文章编号:1671-8879(2014)03-0133-04

交通网络 k-短路径与最小支撑树问题

陈京荣[1],俞建宁[1],李引珍[2]
(1. 兰州交通大学 数理学院,甘肃 兰州 730070;2. 兰州交通大学 交通运输学院,甘肃 兰州 730070)

摘　要:为了给交通管理部门提供多个路径诱导信息,基于经典的最短路径算法——Dijkstra 算法,研究了赋权交通网络的 k-短路径问题。k-短路径问题是在网络 G 中求出给定起讫点对之间的 k 条路径 $P_1,P_2,\cdots,P_k$,满足 $W(P_1)\leqslant W(P_2)\leqslant\cdots\leqslant W(P_k)$,其中 $W(*)$ 表示路径 $*$ 的权值。在网络 G 的基础上,通过对 G 的点、边重新划分以及对边上的权值重新赋值,构造出了 1 个新的网络 G' 并讨论了它的几个性质。从而将 G 的 k-短路径问题转换为求解 G' 的最小支撑树问题,进一步,最小支撑树问题又等价于求 G' 中一条边的权值。研究结果表明:由于最小支撑树问题具有多项式算法,得到关于 k-短路径问题的多项式算法,其时间复杂性为 $O(k(m+n\lg(n)))$,m 和 n 为 G 的边数和顶点数。最后通过算例给出了算法的具体执行过程,同时验证了其可行性。

关键词:交通工程;交通网络;k-短路径;最小支撑树;Dijkstra 算法
中图分类号:U116.2　　　**文献标志码:**A

K-shortest paths and minimum spanning tree in traffic network

CHEN Jing-rong[1], YU Jian-ning[1], LI Yin-zhen[2]
(1. School of Mathematics and Physics, Lanzhou Jiaotong University, Lanzhou 730070, Gansu, China;
2. School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, Gansu, China)

Abstract: In order to provide route guidance information to traffic departments, k-shortest paths were studied based on Dijkstra algorithm. The k-shortest paths problem is to get k paths $P_1, P_2, \cdots, P_k$ for a given pair of points and meet the criteria $W(P_1)\leqslant W(P_2)\leqslant\cdots\leqslant W(P_k)$, where $W(*)$ denotes the weight of path $*$. By partitioning the vertex set and the edge set of G, and reassigning the weight for each edge, a new network G' was constructed from G. Then the k-shortest paths problem of G was converted into the minimum spanning tree (MST) problem of G'. MST problem became equal to computing an edge's weight in G'. The results show that based on the polynomial algorithm of MST problem, an polynomial algorithm for the k-shortest paths problem is obtained and the time complexity is $O(k(m+n\log(n)))$, where m and n denote the number of edges and nodes of G respectively. In the end, an illustrative example verifies the concrete progress and the feasibility of the algorithm. 4 figs, 12 refs.

Key words: traffic engineering; traffic network; k-shortest path; minimum spanning tree; Dijkstra algorithm

收稿日期:2013-05-16
基金项目:国家自然科学基金项目(61164003);教育部博士点基金项目(20136204120007);甘肃省自然科学基金项目(1308RJZA128)
作者简介:陈京荣(1975-),女,甘肃榆中人,教授,工学博士,E-mail:chenjr@mail.lzjtu.cn。

0 引 言

最短路径问题是许多领域内经常出现的一类问题,在交通运输和网络通信等领域的研究中所遇到的许多问题经转化后都可以归结为最短路径问题。传统的最短路径问题是在有向图中求解从 1 个顶点到其他各顶点的最短路径。而实际上,在一些决策支持系统和咨询系统中,用户除了希望得到最短路径以外,还希望得到次短和第 3 短路径等作为决策参考,由此产生了求解 k-短路径的问题。传统的求解 k-短路径的方法是:先求解第 $k-1$ 短路径。然后通过求解第 $k-1$ 短路径的 1 个最小背离来求得第 k 短路径。算法的时间复杂度较大(最坏情况下为 $O(n^{k-1})$),在具有较多顶点数的网络中,该算法并没有多少实际应用价值。虽然近年来也出现了一些解决这一问题的算法,但这些算法本身比较复杂难以编程实现,而且时间复杂度不够理想。

关于最短路等问题,Dijsktra 给出了 1 个时间复杂度为 $O(m+n\lg(n))$ 的算法,其中 m 和 n 为给定网络的边数和顶点数[1]。k-短路径问题最早于 1959 年出现在 Hoffman 等的研究中[2];Yang 等提出了 1 个具有时间窗的 k-短路径的多项式算法[3-4];Carlyle 等得到了 1 个时间复杂性劣于一些算法的多项式算法,但是其空间复杂性 $O(m)$ 是最好的[5];戴树贵等给出了时间复杂性为 $O(n^2)$ 的算法,用以获得次短路径和渐次短路径[6];Antonio 讨论了点对点的 k-短路径的算法,其空间复杂性为 $O(m)$,时间复杂性为 $O(knf(n,m,C_{\max}))$[7];Zijpp 等获得了具有多个约束条件下的可行最短路径的方法[8];关于稀疏图,Bhosle 讨论了最短路径中边删除后的替代路径问题,给出了近似优化算法[9];关于 k-短路径问题的一些变形讨论,可以参考文献[10-12]。

本文基于经典的最短路径算法——Dijkstra 算法,给出了 1 个求解网络。k-短路径的算法,通过构造新的网络图以及证明相关性质,并借助于最小支撑树,给出了时间复杂度和空间复杂度较好的求解算法,并通过算例验证了算法的可行性。

1 预备知识

设有交通网络 $G=(V(G),E(G))$ 为简单图,其中 $V(G)$ 为顶点集,$E(G)$ 为边集,并且 $|V(G)|=n$,$|E(G)|=m$。对于 G 的每条边 e 都有 1 个权值 $\omega(e)$。为了讨论的需要,假设 G 是 k-边连通的,即在 G 中删除 $k-1$ 条边 G 仍然连通,但是删除 k 条边后 G 有可能不连通了。

考虑 G 中任意不同的两点 s 与 t,令 $P_G(s,t)=v_0(=s)e_1v_2e_2\cdots e_lv_l(=t)$ 为 G 中 s 到 t 的最短路,它的长度记为 $d_G(s,t)=\sum_{i=1}^{l}\omega(e_i)$。$G$ 的最小生成树(MST)是指 G 的所有生成树中权值最小的那个生成树,用 W_G(MST)表示 G 的 MST 的权值。对于一个给定的顶点 v,从 v 到其他各点的最短路径形成图 G 的一棵生成树,记为 T_v,并把它称作以 v 为根的最短路生成树,简称最短树。假设 V' 是 $V(G)$ 的 1 个非空子集,以 V' 为顶点集,以两端点均在 V' 中的边的全体作为边集所组成的子图,称为 G 的由 V' 导出的子图,记为 $G[V']$,$G[V']$ 称为 G 的导出子图。用 V_s 和 V_t 表示在 T_s 中删去 $P_G(s,t)$ 的一条边后所得的 2 个导出子图的顶点集合,且有 $s\in V_s$ 和 $t\in V_t$。$E_G(V_s,V_t)$ 表示 1 个端点在 V_s 中,另 1 个在 V_t 中的边的集合。

对给定顶点 s 和 t,k-短路径问题就是在 G 中要求出 s 到 t 的 k 条路 $P_1,P_2,\cdots,P_k$,满足 $W(P_1)\leqslant W(P_2)\leqslant\cdots\leqslant W(P_k)$。

2 性质和算法

根据 $P_G(s,t)$ 和 T_s 的定义,有以下性质。

性质 1:$P_G(s,t)\subset T_s$,即 s 到 t 的最短路径包含在以 s 为根的最短树中。

引理 1[10]:对任意 $e\in P_G(s,t)$,删除 e 后 s 到 t 的最短路恰好使用了 $E_G(V_s,V_t)$ 中的一条边,并且新的最短路其权值为

$$\min_{uv\in E_G(V_s,V_t)}\{d_G(s,u)+\omega_G(uv)+d_G(u,t)\}$$

根据 G 来构造 G',步骤如下。

(1) 在 T_s 中依次删除 $P_G(s,t)$ 的 l 条边,则 $V(G)$ 被分成了 $l+1$ 个子集,按顺序记为 $V_0,V_1,\cdots,V_l$,同时得到了 $l+1$ 个导出子图 $G(V_i)$,且有 $v_i\in V_i(i=0,1,\cdots,l)$;

(2) 对 $E(G)$ 作如下划分为

$$E(G)=\bigcup_{i=0}^{l}E(G(V_i))\cup\bigcup_{i\neq j}E(V_i,V_j)$$

(3)令 $V(G')=V(G)$,$E(G')=E(G)$,并且对任意 $e=uv\in E(G')$,如果

$$e\in P_G(s,t)\cup\bigcup_{i=0}^{l}E(G(V_i))$$

则 $\omega_{G'}(e)=0$;否则

$$\omega_{G'}(e)=d_G(s,u)+\omega_G(uv)+d_G(u,t)$$

性质 2:在 G' 中,T_s 的边构成 G' 的最小支撑树和以 s 为根的最短树。

证明:由权值的重新定义,G' 中 T_s 的边的权值都为 0。

性质 3：对 $P_G(s,t)$ 中任意一条边 e，$G'-e$ 的最小支撑树和 $G-e$ 中 s 到 t 的最短路径有相同的权值。

证明：根据性质 2，G' 的 MST 的权值在没有删除 e 时为 0，删除后得 V_s 和 V_t。由树的定义知，e 的替代边 e^* 必然不属于 $G[V_s]$ 或 $G[V_t]$，否则会形成闭回路，所以 $e^* \in E_{G'}(V_s,V_t)$。则 $G'-e$ 的 MST 的权值为替代边的权值，同时由引理 1 得

$$W_{G'-e}(\mathrm{MST})=\omega_{G'}(e^*)=\min_{uv\in E_{G'}(V_s,V_t)}\{d_{G'}(s,u)+\omega_{G'}(uv)+d_{G'}(u,t)\}$$

由 G' 权值的定义 $d_{G'}(s,u)=d_{G'}(u,t)=0$，则

$$W_{G'-e}(\mathrm{MST})=\min_{uv\in E_{G'}(V_s,V_t)}\{\omega_{G'}(uv)\}$$

$$\omega_{G'}(uv)=d_G(s,u)+\omega_G(uv)+d_G(u,t)$$

所以

$$W_{G'-e}(\mathrm{MST})=\min_{uv\in E_G(V_s,V_t)}\{d_G(s,u)+\omega_G(uv)+d_G(u,t)\}$$

推论 4：对 $P_G(s,t)$ 中任意一条边 e，$G'-e$ 的 MST 的权值就是替代边 e^* 的权值。

推论 5：对 $P_G(s,t)$ 中任意一条边 e，删除后替代边 e^* 的权值为

$$\omega_{G'}(e^*)=\min_{uv\in E_{G'}(V_s,V_t)}\{\omega_{G'}(uv)\}$$

上面的几个性质为求解 k-短路径问题提供了 1 个非常简洁的思想，下面给出具体算法。

(1) 用 Dijkstra 算法得出以 s 和 t 分别为根的最短树 T_s 和 T_t，以及 s 到 t 的 j-短路；

(2) 利用给出的方法构造新图 G'；

(3) 取 $P_G(s,t)$ 中的一条边 e_i，在 T_s 中删除 e_i 得 V_s 和 V_t。计算 $G'-e$ 的最小支撑树的权值为

$$W_{G'-e_i}(\mathrm{MST})=\omega_{G'-e_i}(e^*)=\min\{\omega_{G'}(uv)\mid u\in V_s, v\in V_t\}$$

注意到删除的边不同时，顶点集合 V_s 和 V_t 分别是不同的，因为每次循环完后，前一次的内容不再使用，可以重新赋值，故采用一样的记法；

(4) 如果 $1<i\leqslant l$，返回到(3)。最后有 $\omega_{G'-e_1}(e_1^*)$，$\omega_{G'-e_2}(e_2^*)$，…，$\omega_{G'-e_l}(e_l^*)$，计算 $\omega_{G'-e'}(e^*)=\max\{\omega_{G'-e_i}(e_i^*)\mid i=1,2,\cdots,l\}$。则在 T_s-e' 中，e^* 是 e' 的替代边，并且得到 s 到 t 的 $j+1$-短路 P_{j+1}。

(5) 令 $G=G-e'$，返回到(1)。

现对算法复杂性进行分析。求以 s 和 t 分别为根的最短树 T_s 和 T_t，其计算量为 $O(m+n\lg(n))$，G' 的构造在线性时间内就可以完成，因此算法的时间复杂性为 $O(k(m+n\lg(n)))$。

3 算例

设图 1 为某区域的简化交通网络图 G，s 和 t 为 G 中的两点，下面利用上述算法来计算 s 到 t 的 4-短路径。

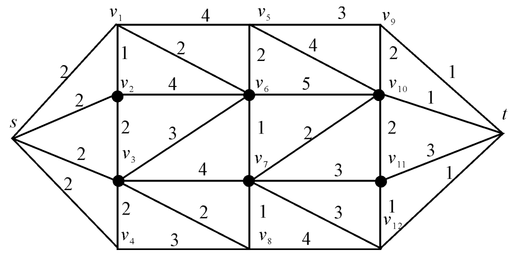

图 1　交通网络图 G

Fig. 1　Traffic network G

(1) 用 Dijkstra 算法得出 T_s，T_t 和 $P_G(s,t)=se_1v_1e_2v_6e_3v_7e_4v_{10}e_5t$，如图 2 和图 3 所示。$P_G(s,t)$ 在图 2 中用粗线表示，顶点旁边的数字表示 s 或 t 到该顶点的距离；

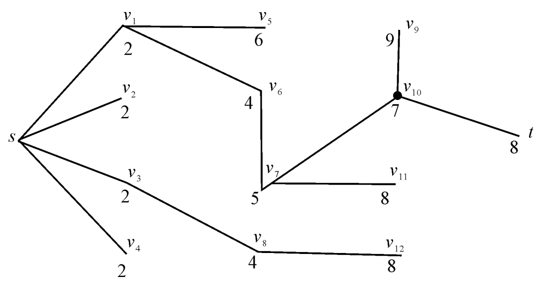

图 2　T_s 和 $P_G(s,t)$

Fig. 2　T_s and $P_G(s,t)$

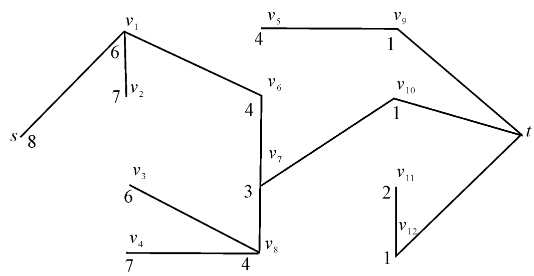

图 3　T_t

Fig. 3　T_t

(2) 在 T_s 中删除 $P_G(s,t)$ 的所有边，则可得到 V_i 和 $G[V_i]$ $(i=0,1,\cdots,5)$；

(3)由 G' 的定义，$P_G(s,t)$ 和 $G[V_i]$ $(i=0,1,\cdots,5)$ 中边的权值都为 0，所以在 G' 中不再划出，如图 4 所示。下面计算其他边的权值，首先是删除 e_1 后导出子图 $G[V_s]$ 和 $G[V_t]$ 之间的边

$$\omega_{G'}(v_1v_2)=d_G(s,v_2)+\omega_G(v_1v_2)+d_G(v_1,t)=2+1+6=9$$

同理可得

$\omega_{G'}(v_2v_6)=10$，$\omega_{G'}(v_3v_6)=9$，$\omega_{G'}(v_3v_7)=9$，$\omega_{G'}(v_8v_7)=8$，$\omega_{G'}(v_{12}v_7)=14$，$\omega_{G'}(v_{12}v_{11})=11$，$\omega_{G'}(v_{12}t)=9$。

类似得出删除 e_2, e_3, e_4, e_5 后相应边的权值,如图 4 所示。

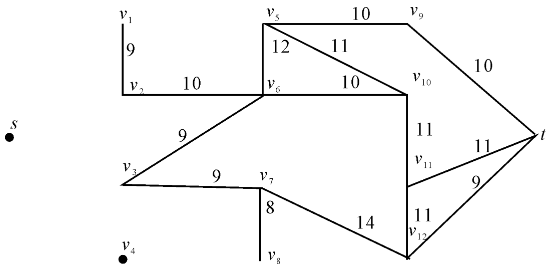

图 4 新网络图 G'

Fig. 4 New network G'

(4) 在 T_s 中删除 e_1 后,有顶点集 $V_s = \{s, v_2, v_3, v_4, v_8, v_{12}\}$ 和 $V_t = \{v_1, v_5, v_6, v_7, v_{10}, v_{11}, t\}$ 。

则

$$W_{G'-e_1}(\mathrm{MST}) = \omega_{G'-e_1}(e_1^*) = \min\{\omega_{G'-e_1}(uv) \mid u \in V_s, v \in V_t\} = \min\{\omega_{G'}(v_2v_1), \omega_{G'}(v_2v_6), \omega_{G'}(v_3v_6), \omega_{G'}(v_3v_7), \omega_{G'}(v_8v_7), \omega_{G'}(v_{12}v_7), \omega_{G'}(v_{12}v_{11}), \omega_{G'}(v_{12}t)\}$$

同理可得

$$\omega_{G'-e_2}(e_2^*) = 8,\ \omega_{G'-e_3}(e_3^*) = 8,\ \omega_{G'-e_4}(e_4^*) = 9,\ \omega_{G'-e_5}(e_5^*) = 9;$$

(5)计算

$$\omega_{G'-e'}(e^*) = \max\{\omega_{G'-e_i}(e_i^*) \mid i = 1, 2, \cdots, 5\} = \max\{8, 8, 8, 9, 9\} = 9$$

则在 T_s 中删除边 v_7v_{10} 或 $v_{10}t$ 后相应的替代边为 $v_{12}t$,$P_2 = sv_1v_6v_7v_{12}t$ 是 s 到 t 的 2-短路径;

(6) 重复以上步骤,得 s 到 t 的 4-短路径为 $P_1 = sv_1v_6v_7v_{10}t$, $P_2 = sv_1v_6v_7v_{12}t$, $P_3 = sv_1v_6v_{10}t$, $P_4 = sv_3v_6v_{10}t$。

4 结 语

(1)通过将交通网络的 k-短路径问题转换为求解最小支撑树问题,得到了时间复杂性为多项式的算法,为交通管理部门进行决策与发布诱导信息提供依据。

(2)构造的新网络 G' 以及证明的相关性质,简化了问题的难度,是算法设计的基础。

(3)以具有 14 个节点的简化网络为算例,对给定的一对起讫点,详细实现了算法的的执行过程,得到了 4-短路径。如果交通部门将此信息发布,出行者一般情况下不会全去选择路径 P_1,否则就会发生交通拥堵,这样提供多种路径选择,提高了交通网络的通行能力。关于管理者如何发布信息及出行者如何选择路径,以及如何平衡这两者之间的利益,将是下一步研究的一个问题。

(4)文中只考虑了最短路径上一条边中断后对路径的选择,但是在实际中会有多条边或多个点或点边同时中断的情形,这个问题将继续研究。

参考文献:

References:

[1] Dijkstra E W. A note on two problem in connexion with graphs[J]. Numerische Mathematik, 1959(1): 269-271.

[2] Hoffman W, Pavley R. A Method for the solution of the N th best path problem[J]. Journal of the ACM, 1959, 6(4): 506-514.

[3] Yang H H, Chen Y L. Finding *k* shortest looping paths with waiting time in a time-window network [J]. Applied Mathematical Modelling, 2006, 30(5): 458-465.

[4] Chen Y L, Yang H H. Finding the first *k* shortest paths in a time-window network[J]. Computers & Operations Research, 2004, 31(4): 499-513.

[5] Carlyle M W, Wood K R. Near-shortest and *k*-shortest simple paths[J]. Networks, 2005, 46(2): 98-109.

[6] 戴树贵,陈文兰. 一个求解 *k* 短路径实用算法[J]. 计算机工程与应用, 2005(36): 63-65.
DAI Shu-gui, CHEN Wen-lan. A practical algorithm for the *k* shortest path problem[J]. Computer Engineering and Applications, 2005(36): 63-65. (in Chinese)

[7] Antonio S N. An efficient time and space *k* point-to-point shortest simple paths algorithm[J]. Applied Mathematics and Computation, 2012, 218(2): 10244-10257.

[8] Zijpp N J, Fiorenzo S. Path enumeration by finding the constrained *k*-shortest paths[J]. Transportation Research Part B: Methodological, 2005, 39(6): 545-563.

[9] Bhosle A M. Improved algorithms for replacement paths problems in restricted graphs[J]. Operations Research Letters, 2005, 33(5): 459-466.

[10] Guerriero F, Musmanno R. Parallel asynchronous algorithms for the k shortest paths problem[J]. Journal of Optimization Theory and Applications, 2000, 104(1): 91-108.

[11] Feixiong L, Theo A, Harry T. Multi-state super networks: recent progress and prospects[J]. Journal of Traffic and Transportation Engineering: English Edition, 2014, 1(15): 13-28.

[12] Ruppert E. Finding the *k* shortest paths in parallel [J]. Algorithmica, 2000, 28(2): 242-254.