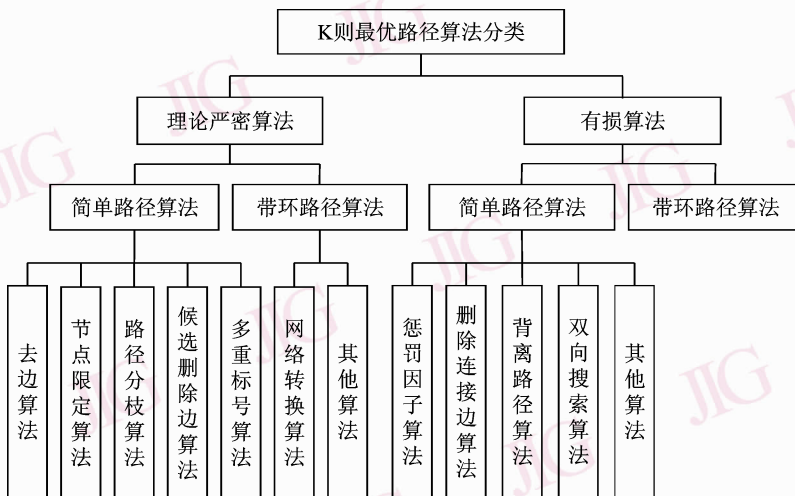


图 2 K 则最短路径算法分类体系

Fig. 2 Classification of the Kth shortestest path algorithms

3.1 理论严密算法

去边算法是最经典的理论严密 K 则最短路径算法^[2]。它的基本思想是:先用 Dijkstra 算法求解起终点间的最短路径;然后轮流把这条路径中的一条边设置为禁行,再分别用 Dijkstra 算法求解起终点间的最短路径,这些路径中权值最小的路径为次短路径;依此类推即得到起始终止节点间的 K 则最短路径。

节点限定算法通过限制中间节点数目来求解起终点间的 K 则最短路径,采用多次迭代求解过程^[3],在第 i 次迭代中,限定起、终点间的节点个数不超过 i ,当进行 n 次迭代后(n 为网络中节点数)算法结束,得到起始终止节点间的 K 则最短路径。

路径分枝算法基于路径分枝和等价分类概念构造已计算出的前 i 条最短路径的分枝结构树,表达前 i 条路径之间在逻辑上的分枝关系,根据树结构把起终点间的其余路径划分在不同的等价类集合中。所有这些集合中路径最小者,就是第 $i+1$ 条最短路径^[7]。

候选删除边算法通过网络分解求解 K 则最短路径。它的基本思路是^[3]:首先在当前网络中求取最短路径,然后把当前网络分解,构造新的候选子图集,新的子图集包含起、终点在原始网络中除已得路径外的所有路径。网络分解中构造的子图都是原始网络删除某些边后的子图。候选删除边算法可以看作是去边算法的改进,但是需要大量子图分解、计算和排序。

多重标号算法对每个节点建立一个标号链表,通过对该链表每个元素的多次松弛和排序,在路径搜索过程中一次性得到起、终点间 K 则最短路径^[8]。每个节点需要维护一个排序的标号链表序列。多重标号算法可以看作是单源 Dijkstra 算法的扩展,将起点到每个节点的唯一标号过程以多重标号链表方式替代,这样,可以一次得到 $1:n$ 的 K 则最短路径。

网络转换算法通过一定规则对原始网络进行转换,构造新的网络,把求解两点间 K 则最短路径问题转换为求解新网络中以某节点为起点、任意节点为终点的前 K 条最短路径问题^[4]。在转换后的网络中可以直接用 Dijkstra 算法求解。但是该算法需要进行网络转换,实现过程复杂,且计算出的备选路径可能包含回路。

3.2 有损算法

惩罚因子算法思想是先用 Dijkstra 算法计算两点间的最短路径,作为第 1 条参考路径;然后依次用“惩罚因子”对该路径上的边进行“惩罚”,即把该路径的权重修改为原权重与惩罚因子相乘的结果,然后重新计算两点间的路径。被惩罚后的边可能在再次计算中被舍弃,因此可以得到新的参考路径。该算法的优点是只需多次运行最短路径算法,但是惩罚因子的设置有一定的主观性。

删除连接边算法通过临时删除当前路径中节点的某些连接边来求取新的备选路径^[13]。该算法通过遍历当前最短路径中的节点,按一定规则依次临时删除节点的某些连接边,并以该节点为中间节点,构造备选路径。在所有备选路径中权值最小者就是新循环中的当前路径。该算法除了多次运行最短路径算法外,还要维护路径集合,进行子路径包含、回路包含判断等操作。

背离路径算法首先以起终点间的最短路径为基础,依次把起、终点间的所有节点作为被“背离”的对象节点,构造候选路径^[9]。遍历完所有节点后,保留权值最小的前 $K-1$ 条路径。然后对求出的 $K-1$ 条最短路径分别再求解 $K-1$ 条背离路径,将上述 $(K-1)^2$ 条路径按照路径的权值大小排列,权值最小的前 $K-1$ 条路径和最短路径即为起终点之间的 K 则最短路径^[8]。

本文提出一种基于双向搜索策略的 K 则最短路径算法,以改进的 Dijkstra 最短路径算法为基础,从起点和终点同时搜索,分别构造正序和逆序最短路径树,当两棵树的节点发生交汇时,则从交汇节点开始,分别向两棵最短路径树的根回溯,构造一条备选路径。双向搜索算法的核心是双向最短路径树的构造。扩展正序树时,与节点相邻的节点为其出度节点,而逆序树中节点的相邻节点为其入度节点。

4 K 则最短路径算法时间复杂度分析

目前求解 K 则最短路径的主要算法都以 Dijkstra 算法为基础,但是它们的时间复杂度、运行效率和计算精度都不尽相同。下面对严密算法中具有代表性、并且时间复杂度相对较低的多重标号算法和候选删除边算法,及其 4 种有损算法——惩罚因子算法、删除连接边算法、背离路径算法和双向搜索算法进行时间复杂度分析。

假设网络节点数为 n , 总边数为 m , 采用四叉堆优先级队列实现单源的 Dijkstra 算法, 针对交通网络, 其时间复杂度为 $O(n \log n)$ [25]。

多重标号算法中, 每循环一次把一个节点的一个标号转化为永久标号。由于共有 Kn 个标号, 所以该循环最多进行 Kn 次。每次循环中, 判断标号的状态和选择一个路径权值最小的标号, 一共需要 $\log(Kn)$ 次比较, 对永久标记节点的每个相邻节点进行松弛时, 需要判断该路径是否包含环路, 平均需要进行 e 次判断, 保留该路径的节点, 需要进行 e 次操作, 修改节点的 K 个链表需要 K^2 次操作, 所以时间复杂度为 $Kn(\log(Kn) + D(e + e + K^2))$, 其中, D 是节点的度。由于 K 是常数, D 在交通网络中较小 (一般不超过 4), e 的上限是 m , 因此该算法的时间复杂度为 $O(Kn(\log(Kn) + m))$ 。

候选删除边算法用数组表示每条路径对应的子图的删除边集合, 在得到一条路径后, 分别删除该路径中的一条边, 再分别用 Dijkstra 算法计算起、终点在子图中的最短路径, 路径上限数为 m , 因此算法的时间复杂度为 $O(Kmn \log n)$ 。

惩罚因子算法多次调用 Dijkstra 算法, 在每次调用之前对已有路径中的边进行“惩罚”, 每次惩罚所花的平均时间为 e (起终点间最短路径的平均边数), 每次调用 Dijkstra 算法都可以得到一条路径, 但是由于可能得到重复路径, 因此需要调用的次数比要求解的路径数 K 大, 设调用次数为 L , $L \geq K$, 是常数。因此“惩罚因子”算法的时间复杂度为 $O(L(e + n \log n))$ 。 e 的上限是 m , L 和 K 相当, 因此该算法的时间复杂度为 $O(K(m + n \log n))$ 。

删除连接边算法每遍历完成一条当前路径后, 就从候选路径集合中取出权值最小的一条路径, 放入结果路径集中。每条路径的平均节点数为 $e + 1$, 所以遍历一条路径中的节点需要 $e + 1$ 次循环。对每个节点 n_i 进行如下处理: 判断该节点的哪些边需要删除, 需要 $adjoin(n_i)Ke$ 的时间, 其中 $adjoin(n_i)$ 是节点的度, 在交通网络中是一个比较小的常数, K 是所求的路径数, 也是常数, e 是路径的平均边数, 其上限是网络中的边数 m 。因此算法复杂度为 $O(Km(adjoin(n_i)Km + n \log n))$, 在交通网络中, 节点的度一般较小, 所以 $adjoin(n_i)$ 可以看成是一个较小的常数, 算法的时间复杂度简化为 $O(Km(Km + n \log n))$ 。

背离路径算法假设每个节点的平均“背离”节点数目为 u' , 求取一条路径的背离路径的时间复杂

度为 $O((e + 1)u'n \log n)$ 。一次遍历可以产生 $(K - 1)$ 条背离路径, 对这 $K - 1$ 条路径又分别遍历, 产生 $(K - 1)^2$ 条背离路径, 总共的时间复杂度为 $(K(e + 1)u'n \log n)$, 由于背离节点的平均值最大不超过节点的最大度数, 而交通网络中节点的度较小 (即为交叉口的数目, 一般不超过 4), 而 e 上限是 m , K 为常数, 因此“背离路径”算法的时间复杂度为 $O(Kmn \log n)$ 。

双向搜索算法构造正序和逆序最短路径树的时间复杂度为 $O(n \log n)$, 标记节点和修改节点信息花费常数时间, 判断重复路径需要 $O(K^2e)$ 的时间, 判断回路的时间是 $O(Ke)$, 其中 K 是求解的备选路径数, e 是路径的平均边数, 其上限是 m , 所以双向搜索算法的时间复杂度为 $O(n \log n + K^2m)$ 。

5 K 则最短路径算法效率与精度分析

最短路径算法时间复杂度与实际运行效率并不总是一致 [1]。为了测试算法针对实际路网的运行效率和精度, 编程实现了上述 6 种算法。为保证算法比较的一致性, 对各算法的实现基于一致的数据组织结构和存取方法。弧段对象和拓扑关系对象数据结构如表 1 和表 2 所示。此外, 笔者实现的各种 K 则最短路径算法中所调用的 1:1 最短路径算法都是基于四叉堆的 Dijkstra 算法。在多重标号算法中, 也采用四叉堆对临时标号的节点进行组织和管理。

表 1 弧段对象数据结构

Tab. 1 Data structure of arc objects

名称	类型	含义
<i>ID</i>	Int	弧段唯一标识
<i>Fnode</i>	Int	起始节点 <i>ID</i>
<i>Tnode</i>	Int	终点节点 <i>ID</i>
<i>Imp</i>	Double	弧段阻抗

表 2 拓扑关系对象数据结构

Tab. 2 Data structure of topological objects

名称	类型	含义
<i>ID</i>	Int	节点唯一标识
<i>AdjNum</i>	Int	与该节点连接的节点数
<i>AdjArcList</i>	Int 列表	与该节点连接的弧段 <i>ID</i> 列表
<i>AdjNodList</i>	Int 列表	与该节点连接的节点 <i>ID</i> 列表

在编程实现各种 K 则最短路径算法后, 采用北京市部分路网进行了测试。路网包含 1 738 个节点和 2 415 条边, 采用系统抽样的方法抽取了均匀分

布的 200 个节点(如图 3 所示),构成了 100 对“起—终点”节点对。分别采用以上 6 种 K 则最短路径算法计算这些节点对之间的前 5 条最短路径。所有算法均用 Visual C ++6 编写,在 P4 3.2 G,内存 1 G 的台式机上运行。记录各算法计算一次 K 则最短路径(K=5)平均耗费的 CPU 时间,并计算平均精度。精度计算如下式

$$a = 1 - \text{abs}(C_i - R_i) / R_i$$

其中, C_i 是算法得到的第 i 短路径长度, R_i 是理论上第 i 短路径长度。算法的总精度为 5 条路径精度的平均值。实验结果如表 3 和表 4 所示。

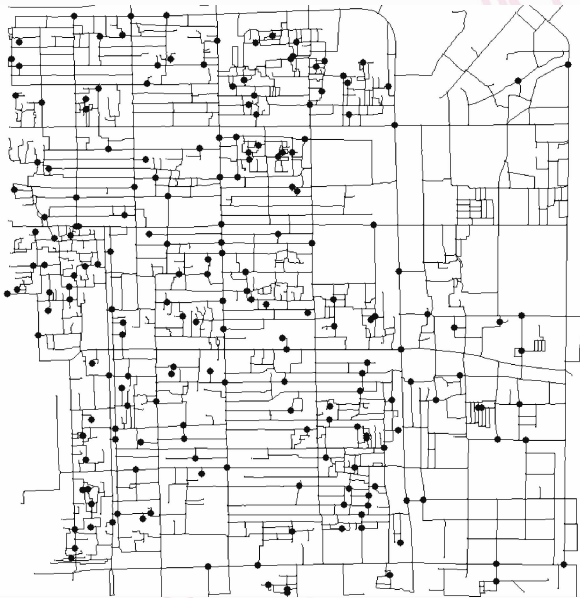


图 3 实验道路网络
Fig. 3 Pilot road network

表 3 K 则最短路径算法运行时间比较

Tab. 3 Efficiency comparison of the Kth shortest path algorithms

	多重标号	候选删除边	惩罚因子	删除连接边	背离路径	双向搜索
平均时间 (ms)	250.1	1 931.7	3.7	185.9	351.9	12.8

表 4 有损 K 则最短路径算法精度比较

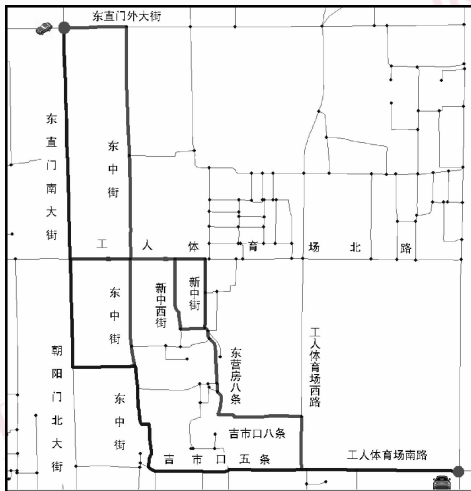
Tab. 4 Accuracy comparison of the lossy Kth shortest path algorithms

	惩罚因子	删除连接边	背离路径	双向搜索
平均精度 (%)	84.16	99.28	98.52	98.11

从实验结果可以看出,多重标号算法和候选删除边算法作为理论严密的 K 则最优路径算法,虽然没有精度损失,但是运行效率相对较低。在有损算法中,惩罚因子算法和双向搜索算法的效率远高于其他算法。但惩罚因子算法的平均精度太低,而其他有损算法的精度都在 98% 以上。尤其是双向搜索算法,仅以不到 2% 的精度损失,换取了运行效率 20 倍的提高,是极具应用价值的 K 则最短路径算法。

6 实例分析

分别采用多重标号算法和双向搜索算法求解东直门地铁站到体育场中学之间的前 5 条最优路径。多重标号算法耗时 235 ms,双向搜索算法仅耗时 15 ms。路径计算结果如图 4 和图 5 所示。



- 路径1: 途经东直门南大街、工体北路、新中西街、东营房八条、吉市口八条、工体西路、工体南路,长度: 2 921米;
- 路径2: 途经东直门南大街、朝阳门北大街、东中街、吉市口五条、工体南路,长度2 925米;
- 路径3: 途经东直门南大街、工体北路、新中西街、东营房八条、吉市口八条、工体西路、工体南路,长度: 2 932米;
- 路径4: 途经东直门外大街、东中街、工体北路、新中西街、东营房八条、吉市口八条、工体西路、工体南路,长度: 2 934米;
- 路径5: 途经东直门南大街、工体北路、东中街、吉市口五条、工体南路,长度2 936米

图 4 多重标记算法 K 则最短路径计算结果

Fig. 4 The Kth shortest paths obtained by multi-label algorithm

- 6 Santos J L K. Shortest Path Algorithms[EB/OL]. <http://www.dis.uniroma1.it/~challenge9/papers/santos.pdf>, 2006-08.
- 7 Hershberger J, Maxel M, Suri S. Finding the K Shortest Simple Paths: A New Algorithm and its Implementation [EB/OL]. <http://www.siam.org/meetings/alensex03/Abstracts/jhershberger.pdf>, 2003-01.
- 8 Dai Shu-gui, Chen Wen-lan. A practical algorithm for the K shortest-path problem [J]. Computer Engineering and Applications, 2005, **41**(36):63-65. [戴树贵, 陈文兰. 一个求解 K 短路径实用算法, 计算机工程与应用, 2005, **41**(36):63-65.]
- 9 Martins E Q, Pascoal M M, Santos J L. Deviation algorithms for ranking shortest paths [J]. International Journal of Foundations of Computer Science, 1999, **10**(3):247-261.
- 10 Dreyfus S E. An appraisal of some shortest path algorithms [J]. Operations Research, 1969, **17**(3):395-412.
- 11 Victor Manuel, Jiménez Pelayo, Andrés Marzal Varó. A Lazy Version of Eppstein's K Shortest Paths Algorithm [EB/OL]. <http://fano.ics.uci.edu/cites/Document/A-lazy-version-of-Eppsteins-K-shortest-paths-algorithm.html>, 2003.
- 12 Wang Feng, You Zhi-sheng. Application of Dijkstra and Dijkstra-based N-shortest-paths algorithm to intelligent transportation systems [J]. Application Research of Computers, 2006, **42**(9):203-208. [王峰, 游志胜. Dijkstra 及基于 Dijkstra 的前 N 条最短路径算法在智能交通系统中的应用 [J]. 计算机应用研究, 2006, **42**(9):203-208.]
- 13 Yuan Hong-tao, Zhu Mei-zheng. A fast algorithm and its implementation of finding the K-Shortest paths [J]. Computer Engineering and Applications, 2004, **40**(6):51-53. [袁红涛, 朱美正. K 优路径的一种求解算法和实现 [J]. 计算机工程与应用, 2004, **40**(6):51-53.]
- 14 Lawler E L. A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem [J]. Management Science, 1972, **18**(7):401-405.
- 15 Katoh N, Ibaraki T, Mine H. An efficient algorithm for K shortest simple paths [J]. Networks, 1982, **12**(4):411-427.
- 16 Yen J Y. Finding the K shortest loopless paths in a network [J]. Management Science, 1971, **17**(11):712-716.
- 17 Martins E Q. An algorithm for ranking paths that may contain cycles [J]. European Journal of Operational Research, 1984, **18**(1):123-130.
- 18 Azevedo J A, Martins E Q V. An algorithm for the ranking of shortest paths [J]. European Journal of Operational Research, 1993, **69**(1):97-106.
- 19 Chong E I, Maddila S R, Morley S T. On Finding Single-source Single-destination K Shortest Paths [EB/OL]. <http://fano.ics.uci.edu/cites/Document/On-finding-single-source-single-destination-K-shortest-paths.html>, 1995-07.
- 20 Miaou S P, Chin S M. Computing K-shortest path for nuclear spent fuel highway transportation [J]. European Journal of Operational Research, 1991, **53**(1):64-80.
- 21 Shier D R. On algorithms for finding the K shortest paths in a network [J]. Networks, 1979, **9**(3):195-214.
- 22 Juliana Castillo. Analysis and Implementation of K-shortest Path Algorithms in Geographic Information System [D]. Dallas, TX, USA: University of Texas at Dallas, 2005.
- 23 Hershberger J, Suri S. Vickrey prices and shortest paths: What is an edge worth [A]. In: Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science [C], Las Vegas, Nevada, USA, 2001:252-259.
- 24 Lu Feng, Lu Dong-mei, Cui Wei-hong. Improved dijkstra algorithm based on quad-heap priority queue and inverse adjacent list [J]. Journal of Image and Graphics, 1999, **4A**(12):1039-1045. [陆锋, 卢冬梅, 崔伟宏. 基于四叉堆优先级队列及逆邻接表的改进型 Dijkstra 最短路径算法 [J]. 中国图象图形学报, 1999, **4A**(12):1039-1045.]