时序最短路径算法

邓冬梅 王冠楠 朱 建 高 辉 陈端兵
(电子科技大学计算机科学与工程学院 成都 611731)

摘 要 最短路径是指网络中两结点间阻碍强度最小的一条路径。传统的最短路径是在静态网络上进行研究的，然而现实生活中很多网络是动态的、有时序性的，因此传统的最短路径算法并不能用于解决所有最短路径问题。为了寻找时序网络上的最短路径，在 Dijkstra 算法思想基础上，提出一种时序最短路径的精确算法。文中利用严格的数学推导证明了本算法的可行性，并通过对构建的网络做实证分析验证了算法的正确性。

关键词 时序网络，最短路径，Dijkstra 算法

中图法分类号 TP312 **文献标识码** A

Temporal Shortest Path Algorithm

DENG Dong-mei WANG Guan-nan ZHU Jian GAO Hui CHEN Duan-bing
(Computer Science and Engineering Institute, University of Electronic Science and Technology, Chengdu 611731, China)

Abstract The shortest path is a path which has the least hinder strength between two nodes in the specified network. Traditionally, it is studied on static network, but many networks are dynamic and temporal in real life. So the traditional algorithms can't solve all problems about shortest path. This paper presented a precision algorithm to find the temporal shortest path based on the Dijkstra's algorithm. It proved the correctness of the algorithm by mathematical derivation and verified the feasibility of the algorithm on a constructed network.

Keywords Temporal network, Shortest path, Dijkstra's algorithm

1 绪论

最短路径应用在生活中随处可见，如出门旅游选择最短路程问题、信息在路由中传输的最佳路径问题、GPS 导航应用等等。从网络模型的角度看，最短路径分析就是在指定网络中的两结点间找出一条阻碍强度最小的路径，根据阻碍强度的不同定义，最短路径可以指一般意义上的距离最短，也可以引申到其它的量度，如时间、费用、油耗等[1,2]。针对最短路径问题，多年来产生了大量相关领域的研究成果。

1.1 研究背景与现状

最短路径的研究贯穿于运筹学、计算机科学、图论学、数学科学、地理信息科学、交通运输等领域[3-7]。可以将现实生活中很多问题抽象成网络，然后通过计算其最短路径进行解决。如百度、google 公交路线查询系统，煤矿等危险工作的应急系统，游戏中最佳路径的选择，计算机网络中信息流在路由器间的最佳传输问题以及 GPS 导航系统等。因此，针对网络中最短路径的有效计算问题研究具有重要的理论和现实意义。多年来，图论与不断发展完善的计算机数据结构及算法的有效结合使得新的最短路径算法不断涌现。目前已有的最短路径算法有 Dijkstra、Floyd 等经典算法以及启发式 A* 算法、CDZ 算法等近似算法[8-12]。但是这些算法都只适用于静态网络，并不能用于解决所有问题。

很多必须放在时序网络中进行研究的问题都不能借助传统的最短路径算法进行计算，比如：在选择公交路线时，一般有两个选择，一个是最少换乘，另一个是花费时间最少。每路公交车在一天中有 N 班次，如果将公交车路线构建成一个静态网络(即只要从一个地点到另一个地点有公交车到达，则认为这两地点之间有连边)，并不能计算出一个点到另一点所需的最少时间。每路公交车在每两个地点之间运行是有时间限制的，换乘公交车必须满足前面乘坐的车的到达时间小于后面要乘坐车的起始时间。也就是说无论是找最少换乘还是最快达到，都必须考虑每路公交车的运行时间段和换乘车次的时间先后顺序等。

对这种事件发生有时间限制和先后次序的网络，传统的最短路径算法是不适合的，而现实生活中大部分网络都是时序网络，因此时序网络中的最短时序路径的寻找是急需解决的问题。但是由于时序网络本身的时间限制和次序关系，最短时序路径求解就变得复杂。

文献[13]中作者 Petter 给出了计算时序网络中节点对之间平均时序距离的算法。算法思想如下：取一个观察窗口 T，在这个观察窗口内，若边(i,j)上有 n 次事件发生(即 i 与 j 接触)，n 次事件起始时间分别为 $t_1,t_2,\cdots,t_n$，持续时间分别为 $\Delta t_1,\Delta t_2,\cdots,\Delta t_n$，则节点 i 到 j 的平均时序距离定义为：

到稿日期：2013-07-21 返修日期：2013-11-10

邓冬梅(1987—)，女，硕士生，主要研究方向为数据挖掘，E-mail：851636104@qq.com；王冠楠(1989—)，男，硕士生，主要研究方向为数据挖掘；朱 建(1981—)，男，硕士生，主要研究方向为智能计算。

$$\tau_{ij}=\frac{1}{T}\Big[t_1\Big(\frac{t_1}{2}+\Delta t_1\Big)+(t_2-t_1)\Big(\frac{t_2-t_1}{2}+\Delta t_2\Big)+\cdots+(t_n-t_{n-1})\Big(\frac{t_n-t_{n-1}}{2}+\Delta t_n\Big)+(T-t_n)\Big(\frac{T-t_n}{2}+t_1+\Delta t_1\Big)\Big] \quad (1)$$

但是此算法计算的是节点间在某个特定的观察窗口内的平均时序距离，而不是节点间的最短时序距离。到目前为止，还没有公开文献提出计算时序网络中最短路径的有效方法。本文在经典算法 Dijkstra 的思想基础上，提出一种计算时序网络中最短时序路径的精确算法。

2 模型描述

实际网络中，如果节点间存在物理链接，则节点间能通过物理链接进行信息交互（比如：人与人之间的通话、论文作者之间的合作等）。在将实际网络抽象成网络图的过程中，为了简单起见，只有存在信息交互的节点之间才构建边，不存在信息交互的节点之间不构建边。构建静态网络图时，在整个观察期间，如果两节点存在一次以上交互，则此节点对之间就存在边。网络图表示为 $G=(V,E)$，其中 V 表示网络中节点集合，E 为网络中边集合。

时序网络是在静态网络的基础上考虑了节点间交互的时间。在时序网络中，节点间的一次交互称为一个事件，每条边只在有事件发生的时间段是活跃的，每条边上可以有多个事件，信息在边上进行传递必须满足一定的先后时间关系。

定义 1(事件)　时序网络中，如果节点 s 与节点 u 在时间段$[t_s,t_f]$有信息交互，则称节点对 s 与 u 在时间段$[t_s,t_f]$有事件发生。每个事件 $event_i$ 用四元组 $e_i=(s,u,t_s,t_f)$代替，其中，$e_i.s$、$e_i.u$ 分别为事件源节点和事件目标节点（即信息从 s 流向 u），$e_i.t_s$、$e_i.t_f$ 分别为事件的起始时间和终止时间，且 $e_i.t_s\leqslant e_i.t_f$。如果事件是瞬时的或者持续时间是可以忽略的，事件用三元组 $e_i=(s,u,t_s)$代替。网络中所有事件用 E 表示，$E=\bigcup_i\{e_i\}$。

定义 2(时序网络图)　时序网络图用 $G=(V,E)$表示，其中 V 表示网络中节点集合，E 为网络中事件(Events)集合。节点 s 与 u 之间的边用 s 与 u 之间事件的集合表示，记为 $l_{s,u}=\{e_i=(s,u,t_s,t_f)\,|\,e_i\in E\}$，边的集合用 L(Links)表示，$L=\{l_{i,j}\,|\,i,j\in V,i\neq j\}$。

定义 3(有效路径)　不同于静态网络，时序网络中边之间不一定存在传递性，信息必须通过有效时序通路才能进行传播[17,18]。节点 s 与 u 之间存在有效通路是指，在从节点 s 经过事件$e_0,e_1\cdots e_{n-1},e_n$ 到达节点u 的过程中，必须满足条件 $e_{i+1}.t_s\geqslant e_i.t_f\wedge 0\leqslant i\leqslant n-1,n\geqslant 0$。一般情况下，节点之间存在多条有效通路。若用 $p(s,u)$表示 s 与 u 之间的通路，$\mu(s,u)$表示 s 与 u 之间的路径，则节点 s 与 u 之间的第 i 条有效通路为 $p_i(s,u)=(v_{i0},v_{i1},\cdots,v_{i(n-1)},v_{in})$，其中 $v_{i0}=s$，$v_{in}=u$。对应的有效路径表示为 $\mu_i(s,u)=(e_{i0},e_{i1},\cdots,e_{i(n-1)},e_{in})$，其中 $e_{ik}\in l_{v_{ik},v_{i(k+1)}}$，$e_{ik}.t_f\leqslant e_{i(k+1)}.t_s\wedge 0\leqslant k\leqslant n-1,n\geqslant 0$。

例如时序网络（有向）示意图 1，节点集合 $V=\{A,B,C,D\}$，箭头方向为信息流动方向，其中节点对之间每个(x,y)序对代表此节点对间存在 x 时刻发生、y 时刻结束的事件。边 AB 上有 2 个事件，分别为 $e_1=(A,B,2,6)$、$e_2=(A,B,7,8)$，AB 边为 $l_{A,B}=\{e_1,e_2\}$。如果一个信息在时刻 $t=6$ 由事件 e_1 从 A 传到 B，但 BC 边上事件发生的最迟起始时间为 3，则 A 与 C 之间不存在有效通路，边 AB 和 BC 不存在传递性，B 无法将信息传给 C。

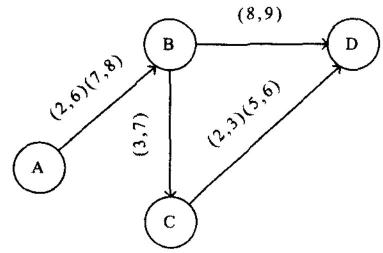

图 1　时序图

定义 4(时序最短路径)　时序最短距离是指信息从一个节点传到另一个节点所需的最短时间。用 $dist(s,u)$表示 s 与 u 之间的时序距离，若规定源点为 s，$dist[u]$表示 s 与 u 之间的时序最短距离。则对 s 到 u 的第 i 条有效时序路径，s 到 u 的时序距离为 $dist_i(s,u)=e_{in}.t_f-T_S$，其中 $e_{in}.t_f$ 表示 $event_i$ 从 s 到 u 结束的时间，T_S 为观察窗口起始时间。那么，s 到 u 的时序最短距离 $dist[u]=\mathrm{Min}(dist_i(s,u))$，$s\neq u$。具有最短时序距离的有效时序路径即为最短时序路径。

3 时序最短路径算法

该算法是在 Dijkstra 算法思想基础上，考虑路径的时序性，用 T-Dijkstra(即 Temporal Dijkstra)表示该算法。

3.1 算法思想

把节点集 V 分成两组：

(1)S：已求出最短路径的顶点的集合

(2)$V-S=U$：尚未确定最短路径的顶点集合

符号 $dist(s,u)$是源节点 s 到 u 的时序距离，$dist[u]$是指源节点 s 到 u 节点的最短时序距离。

求一个节点 V_0 到其他节点的时序路径，算法描述如下：

1. 初始时令 $S=\{V_0\}$，$U=V-S$

若存在关联 V_0，V_i 的弧$\langle V_0,V_i\rangle$，$i\neq 0$，记 $dist(V_0,V_i)$为$\langle V_0,V_i\rangle$弧上大于 T_S(观察窗口起始时间)的事件发生时间；

若不存在关联 V_0，V_i 的弧$\langle V_0,V_i\rangle$或者弧$\langle V_0,V_i\rangle$上事件发生时间皆小于 T_S，则 $dist(V_0,V_i)$为∞。

2. 从 U 中选取一个时序距离值为最小且不在 S 中的顶点 W，加入 S，则这个最小时序距离值就是 $dist[W]$。

3. 加入 W 后，若 S 中节点 V_S 与 U 中节点 V_U 存在新的有效时序边 $V_S\rightarrow V_U$(即弧$\langle V_S,V_U\rangle$上有事件，且$\langle V_S,V_U\rangle$弧上存在大于 $dist[V_S]$的事件起始时间，其中 $V_S\in S$，$V_U\in U$)，将$\langle V_S,V_U\rangle$弧上值大于 $dist[V_S]$的事件结束时间与之前保存的 $dist(V_0,V_U)$进行比较，选择较小的那个值替换掉原先的 $dist(V_0,V_U)$。

重复上述步骤 2、步骤 3，直到 S 中包含所有顶点即 $S=V$ 为止。

3.2 算法证明

已知：V 是所有顶点集合，S 是当前已找到最短路径的目标顶点集合。每次从 S 中的顶点出发扩展时序路径可达且最短的目标点。将源点 s 与其他顶点发生联系的最小开始时间定义为观察窗口时间。节点间链路的传输时间为 0。

证明正确性：

①首先考虑 S 为空的时候，此时还没有 s 到其他顶点的最短的时序路径。假设此时找到了从 s 一步可达且时序距离最短的目标点 s'，时序距离记为 $dist[s']$。那么从 s 到 s' 不存在其他时序距离小于 $dist[s']$ 的路径。这是因为，经过其他顶点到达 s' 的路径长度 $length$＝一步可达的某顶点 k 的时序距离＋k 到 s' 的时序距离。因为 k 到 s' 的时序距离非负，所以 $length \geqslant dist[s']$，因此不存在小于 $dist[s']$ 的其他路径。

②S 非空的情况。不失一般性，当 $|S|=K(K>0)$ 时，算法找到了下一个扩展点 u，$u \in V-S$，当前算法找到从源点 s 到 u 的最短时序距离为 $dist[u]$。我们假设从 s 到 u 还存在一条比 $dist[u]$ 距离更短的路径，记它的时序距离为 $dist(s,u)$。

如果在 $dist(s,u)$ 的路径上只有 u 不属于 S，那么根据我们算法的步骤，当前从源点 s 经过 S 到达 u 的时序路径都是已经比较过的，因此不存在从 s 到 u 比最短时序距离 $dist[u]$ 更短的路径。如果 $dist(s,u)$ 路径上除了 u 以外还存在其他不属于 S 的顶点，那么假设这条更短的路径初次走出 S 之外到达的顶点为 $x(x \in V-S)$，然后从 x 节点到达 u。如图 2 所示。

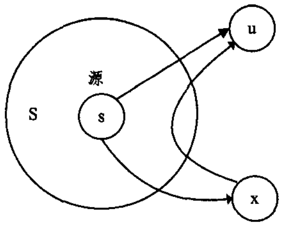

图 2 节点选取示意图

在这条路径上，分别记 $dist(s,x)$、$dist(x,u)$ 为源点 s 到顶点 x、顶点 x 到顶点 u 建立联系所花费的时间（时序距离）。那么有：

若 x 到 u 存在多个联系事件，第 i 个事件 $event_i$ 的发生时间是 $start(event_i)$。那么 $dist(x,u)$ 存在的充要条件为：存在 $start(event_i) \geqslant dist(s,x)$。对 x 到 u 上所有满足条件 $start(event_i) \geqslant dist(s,x)$ 的事件有：$dist(s,x) \leqslant dist(s,u)$。

在我们的假设中，x 是这条更短的路径初次走出 S 之外到达的顶点，也就是说 x 与 S 中的某个顶点存在满足时序要求的连边。根据当前步的选法有，在所有的从 s 经过 S 到达未扩展的顶点 $x(x \in V-S)$ 时序路径（x 是该路径中唯一不属于 S 的顶点）中，$dist[u]$ 是最短的时序距离，即 $dist[u] \leqslant dist(s,x)$。再由证明步骤①可得 $dist[u] \leqslant dist(s,u)$，与假设 $dist(s,x) < dist[u]$ 矛盾，因此 $dist[u]$ 是最短的时序路径。

3.3 算法复杂度分析

本算法是在经典 Dijkstra 算法基础上提出的，因此时间复杂度和空间复杂度都和 Dijkstra 算法大致相同。要计算网络中一个节点到其他节点的时序最短路径，时间复杂度为 $O(n^2)$，但是由于在选取当前最小时序距离的节点之后，并没有修改其他节点到源点的距离，因此与经典 Dijkstra 算法相比，本算法时间复杂度较低。若用 Fibonacci 堆来优化算法，可使复杂度降低到 $O(m+n\log n)$，其中 n 为网络中节点数，m 为边数。空间复杂度取决于所用的存储方式，若用邻接矩阵，空间复杂度为 $O(n^2)$，计算网络中所有节点对之间的最短时序距离时间复杂度为 $O(n^3)$。而文献[13]中，计算所有节点对之间的平均时序距离时间复杂度为 $\mathrm{Max}(O(k\varepsilon), O(n^2))$，其中 ε 为网络中所有事件次数，k 为网络中节点的最大事件发生次数，n 为节点数目。

4 时序最短路径实证分析

寻找图 3 网络中 a 节点到其他节点的最短时序路径。其中节点对之间每个 (x,y) 序对代表此节点对存在 x 时刻发生、y 时刻结束的事件。箭头方向为信息流动方向。假设观察窗口从 0 时刻开始。

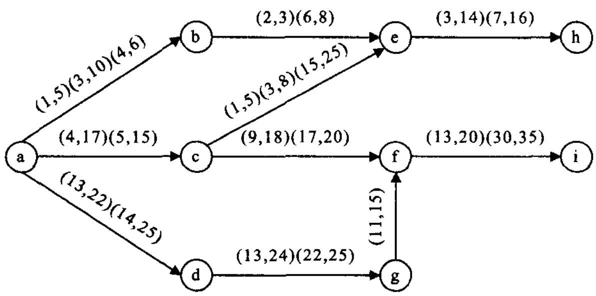

图 3 时序网络图

4.1 实证分析过程

把 V 分成两组：

(1)S：已求出最短路径的顶点的集合

(2)$V-S=U$：尚未确定最短路径的顶点集合

1. 将 a 放入集合 S，且 $dist[a]=0$，其余节点放入 U 集合内。

2. a)首先找与 a 有事件发生且事件发生时间大于观察窗口起始时间 0 的节点 b、c 和 d，选取满足条件的各链接上最小结束时间，则其为 a 到各节点的时序距离，则 $dist(a,b)=5$、$dist(a,c)=15$、$dist(a,d)=24$；

b)选取时序距离最小的那个节点，$dist[b]=\mathrm{Min}(dist(a,b), dist(a,c), dist(a,d))$，即找到 b，将 b 放入 S。

3. a)然后找与集合 S 中新增节点 b 有事件发生且满足 $start(event_i) \geqslant dist[b]$ 的节点，此处满足条件的点有 e，取满足条件的各链接上的最小结束时间，则其为 a 和 b 到各节点的时序距离，则 $dist(a,e)=8$，$dist(a,c)$、$dist(a,d)$ 不变；

b)选取时序距离最小的那个节点，$dist[e]=\mathrm{Min}(dist(a,e), dist(a,c), dist(a,d))$，即找到 e。

4. 以此类推找到 c，$dist[c]=15$，然后是 f，$dist[f]=20$，接着是 d，$dist[d]=22$，再找到 g，$dist[g]=25$，接下来是 i，$dist[i]=35$，最后是与 a 无有效时序路径的点 h，$dist[h]=\infty$。此时 $S=V$，这样就找完了 a 到其他点的最短时序路径值。

计算图 3 中节点 a 到其他各节点的时序最短距离的整个

（下转第 230 页）

International Conference on Services Computing (SCC 2011). Washington, DC, USA, 2011: 160-167

[15] Han X, Shi Z, Niu W, et al. Similarity-based Bayesian learning from semi-structured log files for fault diagnosis of web services [C]// 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). Toronto, Canada, 2010: 589-596

[16] Reiter R. A theory of diagnosis from first principles [J]. Artificial Intelligence, 1987, 32(1): 57-95

[17] Abreu R, Gemund A J C V. A low-cost approximate minimal hitting set algorithm and its application to model-based diagnosis [C]// Symposium on Abstraction, Reformulation and Approximation. Lake Arrowhead, CA, USA, 2009: 2-8

[18] Al-Masri E, Mahmoud Q H. Discovering the best web service [C] // 16th International Conference on World Wide Web (WWW). Banff, Alberta, Canada, 2007: 1257-1258

[19] Al-Masri E, Mahmoud Q H. QoS-based discovery and ranking of web services [C]// IEEE 16th International Conference on Computer Communications and Networks (ICCCN). Maui, Hawaii, 2007: 529-534

(上接第 187 页)

迭代过程如表 1 所列,初始时 S 中只有 a。

表 1 时序最短距离迭代过程

迭代	S	u	dist(a,b)	dist(a,c)	dist(a,d)	dist(a,e)	dist(a,f)	dist(a,g)	dist(a,h)	dist(a,i)
初始	{a}	—	5	15	22	∞	∞	∞	∞	∞
1	{a,b}	b	5	15	22	8	∞	∞	∞	∞
2	{a,b,e}	e	5	15	22	8	∞	∞	∞	∞
3	{a,b,e,c}	c	5	15	22	8	20	∞	∞	∞
4	{a,b,e,c,f}	f	5	15	22	8	20	∞	∞	35
5	{a,b,e,c,f,d}	d	5	15	22	8	20	25	∞	35
6	{a,b,e,c,f,d,g}	g	5	15	22	8	20	25	∞	35
7	{a,b,e,c,f,d,g,i}	i	5	15	22	8	20	25	∞	35

4.2 算法结果对比

从表 1 得知,对给定的时序网络图 3,用本文提出的算法得到节点 a 到其他节点的最短时序距离分别为:$dist[b]=5$, $dist[c]=15$, $dist[d]=22$, $dist[e]=8$, $dist[f]=20$, $dist[g]=25$, $dist[h]=\infty$, $dist[i]=35$。文献[13]的算法(即式(1))在将时间窗口 T 取为 35 时,得到节点 a 到其他节点的平均时序距离分别为:$\tau_{ab}=14.66$, $\tau_{ac}=16.52$, $\tau_{ad}=16.52$, $\tau_{ae}=16.523$, $\tau_{af}=\infty$, $\tau_{ag}=\infty$, $\tau_{ah}=\infty$, $\tau_{ai}=\infty$。

通过对比可知,利用式(1)所得的节点间平均时序距离和本文算法得到的最短时序距离是不同的,式(1)得到的是整个时间窗口内的平均值,并且可能会出现本来可达的节点之间平均时序距离为无穷大的情况,比如节点 a 到 f,而本文中提出的算法得到的是精确的结果。

结束语 本文提出的算法是针对更接近现实生活的时序网络的,可用于解决信息从源到目的地所需要的最短时间、疾病从起源地到其他地方需要的最短时间、挑选最快到达一个地点的航班等问题。文章在第 3 节提出最短时序路径算法,第 4 节进行严格证明,并在第 5 节进行实证分析。由此表明,本文提出的算法具有可行性。但是本算法是一种精确算法,计算复杂度和 Dijkstra 算法是一样的,在大规模复杂网络中,寻找节点间的时序最短路径可能会造成时间过长的问题。

参 考 文 献

[1] 荣玮. 基于道路网的最短路径算法与实现[D]. 武汉:武汉理工大学,2005

[2] 涂海丽. 最短路径算法及其应用探讨[J]. 科技广场,2011(9)

[3] 吴莲. 基于城市道路网的遗传最短路径算法研究[D]. 南京:南京理工大学,2010

[4] 于东凯,刘玉树. 基于平面图的最短路径算法的研究[J]. 北京理工大学学报,2003,21(1)

[5] 董俊,黄传河. 改进 Dijkstra 算法在 GIS 导航应用中最短路径搜索研究[J]. 计算机科学,2012,39(10)

[6] 艾菊梅,周书民,陆玲. 基于 WebGIS 公交查询平台与移动增值服务[J]. 微计算机信息,2007(10)

[7] SCHULTESD. Route planning in road networks[D]. Karlsruhe: UniversitätFridericiana, 2008

[8] Fu L. Real-time vehicle routing and scheduling in dynamic and stochastic traffic networks[D]. Edmonton, Alberta: University of Alberta, 1996

[9] Nicosia G, Oriolo G. An approximate A* algorithm and its application to the SCS problem[J]. Theoretical Computer Science, 2003, 290(3): 2021-2029

[10] Bander J L, White C C. A heuristic search algorithm for path determination with learning[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 1998, 28(1): 131-134

[11] 宋青,汪小帆. 最短路径算法加速技术研究综述[J]. 电子科技大学学报,2012(2)

[12] 唐晋韬,王挺,王戟. 适合复杂网络分析的最短路径近似算法[J]. 中国:软件学报,2011,22(10)

[13] Rajk P, Jari S. Path lengths, correlations, and centrality in temporal networks[J]. Finland: Phys. Rev. E 84, 2011: 016105

[14] Dijkstra E W. A note on two problems in connexion with graphs [J]. Numerical Mathematics, 1959, 1(1): 269-271

[15] 严蔚敏,吴伟民. 数据结构[M]. 北京:清华大学出版社,2001, 186-190

[16] 张先迪,李正良. 图论及其应用[M]. 北京:高等教育出版社, 2005

[17] Petter H, Jari S. Temporal networks[J]. Phys. Rep, 2012, 519: 97-125