A Lagrangian Approach to Storage and Access Methods for Spatio-Temporal Network Datasets: A Summary of Results

Draft - April 29, 2010

Michael R. Evans
University of Minnesota
4-192 EE/CS Building
200 Union Street SE
Minneapolis, MN 55455
mevans@cs.umn.edu

KwangSoo Yang
University of Minnesota
4-192 EE/CS Building
200 Union Street SE
Minneapolis, MN 55455
ksyang@cs.umn.edu

Shashi Shekhar
University of Minnesota
4-192 EE/CS Building
200 Union Street SE
Minneapolis, MN 55455
shekhar@cs.umn.edu

ABSTRACT
Spatio-temporal networks are becoming popular with the rise of traffic information being incorporated into street mapping utilities such as Google Maps. This additional information adds potentially large time series of traffic information to each road segment. Naive implementations of secondary storage of spatio-temporal networks may not be sufficient, as current techniques commonly store entire time series with an edge, or store networks a snapshot at a time. Queries such as route evaluation will require a different node and different time-step for each segment of a route, resulting in excessive disk I/O. We introduce a Lagrangian-Connectivity Framework for storing spatio-temporal networks based on each node’s spatio-temporal connectivity.

Categories and Subject Descriptors
?

Keywords
Spatio-Temporal Networks, Graph Storage, Databases

1. INTRODUCTION
Spatio-temporal network databases are gaining in popularity owing to various applications such as route planning, public transportation, evacuation planning, and trajectory pattern mining. Spatio-temporal network databases deal with a time-dependent network model, where topological connectivity is characterized by both spatial and temporal relationships. The requirements of a spatio-temporal network differs dramatically from those of a traditional spatial network. One of the most appealing properties of spatio-temporal networks is that the topology is changing over time; it represents the dynamic environment of the real road network, e.g., changing traffic levels. This property is also the biggest challenge of using spatio-temporal networks. Each edge has a time series associated with it, potentially of great size when using data from sources such as NAVTEQ [1], who has traffic data in 5 min increments (see Figure 1) for all major highways in the USA. UPS began evaluating new concepts of shortest path, such as shortest path based on gas consumption, which aims to minimize left-hand turns due to idling at stop lights.

U.P.S. Embraces High-Tech Delivery Methods
By CLAUDIA H. DEUTSCH
The United Parcel Service has embraced the latest methods to improve deliveries and air safety.
July 12, 2007 | BUSINESS | NEWS

(a) UPS starts using smarter routing techniques
(b) Local Newspaper’s Traffic Report

Figure 1: Road Networks now contain traffic level data over time.

1.1 Motivation
For years, people have been using routing applications such as driving directions in their everyday lives. More recently, technologies for tracking traffic levels on road networks has allowed companies to compile and sell this information, turning our standard notion of a static road network into a spatio-temporal road network, varying in travel time as the day progresses, constantly changing the estimated travel time for routing queries. However, spatio-temporal networks have proven more difficult to deal with computationally than
their static brethren. The sheer length of temporal information in a time series for each road (edge) makes main memory storage impractical, therefore calling for new techniques for secondary storage that capitalizes on the uniqueness of these datasets to maintain usability.

This paper introduces the idea of modeling and storing these spatio-temporal networks with a Lagrangian approach, that is, following a path through time from one spatio-temporal node to another, see Figure 2. As you are driving to some destination, the traffic levels ahead of you are continuously changing. The longer you spend on your current road segment, the more different the next road's traffic levels could be as compared to when you started driving. It is important to query traffic level (edge weights) just-in-time, as you are set to enter that segment.

Current travel time estimates from services such as Google Maps or Bing Maps are derived from a patchwork of aggregations. While many entities use traffic data to estimate road travel time, they do so in a snapshot format, as in, the travel time for each edge is based on the conditions at the time the route is started, not as the user reaches each road. We highlight that changing traffic levels need to be queried at the time a car is expected to reach each segment, based on the car’s previous travels.

1.2 Related Work

In the literature [17, 18, 20], we have seen that the most popular and efficient method for minimizing I/O cost for spatial networks has been CCAM, which models access patterns of the network data and stores frequently accessed data into the same disk page. Although a promising start, they cannot be applied directly into spatio-temporal since it does not consider the time-dependent model. The Time Aggregated Graph (TAG) [11] has been proposed to model spatio-temporal graphs and also reduce storage requirements in main memory. Spatio-temporal road network data, however, is often too large to store in main memory and is necessarily stored in secondary storage.

Motivated by these disadvantages of state-of-the-art network models, we develop a time-dependent graph model to improve the disk I/O performance by means of a better exploitation of the graph partitioning. Balancing two conflicting requirements is the most challenging and allows us to model new storage structure to capture spatio-temporal aspects. The basic idea is to divide the data such that the resulting partitions are optimized for frequent access route patterns. However, this approach does not remedy the storage space problem. The next step is to remove the redundant data and reduce the size of time-expanded networks using a time-aggregated graph.

1.3 Contributions

In this paper, we study the problem of storing and accessing spatio-temporal networks in secondary memory. Real-world road network datasets are currently available, but it is currently unclear how to handle the large temporal information associated with each road segment. We introduce a Lagrangian-based Connectivity Framework utilizing the innate spatio-temporal connectivity of nodes. Using a Lagrangian model forces one to follow paths through space and time, an important factor when dealing with road networks with constantly changing traffic levels and travel times.

The Lagrangian concept is well-known in other fields such as physics [], however, we intend to use this idea in a new way to model and design efficient secondary storage techniques. Since we know how these spatio-temporal networks will and must be used, in this Lagrangian sense, we can partition our networks for disk page storage based on these connections. We introduce a Lagrangian-Connectivity Framework for describing these spatio-temporal paths, connections, cuts, etc, that are relevant to our disk storage problem. We also introduce two storage techniques for efficiently storing spatio-temporal networks. One, the Time-Aggregated Snapshot Model, is most efficient when the travel time is not varying significantly and another, the Lagrangian-Connectivity Model is most efficient when travel times fluctuate quickly, such as during rush-hour. We then investigate the performance compared to existing storage techniques and validate our model with single route evaluation queries using the Minneapolis road network.

The contributions of this paper are as follows:

- Introduce a Lagrangian-based framework to describe storing spatio-temporal networks.
- Formulate an physical model to store and access spatio-temporal networks.
- Empirically evaluating and analyzing our model on real road network data.

1.4 Outline

The paper is organized as follows. In Section 2, we briefly review related work that influenced our design. Section 3 presents a motivational example of single route evaluation queries on a spatio-temporal network. Section 4 introduces our Lagrangian-Connectivity Framework, followed by Section 5 discussing disk storage techniques based on this framework. In Section 6, we present our experimental validation, utilizing the Minneapolis, MN road network with simulated traffic levels. Lastly, Section 7, we conclude and present future work.

2. RELATED WORK

Over the last decade, considerable works on static and spatio-temporal networks focused either on efficient indexes or performance enhancing algorithms for time-dependent route planning in road networks. In particular, Samet et al. [16] proposed an efficient index structure of the spatial network,
which found k-nearest neighbors using a single route evaluation over a shortest path quad tree, while Daniel Delling et al. [6] developed a time-dependent shortest path algorithm, which pre-computed the shortest paths between all possible sources and destinations, and encoded them in a compact form to speed up the shortest path evaluation. However, none of these works fulfilled on the I/O efficient storage and access methods in spatio-temporal network database. Even though secondary memory structures were designed towards the processing of queries over a spatial network, there is no previous work on time-dependent networks.

In the discrete-time model, a well-known approach for dealing with time dependent network problems is to construct a time-expanded network which was first introduced by Ford and Fulkerson [9, 10]. The advantage of this approach is that it converts the time-varying optimization problem into a classical static network problem using the time-expanded network. Unfortunately, the major drawback of the time expanded graph is the inefficiency in terms of disk storage cost. Due to the time-expanded replica of the static model, it turns out that the input size in the time-expanded networks increases drastically according to a time horizontal set. In other words, the storage cost in the time-dependent model becomes higher than the cost of the time-independent model.

Topological ordering, traversal partitioning, and graph partitioning have been used to optimize access methods [3, 14, 7]. Topological ordering and traversal partitioning, however, need preprocessing to layout the graph and is slower than graph partitioning. In the literature [20, 18, 17, 19], possibly the most popular and efficient method in terms of I/O cost has been CCAM, which clusters a pair of connected nodes that are more likely to be accessed together. CCAM exploits a two-way partitioning algorithm [4] based on min-cut criteria and mainly focus on the static road network. Although KL, FM, and two-way approach [13, 8, 4] give good partitions, they are relatively slow and not scalable to large scale data. There has been considerable work in the graph partitioning literature [2]. The multi-level partitioning, which runs in linear time and produces high quality partitions, are introduced to remedy this problem. The main known fast algorithm is Metis [12], which minimizes the edge-cuts and also satisfies a balancing multi-constraints associated with each node and edge.

Ford and Fulkerson [9, 10] introduces the concept of time-expanded network model with the maximal dynamic network flow problem. A key property in time-expanded network is that it solves the time-dependent routing problem by means of a topological visiting of static acyclic expanded graph, which encapsulates the time-dependent attributes. The classic time-expanded model is straightforward to design and implement the time-dependent network and widely used to evaluate the routing with various time-dependent network problems. However, it needs an amount of node and edge expansion to model the time-dependent network. Orda and Rom [15] proposes the continuous time-dependent network model, which defines edge delay over time as piecewise linear functions. This approach assumes that delay functions are represented by piece-wise linear function and a significant amount of the edge weight on the network is static or does not change substantially over time [5]. The process of linear interpolation between points removes redundant attributes and stores the network data into the space-efficient form as piece-wise linear function. However, the number of interpolation points increased when the network is perturbed over time. Time Aggregated Graphs (TAG) [11] were proposed to address the storage cost problem in time-expanded network. This model aggregates time-dependent attributes instead of replicating the entire graph for time set and efficiently saves the memory usage. Since it aggregates edge weights over time set within the one pair of node, it could remove the duplicated attributes. However, it assumes that the whole graph is loaded into the memory and does not consider the disk storage model.

3. ROUTE EVALUATION EXAMPLE

In this section, we will step through an example of single route evaluation on a spatio-temporal network. Note how the network is continually changing even while a car may be traveling along an edge, something that must be accounted for when determining travel time.

Figure 3 shows six time snapshots of the same road network, 1 minute apart, visualizing two separate cars traveling from Home to Work. Note that while most of the edge values, representing travel time for that edge at that time instant, do not change, the edge DE is changing. This represents changing traffic levels on a highway. In each time-step, the cars advance closer to their destination. As Car 2 enters the road segment DE at 8:02 AM, the travel time of edge DE is 3 minutes.

Note how as each car reaches another node in the graph, we need to know the next edge’s travel time at that time instant. If Car 2 had entered road segment DE at 8:05 AM, it’s travel time would have only been 1 minute. Figure 5 shows a Time Aggregated Graph, visualizing the temporal changes to edge DE, along with the varying travel time each car would have taken on when it started its trip. When talking about actually storing this type of network on disk, a potentially more intuitive way to represent these spatio-temporal edges is using a Time Expanded Network.

Figure 4 shows these temporal edges connecting nodes dependent on their travel time. Note how the two cars are moving over time: during some time-steps they are on edges, traveling through time, and others they are on nodes, about to depart on their next edge.

Time Expanded Network: Let $G = (V, E)$ be a network (directed graph) that consist of a set of nodes $V$ and a set of edges $E$ with $n = |V|$ nodes and $m = |E|$ edges. In time independent network, on each edge $(v, w) \in E$, constant travel times $c$ are associated. For a given time horizon set $T$, time expanded network is defined as $G_T = (V_T, E_T)$ such that

$V_T = \{v(t) \mid v \in V, t = 1, \ldots, T\}$

$E_T = \{(v(t), w(t')) \mid (v, w) \in E, t' = t + s_{vw}(t) \leq T\}$

where the non-negative travel time $s_{vw}(t)$ is associated to traverse an edge $(v, w)$ if we leave the node $v$ at time $t$. 
Time-expanded networks can be viewed as a static graph by expanding the time space. Notice that the time expanded network has \((T + 1)\) copies of each node such that the size is linear in \(T\) and needs a large amount of storage as a worst case lower bound of \(\Omega(nT + mT)\). We call such a copy a time-stamping layer. A key property in the time-expanded network is that every edge connects a lower-layer node to a higher-layer node over time \(t\) and the network is a topological ordered in increasing temporal order. This observation shows that the network is acyclic, such that search algorithms only expand a search space to equal or higher order nodes.

Due to the properties of a Time Expanded Network, certain connectivity characteristics become apparent. In the next section, we will introduce a framework designed to capture these traits to improve efficiency and performance for secondary storage.

4. LAGRANGIAN-CONNECTIVITY FRAMEWORK

In this section, we compare more traditional views of spatio-temporal networks to a Lagrangian approach, emphasizing temporal as well as spatial connections between nodes and the continuing global network changes during edge traversal.

4.1 Basic Concepts

When examining the use-cases of spatio-temporal networks, such as query processing for road networks, some key concepts become more clear. Connections between nodes are represented in time, as in, traversing from one node to another necessarily moves through time. For example, the network not only changes as each node is reached, but also as an edge is traversed as travel times are associated with edges. This Lagrangian-based concept steers us to change the way we store and access information regarding these spatio-temporal networks. Below are some key concepts when viewing networks from a Lagrangian sense:

Lagrangian Connectivity: Traditional road network connectivity is based on spatial connections between nodes, possibly with an edge weight representing distance or travel time. The Lagrangian corollary describes measurements recorded while traveling with a network. That is, nodes cannot be connected without time, so any edge traveled is...
Lagrangian Path: An object moving through a spatio-temporal network moves through time by definition. Therefore, when moving from node to node, the edge being used for the traversal is existent at that moment only. As time progresses, this edge connecting these nodes will likely change in value. Figure 2 shows an object moving through a simple spatio-temporal network. Since this network has both spatial and temporal information, we need to access edges based on their spatio-temporal location. The path cannot be determined from a single snapshot, it must traverse through the network in both time and space. This is the key idea behind our storage technique. Information at a single time-step does not need to be stored together as a Lagrangian path will only access one node per time-step, therefore it is more efficient to store nodes connected through their temporal connections.

Lagrangian Adjacency List: <nodeId, timeInstant> All nodes connected to a node through both space and time as represented by an edge in a Time Expanded Graph.

4.2 Operations

The following operations are needed to solve simple and complex queries on a spatio-temporal road network.

Lagrangian Find: <nodeId, timeInstant> A node at a given time instant.

Lagrangian Duration: <startNodeId, timeInstant, destinationNodeId> A node at a given time instant.

Lagrangian Successor: <nodeId, timeInstant, successorNodeId> A specified node connected a node through both space and time as represented by an edge in a Time Expanded Graph. A successor node s to node n at time t can be represented as find(<s, t + duration(<n, t, s)>)).

Lagrangian Predecessor: <nodeId, timeInstant, predecessorNodeId> A specified node connected to a node through both space and time as represented by an edge in a Time Expanded Graph. A predecessor node p to node n at time t can be represented as find(<p, t - duration(<p, t, n)>)).

Lagrangian Adjacency List: <nodeId, timeInstant> All nodes connected to a node through both space and time as represented by an edge in a Time Expanded Graph.

4.3 Queries

4.2 Operations

The following operations are needed to solve simple and complex queries on a spatio-temporal road network.

Lagrangian Find: <nodeId, timeInstant> A node at a given time instant.

Lagrangian Duration: <startNodeId, timeInstant, destinationNodeId> A node at a given time instant.

Lagrangian Successor: <nodeId, timeInstant, successorNodeId> A specified node connected a node through both space and time as represented by an edge in a Time Expanded Graph. A successor node s to node n at time t can be represented as find(<s, t + duration(<n, t, s)>)).

Lagrangian Predecessor: <nodeId, timeInstant, predecessorNodeId> A specified node connected to a node through both space and time as represented by an edge in a Time Expanded Graph. A predecessor node p to node n at time t can be represented as find(<p, t - duration(<p, t, n)>)).

Lagrangian Adjacency List: <nodeId, timeInstant> All nodes connected to a node through both space and time as represented by an edge in a Time Expanded Graph.

4.3 Queries

4.3 Queries

The following operations are needed to solve simple and complex queries on a spatio-temporal road network.

Lagrangian Find: <nodeId, timeInstant> A node at a given time instant.

Lagrangian Duration: <startNodeId, timeInstant, destinationNodeId> A node at a given time instant.

Lagrangian Successor: <nodeId, timeInstant, successorNodeId> A specified node connected a node through both space and time as represented by an edge in a Time Expanded Graph. A successor node s to node n at time t can be represented as find(<s, t + duration(<n, t, s)>)).

Lagrangian Predecessor: <nodeId, timeInstant, predecessorNodeId> A specified node connected to a node through both space and time as represented by an edge in a Time Expanded Graph. A predecessor node p to node n at time t can be represented as find(<p, t - duration(<p, t, n)>)).

Lagrangian Adjacency List: <nodeId, timeInstant> All nodes connected to a node through both space and time as represented by an edge in a Time Expanded Graph.

5. SPATIO-TEMPORAL NETWORKS IN SECONDARY STORAGE

This section discusses both how the network is stored on disk pages, and how the network is partitioned for dividing nodes and edges for each page.

5.1 Lagrangian-based Storage

Disk Page Format: In our storage scheme, we mainly focus on an edge record format to store incident nodes and their edge weights. At first, we assign each node a unique number, called its ID. We store a source node id and outgoing edges from the source node into the same record, i.e. given a source id, incident node IDs are sequentially stored as an array. Figure 6 illustrates the variable-length edge record format. Since the number of out-going edges is variable, we need extra information, such as the number of incidents and bitmaps, to represent the variable-length records. In most road networks, the edge-to-node ratio is approximately close to 3 in most road networks, therefore, this approach reduces the duplicated source id to store the edge data and the extra overhead is justified by the added flexibility.

Figure 6: Format of a disk page used for storing the network.

Lagrangian-Connectivity Ratio: Depending on a node’s position in space and time, it may be part of multiple Lagrangian Paths. As defined above, a Lagrangian Cut is when a path is stored on more than one disk page, therefore, it is possible to measure the number of split paths that are coming into a particular node. Each node at each time instance (see nodes in Figure 4) is a member of some number of Lagrangian Paths. We can count the number of incoming Lagrangian Paths to that node, and also count the number of those incoming paths that are Lagrangian Cuts. We can use this to measure the connectivity ratio of that particular spatio-temporal node.

LCR(node) = \frac{\# of Lagrangian Paths - \# of Lagrangian Cuts}{Number of Lagrangian Paths}
Weighted Lagrangian-Connectivity Ratio: If we take the same equation above and add weights to the paths, due to some paths being more important than others, we can influence the disk page partitioning. For example, during rush hour, it would be prudent to value paths involving major highways.

5.2 Partitioning for Disk Storage
To achieve an efficient disk storage structure, we need to minimize the I/O cost to execute spatio-temporal network queries. In a database environment, the I/O cost to answer queries is determined by the number of pages which are transferred between disks and main memory. If data records topologically related for frequent access can also be stored physically into the same disk, the retrieval of records is reduced with fewer page accesses. Thus, the data clustering plays a crucial role to decrease the I/O cost of access method. An important observation is that as the storage space decreases, the I/O cost is also reduced due to fewer numbers of pages with the same data.

Spatio-temporal data clustering is not a straightforward task due to the complexity of incorporating temporal data into the network and requires careful analysis to satisfy both objectives. In spatio-temporal networks, the accessibility of data records is constrained by spatio-temporal network topology. It is therefore realistic to define the network model as time-expanded graph due to its relative simplicity to model the spatio-temporal network.

In the rest of this section, we will highlight the following proposed spatio-temporal network physical storage techniques. The first two represent naive methods, while the last two are our contributions.

5.3 Snapshot Partitioning
Spatio-temporal network data can be stored with different modeling approaches. One of the simplest model is the snapshot model where change of spatio-temporal network is described by each snapshot graph. However, the paradigm of the snapshot model, which produces a time stamped static graph at each time layer, leads to great I/O cost due to the need of accessing a new page for every time-step (see Figure 7 (a)).

5.4 Time-Series Partitioning
One interesting remedy to this problem is the Time Aggregated Graph (TAG) [11]. In the TAG model, only one pair of node ids are recorded to avoid data redundancy and time-varying information are stored as vector structure, while the snapshot model duplicates the redundant information, e.g., source node id and sink node id. This reduction relieves storage requirements substantially, and makes the computation of best routing problems less costly in the memory. Even though the time-aggregated graph meets the requirement of storage space problem, this solution, as shown in Figure 7 (b), suffers from the increasing disk I/O to evaluate the Lagrangian path in the secondary memory. This is due to the long time series being stored with each edge, allowing a small number of edges to be stored on each disk page.

The following are two techniques to store the spatio-temporal networks into a database system based on our Lagrangian-Connectivity Framework: the Aggregated Time-Stamped Snapshot Partitioning Model and the Lagrangian-Connectivity Partitioning Model.

5.5 Aggregated Time-Stamped Snapshot Partitioning Model
One way to incorporate the idea of Lagrangian Paths in disk storage is to slice the time-expanded graph into several graphs along the time dimension and then cluster graphs using the average weight of each edge within each slicing graph. Figure 8 (a) shows the result of this slicing. Since the network is sliced with respect to time intervals, we need to define a parameter as time intervals to slice the graph. The model is the simplest way to store Lagrangian paths, but its capability is limited. This is because it is difficult to determine a suitable value for this time window parameter. A naive approach is to slice the network with a fixed time interval (equi-width). However, the equi-width approach produces very poor results without considering the magnitude of travel-time changes. The strength of this model is that it is relatively practical when the travel time is not varying too much and the storage space is limited. The main disadvantage is that it is not possible to determine the appropriate time interval to yield a better performance. Especially, when a travel time abruptly and sharply changed, this model could not effectively capture this behavior.
5.6 Lagrangian-Connectivity Partitioning Model

The Lagrangian-Connectivity Partitioning Model is a network storage model to optimize disk storage based upon Lagrangian Paths to reduce I/O’s for Lagrangian Queries, such as single route evaluation queries. Simply put, it is a way of measuring frequently accessed routes, gathering edges where the successor follows as it moves along an individual route, and storing these edges into the same data page. A definition requires organizing network connectivity in terms of a degree of accessibility through time and network space, then to partition frequently accessed routes into the same disk page, while preserving two constraints, which minimizes the disk I/O and the storage space. We call such a partitioning a Lagrangian-Connectivity Partitioning. Since the min-cut constraint, by default, does not guarantee balanced partitions, a bi-criteria constraint is added to encourage more balanced partitions. Figure 9 illustrates such a Lagrangian model. A fundamental requirement to this design is a time bitmap to capture the irregular Lagrangian Path. As an example, Figure 8 (b) shows the data structure of a Lagrangian Path in the spatio-temporal network. A header file contains information about its incident nodes, bitmap, and the number of incidents. 64 bits bitmaps represent the existence of data in each position. Lagrangian model physically organize and store a collection of frequently accessed routes into the same disk page and captures spatio-temporal movements for most frequently used queries. In addition, it does not need to specify a parameter to cluster data. Lagrangian model shows its capabilities and efficiency to support a single-path routing evaluation.

6. VALIDATION

In this section we introduce the formal problem statement for our work.

Input: Acyclic graph consisting of nodes(N) and directed edges(E).

Time dependent travel time (T) on each edge.

Single-Route Queries

Output: Total travel time for each query.

Objective: Minimize I/O cost.

Constraints: Spatio-temporal network is too large to store in main memory.

Travel time is deterministic.

6.1 Experiment Setup

We implemented our databases in java and all tests were performed on an Intel Core(TM) 2 Duo CPU machine running Microsoft Window XP with 4GB of RAM. One of the

Figure 9: Bi-criteria constraint is added to encourage more balanced partitions

Figure 10: Hennepin County, MN road network.
main goals of our experimental analysis was to measure the performance of the proposed network data model on the real road networks. We first compared the I/O performances for a Lagrangian path computation. We also investigated how much various ingredients of our models contribute to the overall performance.

Our analysis used the Hennepin road map from U.S. TIGER/line road network data as shown in Figure 10[]. The data set consisted of 1,998 nodes and 6,938 edges. The travel time series were synthetically generated. The travel time of each edge was set to the Euclidean distance between two nodes multiplied by a random circular function value, e.g. sin(x) and cosine(x), with the range [1, M]. We set the number of time instants as 1,000 and the disk block size as 4K. We used Metis[12] to partition the network.

6.2 Results

![Figure 11: Travel Time Comparison](image)

(a) Given 1000 travel time, compare the performance according the route length

(b) Given different travel time, compare the performance according the route length

Figure 11: Travel Time Comparison

Experiment 1: Compare the performance of storage models according to the fixed travel time and different number of nodes in a route.

The purpose of this experiment was to evaluate the disk I/O according a fixed travel time and different number of nodes in a route. We used 100 random Lagrangian paths to evaluate the performance. We fixed the travel time as 1,000 time instants and varied the number of nodes in the path from 60 to 360. The experiment was done with five storage models: TAG (naive approach), ASTS with 10 time tags, ASTA with 40 time tags, ASTA with 70 time tags, and Lagrangian model. We used three different time tags for ASTA to examine the effect of the time series size. Figure 11 (a) shows the performance comparison using different network models. The number of data page accesses for route evaluation queries decreases with the increase the number of nodes in the route. Since each edge in a large number of nodes has small travel time, Metis clustered the short edge at first. This property is desirable when a user preferred a shorter edge to compute a trip route. As we stated before, TAG shows poor performance. Note that Lagrangian has stable performance compared to other models. Even though ASTA with 40 tags somewhat yields better performance in a large number of nodes in a route, it does show good performance in a small number of nodes. Since ASTA does not consider the network connectivity of the whole time series, it cannot capture the long travel time route connected by few edges. Moreover, it has the disadvantage that we had to identify a good parameter setting, such as time tag size, to store network data. This result justifies the quality of our Lagrangian showing the desirable behavior for a Lagrangian path evaluation.

Experiment 2: Compare the performance of storage models according to different travel times and different number of nodes in a route.

The purpose of this experiment was to understand the average performance on the introduced models with respect to different travel times and different lengths. The results of experiments are shown in Figure 11 ATSS with the moderate length (Length= 40) is expected to produce similar results compared to Lagrangian model, whereas it is inapplicable to all network data. Note that the length parameter of Aggregated Time-Stamped Snapshot is hard to define according to various traffic patterns.

Experiment 3: Compare the effect of partition criteria

We compared the partitioning performance between single-constraint and multi-constraints. Figure 12 shows that multi-constraints yields better performance. Since multi-constraints partitioning balances each page size with respect to the edge number, it can reduce the number of disk pages. We expect that as the number of disk pages decreases, the disk I/O decreases.

Experiment 4: Compare the effect of buffering In Lagrangian path evaluation, the buffer size does not improve the performance. Since our models cluster the next edge along the time expansion into the same disk page, it does...
not need buffers to store data for future requests.

7. CONCLUSION
Examining the problem of spatio-temporal networks in a popular but specific area, road network queries, we need new techniques to store and access this type of data. The proposed framework and subsequent operators and disk storage techniques attempt to handle the unique characteristics of spatio-temporal networks. We have shown that naive approaches to storing a spatio-temporal network are suboptimal. Our experiments provide validation for storing spatio-temporal networks with a Lagrangian approach.

8. REFERENCES