# 时态图中的可达性和基于时间的路径查询

**摘要**

时间图是一个图，其中顶点在特定时间相互通信，例如，𝐴在上午11点调用and并进行7分钟的谈话，这是通过从𝐴到edge的边缘建模的，开始时间为“上午11点” 和持续时间“7分钟”。 时间图可以用于模拟具有时间相关活动的许多网络，但是用于分析时间图的有效算法严重不足。 我们研究基本问题，例如在时间图中回答可达性和基于时间的路径查询，并提出专门设计用于在时间图中处理这些查询的有效索引技术。 我们的结果表明我们的方法在索引构造和查询处理中都是高效和可扩展的。

## 导言

图已被广泛用于模拟和研究各种在线社交网络，移动通信网络，电子商务网络，电子邮件网络等。在这些图的结构，顶点是用户或公司，而边缘模型它们之间的关系。 然而，为了简化分析，这些图中经常缺少一种类型的信息：实际上，关系发生在特定时间并且持续一段时间。 形式上，这可以被建模为一个时间曲线图，其中，每个边缘由（𝑢，𝑣，𝑡，λ），这表明从𝑢到𝑣的关系开始于时刻𝑡并持续λ的持续时间。 可以存在𝑢和𝑣之间的多个边缘，指示在不同的时间段发生的他们的关系。

时间图可用于模拟和研究上述图中的许多与时间有关的活动。 例如，用户在在线社交网络中跟踪或标记不同时期的其他用户; 朋友们在手机网络的不同时段互相聊天; 人们在电子邮件网络或即时消息网络中的不同时间向对方发送消息; 客户在网上购物平台的不同时间从卖家购买产品，或在电子商务网络等不同时期的不同方之间发生不同类型的交易。

已经广泛研究了非时间图（即，没有时间信息的一般图）的研究。 然而，对于时间图，甚至一些基本问题还没有得到很好的研究（例如，图遍历，连通分量，可达性，“最短路径”等）。 在本文中，我们研究了计算时间图中从顶点到另一个顶点的可达性和“最短路径距离”的问题。

图形可达性和最短路径都有许多重要的应用。 然而，在时间图中，由于时间施加的顺序，问题变得更加复杂。 例如，考虑图1（a）中所示的玩具火车时刻表图表，其中每个边缘旁边的数字是火车离开的一天（例如，第1天，第2天等），并假设持续时间 每列火车需要2天。 假设现在想要从a到d旅行。 如果他选择通过b去d，那么他可以在第1天或第2天离开，并且他将在第6天到达d。现在假设他想要在第4天稍后离开，那么他无法到达d因为 火车在第6天从第6天的c到达，但火车在第5天从c离开到d。 然而，如果我们不考虑时间信息，那么旅行者仍然可以在第4天从a到c乘坐火车，在这种情况下没有意识到他不会从c到d赶火车。

给定两个顶点，𝑢和𝑣，在时间图和时间间隔𝑇，我们研究如何计算（1）𝑢是否可以达到𝑣内，（2）最早的时间𝑢可以达到𝑣内的𝑣，和（ 3）从𝑢到𝑣的最快路径的持续时间。 从时间点网络连通[1]，时间中间性和接近性[2]，[3]，时间连通等时间网络的研究中发现了从源顶点到目标顶点的可达性，最早到达时间和最小持续时间。 [4]中的组成部分，信息传播[5]，信息潜伏期[6]，[7]，时间效率和聚类系数[4]，时间小世界行为研究[8]等。

然而，上述引用的工作并未集中于设计用于计算可达性，最早到达时间和最小持续时间的有效算法，并且它们的结果主要来自小时间图。吴等人。 [9]对现有算法进行了重大改进[10]，并且他们的算法可以处理比现有算法更大的时间图。然而，他们的算法并非设计用于在线查询，而在许多应用中，要求实时地从源顶点到目标顶点找到可达性，最早到达时间或最小持续时间。王等人。 [11]提出了一种索引方法，用于回答从源顶点到目标顶点的最早到达时间或最小持续时间的在线查询。但是，它们的索引方法无法扩展到大的时间图。此外，它们的索引方法不支持动态更新，这对于时间图是实际上重要的，因为在大多数真实世界的时间图中更新是频繁的。有鉴于此，我们提出了一个索引来支持大型时态图的高效在线查询，这也支持高效的动态更新。

我们的方法首先将时间图转换为新图，这是一个有向无环图（DAG），现有的可达性查询索引方法[12]，[13]，[14]，[15]，[16]，[ 也可以应用17]，[18]，[19]，[20]。 但是，此DAG通常比现有方法处理的DAG大得多。 它还具有时间图的独特属性，而所有现有方法都是为处理非时间图而设计的。 因此，需要设计更加可扩展的方法，这些方法也考虑时间图的属性。

我们提出TopChain，这是一种用于回答可达性查询的标签方案。 标记方案，例如2跳标签[21]，为每个顶点构造两个标签𝑣，𝐿𝐿（𝑣）和𝐿𝐿（𝑣），其中𝐿𝐿（𝑣）和𝐿𝐿（𝑣）是一组 可以达到𝑣并且可以从𝑣到达的顶点。 通过交叉𝐿𝑜𝑢𝑡（𝑢）和𝐿𝐿（𝑣）来回答是否能达到query的问题，因为如果𝑢能够达到𝑣，则在𝐿𝐿（𝑢）和𝐿𝐿（𝑣）中存在共同的顶点。 然而，𝐿𝐿（𝑣）和𝐿𝐿（𝑣）往往太大，并且已经提出了各种方法来减小它们的尺寸[12]，[13]，[11]，[18]，[20]。

TopChain将输入DAG分解为一组链[22]，[23]，其中链是有序的顶点序列，使得每个顶点可以到达链中的下一个顶点。 因此，𝐿𝐿（𝑣）和𝐿𝐿（𝑣）只需要保持链中的最后一个和第一个顶点可以达到𝑣，并且可以从𝑣到达。 但是，对于大图，链的数量仍然可能太大，作为解决方案，TopChain对链进行排名，并且仅对每个顶点使用顶部的𝑘链。 这样，每个顶点的标签大小最多保持在2𝑘，而索引构造只需要线性时间，因为𝑘是一个小常数。 𝑘标签可能无法回答每个查询，因此仍可能需要在线搜索。 但是，标签可用于进行有效的修剪，在线搜索可快速收敛。

我们的工作贡献总结如下：

∙ 我们提出了一种有效的索引方法TopChain，用于在时间图中回答可达性和基于时间的路径查询，这对于分析具有基于时间的活动的真实世界网络非常有用。

∙ TopChain具有线性索引构造时间和线性索引大小。 尽管现有方法可以应用于我们的变换图以回答可达性查询，但我们的方法是唯一利用时态图的属性来设计索引方案的方法。 TopChain还应用时态图的属性来设计用于动态更新索引的有效算法。

∙ 我们在一组15个真实时间图上评估了TopChain的性能。 与最先进的可达性指标[14]，[16]，[17]，[19]，[20]相比，TopChain在查询处理中的速度提高了几倍到几个数量级， 较小或可比较的指数规模和指数建设成本。

**论文大纲**

第二节定义了问题。 第III节描述了图形转换。 第IV和V节介绍了索引和查询处理的细节。 第六节介绍了标签的一些改进。 第VII节报告实验结果。 第八节讨论相关工作，第九节给出结论性意见。

1. **问题定义**

设𝒢=（𝒱，ℰ）为时间图，其中𝒱是𝒢中的顶点集，ℰ是𝒢中的边集。 边𝑒∈ℰ是四（𝑢，𝑣，𝑡，λ），其中𝑢，𝑣∈𝒱，𝑡是起始时间，λ是从时间𝑡开始从𝑢到𝑣的遍历时间。 我们用𝑡（𝑒）表示𝑒的起始时间，用λ（𝑒）表示e 的遍历时间。 或者，我们可以认为𝑒在[𝑡，𝑡+λ]期间是活跃的。

如果边是无向的，那么边的起始时间和遍历时间从𝑢到𝑣是相同的，从𝑣到𝑢。 我们关注本文中的有向时间图，因为无向边可以通过两个双向边来建模。

我们用Π（𝑢，𝑣）表示从𝑢到𝑣的时间边缘集，以及𝒢到𝑢（𝑢，𝑣）的𝑢到𝑣的时间边数，即π（𝑢，𝑣）=|Π（𝑢，𝑣）|。 我们还定义了从𝑢到𝑣的最大时间边缘数，对于𝒢中的任何𝑢和𝑣，乘以π= max {π（𝑢，𝑣）:(𝑢，𝑣）∈（𝒱×𝒱）}。

我们将𝒢中的顶点out的邻域集定义为Γ𝑜𝑢𝑡（𝑢，𝒢）= {𝑣：（𝑢，𝑣，𝑡，λ）∈ℰ}，并将out的out出度定义为𝑑 𝑜𝑢𝑡（𝑢，𝒢）=Σ𝑣∈Γ𝑜𝑢𝑡（𝑢，𝒢）π（𝑢，𝑣）。 类似地，我们将𝑢的in-neighbors和in-degree定义为Γ𝑖𝑛（𝑢，𝒢）= {𝑣：（𝑣，𝑢，𝑡，λ）∈ℰ}和𝑑𝑑（𝑢，𝒢）=Σ𝑣∈ Γ𝑖𝑛（𝑢，𝒢）π（𝑣，𝑢）。

在时间图表𝒢甲颞路径𝑃是边缘𝑃=⟨𝑒1，𝑒2，...，𝑒𝑝⟩的序列，使得𝑒𝑖=（𝑣𝑖，𝑣𝑖+ 1，𝑡𝑖，λ𝑖 ）∈ℰ是𝑖个时间上𝑃1≤𝑖≤𝑝，和（𝑡𝑖+λ𝑖）边缘≤𝑡𝑖+ 1 1≤𝑖<𝑝。 请注意，在𝑃最后边缘（𝑣𝑝，𝑣𝑝+ 1，𝑡𝑝，λ𝑝），我们不把（𝑡𝑝+λ𝑝）的约束，因为𝑡𝑝+ 1没有为路径𝑃定义。 事实上，（𝑡𝑝+λ𝑝）是𝑃的结束时间，通过𝑒𝑛𝑑（𝑃）表示。 我们还将𝑃的起始时间定义为𝑠𝑡𝑎𝑟𝑡（𝑃）=𝑡1。 我们将𝑃的持续时间定义为𝑑𝑢𝑟𝑎（𝑃）=𝑒𝑛𝑑（𝑃） - 𝑠𝑡𝑎𝑟𝑡（𝑃）。

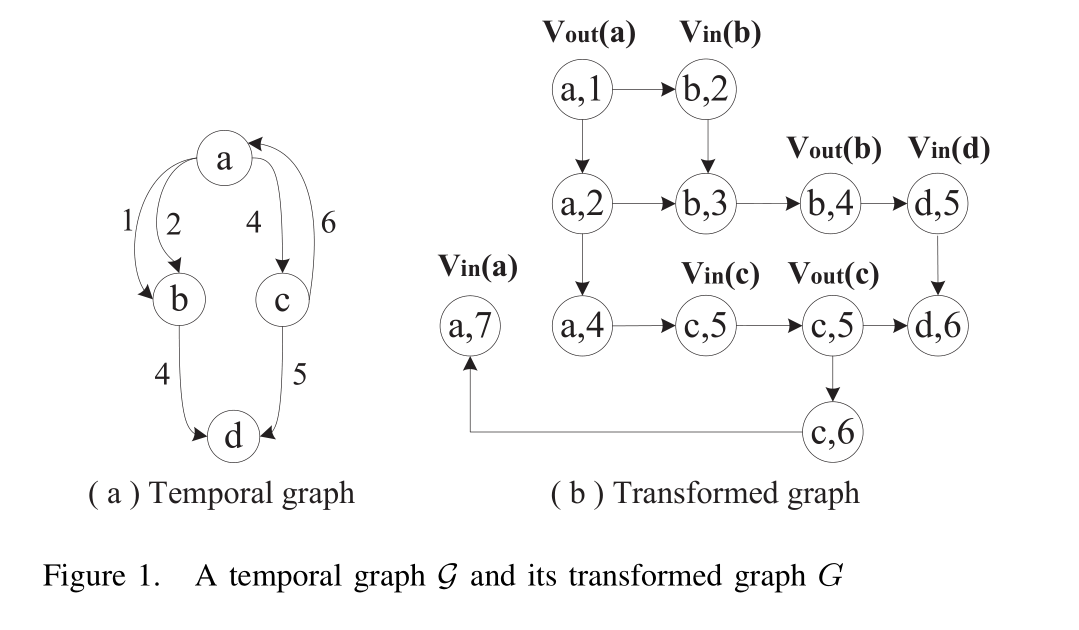
基于时间路径，我们给出了最小时间路径[9]和时间可达性的定义如下。

定义1（最小时间路径[9]）：设P（𝑢，𝑣，[𝑡α，𝑡ω]）= {𝑃：𝑃是从𝑢到temporal的时间路径，使得𝑠𝑡𝑎𝑟𝑡（𝑃）≥𝑡α，𝑒𝑛𝑑 （𝑃）≤𝑡ω}。

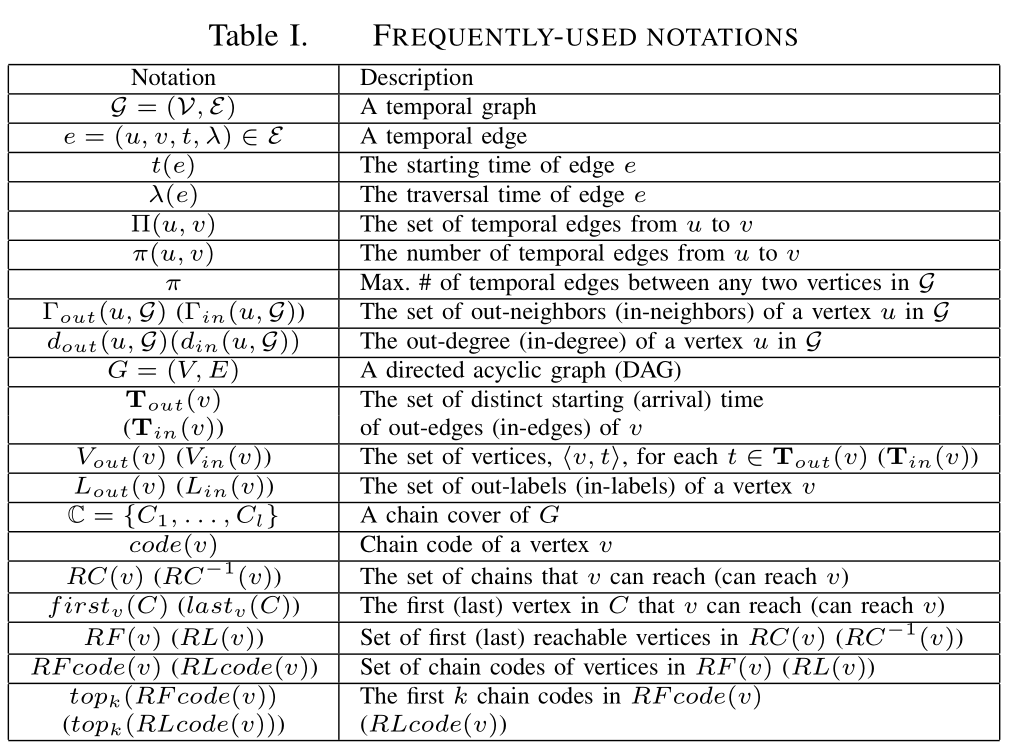
时间路径𝑃∈P（𝑢，𝑣，[𝑡α，𝑡ω]）是最早到达路径，如果𝑒𝑛𝑑（𝑃）= min {𝑒𝑛𝑑（𝑃'）：𝑃'∈P（𝑢，𝑣，[𝑡 α，𝑡ω]）}。 在[𝑡α，𝑡ω]内从𝑢到达𝑣的最早到达时间由𝑒𝑛𝑑（𝑃）给出。

时间路径𝑃∈P（𝑢，𝑣，[𝑡α，𝑡ω]）是最快的路径，如果𝑑𝑢𝑟𝑎（𝑃）= min {𝑑𝑢𝑟𝑎（𝑃'）：𝑃'∈P（𝑢，𝑣，[𝑡α， 𝑡ω]）}。 在[𝑡α，𝑡ω]内从𝑢到𝑣的最小持续时间由𝑑𝑢𝑟𝑎（𝑃）给出。

定义2（时间可达性）：给定两个顶点𝑢和𝑣，以及时间间隔[𝑡α，𝑡ω]，如果P（𝑢），𝑢可以在[𝑡α，𝑡ω]内达到𝑣（或𝑣可以从𝑢到达） ，𝑣，[𝑡α，𝑡ω]）/ =∅，即从𝑢到exists存在时间路径𝑣，使得𝑠𝑡𝑎𝑟𝑡（𝑃）≥𝑡α和𝑒𝑛𝑑（𝑃）≤𝑡ω



示例1：图1（a）示出了时间图𝒢。 为简单起见，我们假设每个边的遍历时间是1个时间单位。 在𝒢，since可以在时间间隔[2,5]内达到since，因为有一个时间路径𝑃=⟨（𝑎，𝑏，2,1），（𝑏，𝑑，4,1）⟩，而𝑎不能达到𝑑 在[1,3]内，因为在[1,3]内没有从𝑎到𝑑的时间路径。 给定源顶点𝑎，目标顶点𝑑和时间间隔[1,10]，𝑃1=⟨（𝑎，𝑏，2,1），（𝑏，𝑑，4,1）⟩是最早到达路径 到达时间4 + 1 = 5，𝑃2=⟨（𝑎，𝑐，4,1），（𝑐，𝑑，5,1）⟩是最快的路径，持续时间（5 + 1） - 4 = 2。注意 𝑃1不是最快的路径，𝑃2不是[1,10]中最早到达的路径。、



问题定义：给定时间图𝒢，我们建议构造一个索引，使得给定源顶点𝑢和目标顶点𝑣，以及时间间隔[𝑡α，𝑡ω]，我们可以有效地回答以下查询:( 1）𝑢是否能够在[𝑡α，𝑡ω]内达到𝑣，（2）在[𝑡α，𝑡ω]内从𝑢到ear的最早到达时间，以及（3）从𝑢到minimum的最短持续时间 到[𝑡α，𝑡ω]内的𝑣。

我们的方法也可以应用于计算另一种称为最近离开路径的最小时间路径[9]。 然而，最新出发路径的概念与最早到达路径的概念对称。 因此我们省略了细节。

1. **图形转换**

除了索引和查询效率之外，设计用于查询时间图的索引方法的另一个关键考虑因素是索引必须支持有效的动态更新，因为在大多数实际应用中随时间创建和频繁地添加时间边缘。 我们发现[9]中的图变换方法可以应用于设计一个有效的索引，该索引也允许有效的动态更新。

我们首先介绍如何将时间图𝒢=（𝒱，ℰ）转换为新图，𝐺的构建包括两个阶段：

1）构造顶点集：对于每个顶点𝑣∈𝒱，在𝑉中创建顶点如下。

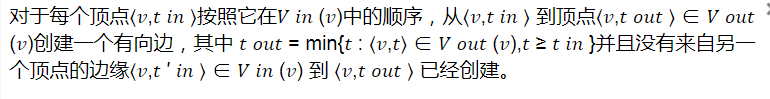
a）设𝑇𝑖𝑛（𝑢，𝑣）= {𝑡+λ：（𝑢，𝑣，𝑡，λ）∈Π（𝑢，𝑣）}对于每个𝑢∈Γ𝑖𝑛（𝑣，𝒢），并且 T 𝑖𝑛 (𝑣)= ∪ 𝑢∈Γ 𝑖𝑛 (𝑣,𝒢) 𝑇 𝑖𝑛 (𝑢,𝑣)，即T 𝑖𝑛（𝑣）是不同时间实例的集合，其中来自in的邻居的边到达𝑣。 创建| T𝑖𝑛（𝑣）| 复制的副本，每个都标有⟨𝑣，𝑡⟩，其中𝑡是T𝑖𝑛（𝑣）中的明显到达时间。 将这组顶点表示为 𝑉 𝑖𝑛 (𝑣)，即𝑉 𝑖𝑛 (𝑣) = {⟨𝑣,𝑡⟩ : 𝑡 ∈ T 𝑖𝑛 (𝑣)}.。 按时间降序排列𝑉 𝑖𝑛 (𝑣) 中的顶点，即任何⟨𝑣,𝑡 1 ⟩,⟨𝑣,𝑡 2 ⟩ ∈ 𝑉 𝑖𝑛 (𝑣), ⟨𝑣,𝑡 1 ⟩⟨𝑣,𝑡 2 ⟩ in 𝑉 𝑖𝑛 (𝑣) iff 𝑡 1 > 𝑡 2

b）设𝑇 𝑜𝑢𝑡 (𝑣,𝑢) = {𝑡 : (𝑣,𝑢,𝑡,𝜆) ∈ Π(𝑣,𝑢)} 对于每个𝑢 ∈ Γ 𝑜𝑢𝑡 (𝑣,𝒢), 和 T 𝑜𝑢𝑡 (𝑣)= ∪ 𝑢∈Γ 𝑜𝑢𝑡 (𝑣,𝒢) 𝑇 𝑜𝑢𝑡 (𝑣,𝑢)。 创建|𝑣 的副本∣T 𝑜𝑢𝑡 (𝑣)∣，每个都标有⟨𝑣，𝑡⟩，其中𝑡是T𝑜𝑢𝑡（𝑣）中不同的起始时间。 将这组顶点表示为 𝑉 𝑜𝑢𝑡 (𝑣)，即𝑉 𝑜𝑢𝑡 (𝑣) = {⟨𝑣,𝑡⟩ : 𝑡 ∈ T 𝑜𝑢𝑡 (𝑣)}。 按照时间的降序对𝑉𝑜𝑢𝑡（𝑣）中的顶点进行排序。

2）边集的构造：在𝐸中创建边，如下所示。

a）设𝑉𝑖𝑛（𝑣）= {⟨𝑣，𝑡1⟩，⟨𝑣，𝑡2⟩，...，⟨𝑣，𝑡ℎ⟩}，其中𝑡𝑖>𝑡𝑖+ 1为1≤𝑖<ℎ 。从每个⟨𝑣，𝑡 𝑖+1⟩到⟨𝑣，𝑡 𝑖⟩创建一个有向边，1≤𝑖<ℎ。如果ℎ≤1，则不会创建边缘。以相同的方式为𝑉 𝑜𝑢𝑡（𝑣）创建边。

b）



C）



以下示例说明了图形转换。

实施例2：图1（b）示出了曲线图转化𝐺的时间图表𝒢在图1（a），其中，λ= 1对于所有的边缘。 虽然在𝒢中有一个循环⟨（𝑎，𝑐，4,1），（𝑐，𝑎，6,1）𝒢，但𝐺是DAG。

1. **Top-K链标签**

在本节中，我们将介绍top- 我们专注于讨论可达性查询，并讨论如何应用TopChain来回答第V-B节中基于时间的查询。

大多数标记方案首先通过将每个强连通分量（SCC）折叠成超顶点，将输入有向图转换为有向无环图（DAG），然后在更小的DAG上构建标记。 在这里，我们证明转换图是DAG。

给定一个变换图 为简单起见，我们经常在讨论中使用 只有在需要参考time的时间戳。

由于空间限制，所有的引理和定理的证明都在本文的完整版本中给出[24]。