Towards a Tighter Bound on Possible-Rendezvous Areas: Preliminary Results

Arun Sharma
sharm485@umn.edu
University of Minnesota, Twin Cities
Minneapolis, Minnesota, USA

Jayant Gupta
gupta423@umn.edu
University of Minnesota, Twin Cities
Minneapolis, Minnesota, USA

Subhankar Ghosh
ghosh117@umn.edu
University of Minnesota, Twin Cities
Minneapolis, Minnesota, USA

ABSTRACT

Given trajectories with gaps, we investigate methods to tighten spatial bounds on areas (e.g., nodes in a spatial network) where possible rendezvous activity could have occurred. The problem is important for reducing manual effort to post-process possible rendezvous areas using satellite imagery and has many societal applications to improve public safety, security, and health. The problem of rendezvous detection is challenging due to the difficulty of interpreting missing data within a trajectory gap and the very high cost of detecting gaps in such a large volume of location data. Most recent literature presents formal models, namely space-time prism, to track an object’s rendezvous patterns within trajectory gaps on a spatial network. However, the bounds derived from the space-time prism are rather loose, resulting in unnecessarily extensive post-processing manual effort. To address these limitations, we propose a Time Slicing-based Gap-Aware Rendezvous Detection (TGARD) algorithm to tighten the spatial bounds in spatial networks. We propose a Dual Convergence TGARD (DC-TGARD) algorithm to improve computational efficiency using a bi-directional pruning approach. Theoretical results state the proposed spatial bounds on the area of possible rendezvous are tighter than that from related work (space-time prism). Experimental results on synthetic and real-world spatial networks (e.g., road networks) show that the proposed DC-TGARD is more scalable than the TGARD algorithm.

CCS CONCEPTS

• Information Systems; • Geographic Information Systems; • Computing Methodologies; • Spatial and Physical Reasoning;

KEYWORDS

Spatio-Temporal Data Analysis, Trajectory Data Mining, Spatial Modeling and Reasoning

ACM Reference Format:

Arun Sharma, Jayant Gupta, and Subhankar Ghosh. 2022. Towards a Tighter Bound on Possible-Rendezvous Areas: Preliminary Results. In The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3557915.3561033

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGSPATIAL ’22, November 1–4, 2022, Seattle, WA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9258-8/22/11...$15.00
https://doi.org/10.1145/3557915.3561033

1 INTRODUCTION

Given trajectories with gaps (i.e., missing data), we investigate methods to tighten bounds on the spatial networks (e.g., road network, river network, etc.) for detecting potential rendezvous or meetup locations. Figure 1 (a) shows a pair of trajectory gaps on an underlying spatial network topology where a gap exists from \( t = 2 \) to \( t = 6 \) for both trajectory 1 (blue) and trajectory 2 (red) with a given object speed at 1 unit/second. Current approaches output possible meeting locations of two objects via the intersection of two space-time prisms. For instance, Figure 1 (b) shows six interpolated nodes (green) that qualify as rendezvous locations for both objects. Figure 1(c) shows the result after a time slicing method has reduced the number of nodes (by a factor of 3), filtering out nodes \( N_{10}, N_{17}, N_{12}, \) and \( N_{19} \) (yellow) and leaving \( N_{11} \) and \( N_{18} \) (green) as the output. The resultant interpolated nodes are then sent to human analysts for further investigation.

Reducing the size of the space with possible rendezvous nodes is important for helping human analysts to detect and analyze trajectory data gaps. The smaller spatial area can then be more effectively verified with ground truth via satellite imagery to further aid early-stage decision-making. The problem has many societal applications related to homeland security, public health and safety etc. For instance, maritime safety involves monitoring activities such as illegal fishing and illegal oil transfers and transshipments [1]. Similarly, public health officials can analyze a given affected area where two objects could have met to control the potential spread of the disease. This paper focuses on the specific use case of improving public safety when two objects intentionally mask their movements to meet secretly within a trajectory gap area.

The problem is challenging since it is hard to model and interpret specific behavioral patterns, especially the rendezvous of two or more objects within a trajectory gap. Many methods rely on linear interpolation, which may lead to many missed patterns since moving objects do not always travel in a straight path. Methods based on the space-time prism are more geometrically accurate but identify large spatial regions, resulting in a time-intensive operation for the post-processing step performed by the human analyst. Further, such methods have a high computation cost due to large data volume. This work proposes computationally efficient time-slicing methods to effectively capture rendezvous regions with tighter spatial bounds in spatial networks.

The traditional literature [3, 4, 29] on the mobility patterns of objects in spatial networks considers realistic scenarios and events (e.g., traffic congestion) and other behavioral patterns [6]. However, little attention has been given to movement patterns within trajectory gaps. Most works in this area are limited to linear interpolation methods using shortest path discovery [5, 26]. Other works...
consider an object’s motion uncertainty via geometric-based methods (e.g., space-time prisms [13, 15, 17]) using spatial geo-ellipse boundaries constructed via motion parameters (e.g., speed). One recent work [25] does explicitly consider rendezvous or meetup queries in a spatial network, which the authors call assembly queries, but they used a loosely bounded geo-ellipse estimation from the space-time prism model. Our work proposes time-slicing methods to give tighter bounds on a given rendezvous region and provide further computational speedup using a dual convergence approach. More detailed related work can be found in Appendix B.

Contributions: The paper contributions are as follows:

- We propose a time slicing model and theoretically show that it provides a tighter bound on possible rendezvous area relative to traditional space-time prisms.
- Using the time slicing model, we propose a Time Slicing-based Gap-Aware Rendezvous Detection (TGARD) algorithm to effectively detect rendezvous nodes and a Dual Convergence TGARD (DC-TGARD) algorithm using bi-directional pruning to improve the computational efficiency.
- We provide a theoretical evaluation for both algorithms based on correctness, completeness, and time complexity.
- We validate both algorithms experimentally based on solution quality and computation efficiency on both synthetic and real-world datasets.

Scope: The paper proposes TGARD and DC-TGARD algorithms to tighten the spatial bounds of the rendezvous region in the spatial networks. We do not consider gaps with short time intervals (i.e., minutes, seconds etc.) or spatial areas that are low density or sparse. Kinetic prisms [14] fall outside the scope of the paper. In addition, we do not model the rendezvous of the objects in trajectories without gaps. The proposed framework has multiple phases (i.e., filter, refinement, and calibration), but we limit this work to the filter phase. The refinement phase requires input from a human analyst and is not addressed here, and calibration of the cost model parameters is outside the scope.

Organization: The paper is organized as follows: Section 2 introduces basic concepts, framework and the problem statement. Section 3 provides an overview of time slicing model. Section 4 describes the baseline TGARD and refined DC-TGARD algorithms. Section 5 evaluates the proposed algorithms theoretically on correctness, completeness and asymptotic complexity. Experimental evaluations are presented for both algorithms and related work are presented in Section 6. A broad and detailed literature survey is given in Appendix B. In addition, mathematical notations can be found in Appendix A. Finally, Section 7 concludes this work and briefly lists the future work.

2 PROBLEM FORMULATION

2.1 Framework

We aim to identify possible rendezvous locations in a given set of trajectory gaps through a two-phase Filter and Refine approach. We introduce an intermediate time slicing filter that reduces the number of interpolated nodes residing within the rendezvous region. The refinement phase further improves the solution quality so that human analysts may extract and analyze a comparatively fewer number of nodes involved in possible rendezvous by two (or more) objects. The inspection is further verified via satellite imagery to derive a possible hypothesis about the rendezvous activity (as shown in Figure 2).

Figure 2: Framework for detecting possible rendezvous locations

2.2 Basic Concepts

Definition 2.1. A spatial network is a set of nodes and edges, where each node $N$ is a geo-referenced point, while each edge $E$ has an edge weight $E_w$, i.e., the minimum time to travel from $N_i$ to $N_j$.

Figure 1 shows a spatial network where circles represent nodes (e.g., $N_i$) and the lines represent edges. A road system is an example of a spatial network where nodes are intersections, and edges are segments.

Definition 2.2. A effective missing period (EMP) is the actual time period between consecutive points of a given trajectory where no GPS location data is available.

For example, an object’s effective missing period can range from 15 mins to a couple of hours.

Definition 2.3. A trajectory gap ($G_i$) is the actual missing location information corresponding to an effective missing period (EMP). A trajectory gap is temporally considered when the EMP is greater than a certain threshold (θ).
A trajectory gap can be spatially modeled in the x-y plane using geo-ellipses using two parameters, i.e., maximum speed (MS) and effective missing period (EMP). Figure 3 shows a trajectory gap modeled as a spatially interpolated region in the form of a geo-ellipse (in blue) with \( C_1 \) and \( C_2 \) as the foci and a time interval greater than the EMP threshold with MS as the maximum speed. We assume that, most shorter trajectory gaps occur due to device malfunction or weather-based interference. Thus, for our analysis, we use the EMP threshold to filter out shorter trajectory gaps.

**Definition 2.4.** A Possible Rendezvous Region (\( R \)) is the common region within the intersection of two (or more) geo-ellipses derived from two trajectory gaps. It represents the region where a possible rendezvous between a pair of objects can occur.

Figure 4 shows a possible rendezvous region as the intersection of two geo-ellipses derived from Ellipse 1 and Ellipse 2 (i.e., blue outline region), providing euclidian bounds for a possible interaction of two objects.

**Definition 2.5.** Candidate Interpolated Nodes (\( N_u \)) are the set of interpolated nodes defined within a possible rendezvous region candidate (\( R \)) where each node is defined with the possible interaction of two (or more) objects.

Figure 1(b) shows interpolated nodes (in green) derived from the intersection of the two ellipses, from trajectory T1 and T2 [25]. Figure 1(c) shows a more refined time-slicing approach that considers candidate nodes at a given time instant \( t \) where the nodes lie within the red outline region (as shown in Figure 4).

**Definition 2.6.** An Object Availability Interval (\( Α \)) is the maximum time an object can wait at a given candidate interpolated node (\( N_u \)) within a rendezvous region. We calculate the difference between the shortest time to reach the start point (i.e., earliest arrival time) and endpoint (i.e., latest departure time) to a given Candidate Interpolated Node (\( N_u \)).

For instance, Figure 5 shows nodes N0 to N4 with object availability intervals [0,4], [2,6], [3,7], [4,8], and [6,10], where N0 and N4 are terminal nodes and nodes N1, N2, N3 are candidate interpolated nodes (\( N_u \)). The earliest arrival is calculated by adding \( E_w \) from the start time and departure time by subtracting \( E_w \) from the end time. More details with intuitive examples can be found in Section 4.

**Definition 2.7.** Possible Rendezvous Nodes (\( N_r \)) are the set of candidate interpolated nodes defined within the possible rendezvous region where a intersection of the availability intervals of two objects is non-empty (i.e., nodes where two objects are more likely to be met).

For instance, if another object is present on \( N_i \) with an availability interval of [3, 7], then the node is qualifies as a possible rendezvous node since [2, 6] \( \cap \) [3, 7] which is non-empty. Similarly, a subset of interpolated nodes in Figure 1c can be considered possible rendezvous nodes if both \( N_{11} \) and \( N_{18} \) have a non-empty availability interval. Such nodes can be further sent to human analysts for ground truth verification. More formal equations and details can be found in Section 4.

### 2.3 Problem Formulation

The problem to optimally identify a trajectory gap region in a spatio-temporal domain is formulated as follows:

**Input:**

1. A Spatial Network
2. A set of \( |N| \) Trajectory Gaps
3. Historic Traffic Data.

**Output:** A more tightly bound Possible Rendezvous Region

**Objective:** Solution Quality and Computational Efficiency

**Constraints:**

1. Trajectories have long gaps
2. Maximum Acceleration is not available
3. Correctness and Completeness

Figure 1(a) shows the input as a two-dimensional representation of a trajectory gap. Figure 1(b) shows the output based on the intersection of geo-ellipses resulting in the interpolated nodes in green. Figure 1(c) provides a more refined output resulting in a smaller number of interpolated nodes for human analysts to inspect.

### 3 TIME SLICING MODEL

Our time slicing model uses (a space-time prism) [17] to provide a detailed representation of an object’s physical space. We first describe space-time prisms and then describe the time slicing model.

Space-Time (ST) Prisms [17] are a collection of spatial points bounded by a physical space which is represented as an interpolated region where moving objects could have passed at a given maximum speed MS. Figure 3 shows the ellipse region in blue for a given time range \( [t_s, t_e] \) where \( t_s \) denotes the start time and \( t_e \) denotes the end time of the trajectory gap \( G_i \). Equation 1 defines the geo-ellipse with foci \((x_s, y_s)\) and \((x_e, y_e)\) for a missing period \( t_e - t_s \) as follows:

\[
\sqrt{(x - x_s)^2 + (y - y_s)^2} + \sqrt{(x - x_e)^2 + (y - y_e)^2} \leq (t_e - t_s) \times MS_1
\]

where, \((x_s, y_s)\) and \((x_e, y_e)\) are the start and end points of the trajectory gap with start time \( t_s \) and end time \( t_e \) \( (t_e > t_s) \). The ellipse spatially delimits the extent of a moving object’s mobility with maximum speed \( MS_1 \).

Figure 4 shows the possible rendezvous region where we perform the spatial intersection of two ellipses \( Ellipse_1 \in G_1 \) and \( Ellipse_2 \in G_2 \) and the time range is calculated via inequality 2:

\[
[t_s^{Ellipse_1}, t_e^{Ellipse_1}] \cap [t_s^{Ellipse_2}, t_e^{Ellipse_2}] \neq \emptyset
\]

![Figure 3: Time slicing model with a lens at time instant t](image)

A Time Slice is an object’s physical space sampled within a geo-ellipse (i.e., a ST-Prism) of a trajectory gap. A time slice bounds the spatial intersection of two circles \( C_1 \) and \( C_2 \) at a given time instant \( t \) where \( t_s \leq t \leq t_e \). Equations 3 and 4 represent the two
circles generated from \((x_i, y_i)\) and \((x_e, y_e)\) at given time instant \(t\) (where \(t_s \leq t \leq t_e\)) as follows:

\[
C_1 : (x_i - x_s)^2 + (y_i - y_s)^2 \leq (t - t_s)^2 M^2 S_1^2 \\
C_2 : (x_e - x_s)^2 + (y_e - y_s)^2 \leq (t_e - t)^2 M^2 S_2^2
\]

Figure 3 shows the spatial bounds of a time slice denoted as \(Lens_i\) which is defined as the geometry derived via \(C_1 \cap C_2\) at a given time instant \(t\). For instance, \(Lens_i\) within an ellipse \(Ellipse_{\mathcal{E}}\) is generated via maximum speed \(M_S\) at time \(t \in [t_s, t_e]\) i.e.,

\[
t_s \leq t \leq t_e \\
0 \leq M_S(t - t_s) \leq M_S(t_e - t_s) \\
0 \leq M_S(t_e - t) \leq M_S(t_e - t)
\]

Subtracting Inequality 6 and 7, we get Inequality 8, which provides the condition to define a lens \(Lens_i\) as follows i.e., whether the radii intersection of \(C_1 \cap C_2 \geq 0\).

\[
M_S(t - t_s) - M_S(t_e - t) \geq 0 
\]

The bounded region in Figure 3 shows \(Lens_i\) i.e., the intersection of \(C_1 \cap C_2 \geq 0\). The bounded rendezvous regions at a given time instant \(t\) are shown in Figure 4 with the intersection of \(Lens_{\mathcal{E}}\) and \(Lens_{\mathcal{L}} \subseteq Ellipse_1 \cap Ellipse_2\).

![Figure 4: Time slicing model with lens intersection at time instant t](image)

A rendezvous region using a time slicing model is defined as the intersection of \(Lens_{\mathcal{E}}\) and \(Lens_{\mathcal{L}}\), derived from \(Ellipse_{\mathcal{E}} \subseteq G_1\) and \(Ellipse_{\mathcal{L}} \subseteq G_2\). Figure 4 further represents the rendezvous region via \(Lens_{\mathcal{E}} \cap Lens_{\mathcal{L}}\) at a given time instant \(t\) which provides an even tighter bound as compared to the \(Ellipse_{\mathcal{E}} \cap Ellipse_{\mathcal{L}}\) Lemma 3.1 and Lemma 3.2 provide formal proofs to the Theorem 3.3 which states that a time slice is a subset of the intersection of two space time prisms.

**Theorem 3.3.** Given a pair \(<G_1, G_2>\) of trajectory gaps in an isometric euclidean space, the space-time prism model bounds the area of possible rendezvous by the intersection of two ellipses \(Ellipse_{\mathcal{E}}\) and \(Ellipse_{\mathcal{L}}\) where \(Ellipse_{\mathcal{E}}\) specifies the possible locations during gap \(G_1\) given maximum speed \(M_S\) and \(Ellipse_{\mathcal{L}}\) specifies the possible locations during gap \(G_2\) given maximum speed \(M_S\).

**Proof.** Given points \((x_i, y_i, t_s)\) and \((x_e, y_e, t_e)\) are focii of the ellipse \(E\), the according to Equation 1, a point \((x,y)\) can lie anywhere in the geo-ellipse such that the sum of the distance from focii \((x_i, y_i)\) and \((x_e, y_e)\) is \((t_e - t_s)\). Equation 13, derived from the addition from the Equations 3 and 4 of two circles, also denotes a property of the ellipse:

\[
(x_i - x)^2 + (y_i - y)^2 + (x_e - x)^2 + (y_e - y)^2 \leq 2(t_e - t_s)^2 \times M_S^2
\]

The left hand side of Inequality 13 also denotes each \(Lens_t\) is valid \(\forall t \in [t_s, t_e]\):

\[
\bigcup_{t=t_s}^{t_e} Lens_t \subseteq \bigcup_{t=t_s}^{t_e} (t_e - t_s)^2 \times M_S^2
\]

In addition, using Lemma 3.1, and 3.2, the areal bounds defined by \(Lens_t\) will not exceed the bounds defined by the geo-ellipse. □

These lemmas and theorems can easily be generalized to spatial networks (e.g., road networks) by generalizing the ellipses and lenses to the subgraphs reachable during a gap time interval and the subgraphs reachable at a particular time instant within a gap time interval respectively, given a gap-start node, gap-end node, maximum speed and a gap time-interval. Due to lack of space, we are omitting the detailed proofs.
4 PROPOSED APPROACH

In this section, we first explain some underlying concepts which are later used to define the output of the problem (i.e., possible rendezvous nodes) in spatial networks. We then discuss the required pre-processing steps related to gathering candidate trajectory gap pairs which are later temporally sampled to construct spatial sub-networks. Finally, we present our proposed TGARD and DC-TGARD algorithms which leverage time slicing properties for better solution quality and computational efficiency.

To calculate possible rendezvous locations, we first estimate the availability interval for each object and then calculate how early that object is able to reach (i.e., earliest arrival time) and the latest time the object can depart (latest departure time) at a given node. We formally define them as follows:

- **An Earliest Arrival Time** ($N_t^E$) is the minimum time object $O_i$ takes from start node $N_0$ to an intermediate node $N_u$ i.e., $N_t^E = N_t^u + E_w$. 
- **A Latest Departure Time** ($N_t^{EA}$) is the maximum time the object $O_i$ takes from end node $N_0$ to the intermediate node $N_u$ where $N_t^E = N_t^{LD} - E_w$. An availability interval ($a$) is the time interval that object $O_i$ waits in node $N_u$ such that $a(u) = [N_t^{EA}, N_t^{LD}] \neq 0$, where $E_A(u) \leq L_D(u)$, $N_0$ is defined as reachable when $a(u) \neq \emptyset$ and not reachable when $a(u) = \emptyset$ or $N_t^{EA} \leq N_t^{LD}$.

![Figure 5: An illustration of Possible Rendezvous Nodes](image)

Figure 5 shows Time Aggregated Graph (TAG) [8] where edge weights $[t_1, t_2]$ for nodes $N_0$ to $N_4$ are $[0.4, [2.6, [3.7], [4.8]]$ and $[6.10]$ respectively. Nodes $N_0$ and $N_4$ are start and end nodes respectively and nodes $[N_1, N_2, N_3] \in N_u$. The earliest arrival is calculated by adding $E_w$ to a given intermediate node $N_u$. For instance, $N_t^{EA} = N_t^{E} + E_w(N_0, N_1) = 0 + 2 = 2$. Similarly, the values of $N_t^{EA}$, $N_t^{LD}$ and $N_t^{EA}$ are $3, 4$ and $6$ respectively. In contrast, the latest departure is calculated by subtracting $E_w$ to a given intermediate node $N_u$. For instance, $N_t^{LD} = N_t^{E} - E_w(N_3, N_0) = 10 - 2 = 8$. Similarly, the values of $N_t^{LD}$, $N_t^{LD}$ and $N_t^{LD}$ are $3, 4$ and $6$ respectively.

In the case of multiple paths $P$ from a single source (i.e., $N_0$ or $N_4$), shortest path algorithms (e.g., Bi-directional Dijkstra's) or other breadth first search approaches are considered. Hence, multiple arrival times and late departure times need to be considered for a given node $N_0$ such that the earliest arrival time will be the minimum of all the arrival times derived via the shortest path from $N_0$ to $N_u$ (i.e., min($[N_t^{EA}, N_t^{EA}, \ldots, N_t^{EA}]$)). In contrast, the latest departure time will be the maximum of all the departure times that result via the shortest paths from $N_0$ to $N_u$ (i.e., max($[N_t^{LD}, N_t^{LD}, \ldots, N_t^{LD}]$)).

$$a(N_u) = [\min([N_t^{EA}, \ldots, N_t^{EA}]), \max([N_t^{LD}, \ldots, N_t^{LD}])]$$

(15)

A Possible Rendezvous Node ($N_0$) is defined as a possible node $N_u$ (or location) where two or more gaps belonging to different gaps could have physically met. For instance, the availability intervals of gaps $G_i$ and $G_j$ at $N_0$ are defined as $a(N_u, G_i)$ and $a(N_u, G_j)$ respectively. The rendezvous is feasible only if:

$$a(N_u, G_i) \cap a(N_u, G_j) \geq \text{TO}$$

(16)

where TO ($\neq \emptyset$) is defined as the Time Overlap Threshold.

If the above conditions are satisfied, the nodes are then forwarded to human analysts for manual inspection.

4.1 Constructing Spatial Sub-networks

Section 3 describes how time slicing in an unconstrained (i.e., euclidean) space narrows down the search space through the construction of a lens at given time instant $t$ to provide tighter bounds as compared to the entire geo-ellipse region.

In spatial networks, we restrict such unconstrained movements which closely resemble a vehicle’s mobility in a road network. However, we use the combination of geo-ellipses and lenses to geometrically restrict the object movement capabilities in a Manhattan search space, thereby providing an upper bound towards the mobility. In addition to tightening the spatial bounds, time slicing also captures the time-dependent properties of spatial networks, which more accurately model an object’s mobility in real-world scenarios.

Algorithm 1 first gathers trajectory gap pairs based on their temporal properties via a Filter and Refine approach to minimize the combinatorial computations. Then we temporally sample each trajectory gaps which captures a time-dependent edge which and could later affect the object’s overall mobility.

4.1.1 Extract Trajectory Gap Pairs: For effective gap collection, we use a plane-sweep approach to sort the trajectory gaps $G_i$ and then filter out gaps that are not involved in a rendezvous. First, we sort all gaps $G_i \in [t_1, t_2]$ based on the start time $t_1$ of the coordinates of $G_i$. Then, we filter gap pairs by checking the necessary condition i.e., whether their respective time ranges overlap. For instance, Gap $G_i$ and $G_j$ should satisfy their respective time ranges $[t_1^{G_i}, t_2^{G_i}]$ and $[t_1^{G_j}, t_2^{G_j}] \neq \emptyset$ (using Equation 2).

For the sufficient condition, we check if $G_i$ and $G_j$ are spatially intersecting such that $\text{Ellipse}_{G_i} \cap \text{Ellipse}_{G_j} \neq \emptyset$. If yes, then we save the resultant shape of $\text{Ellipse}_{G_i} \cap \text{Ellipse}_{G_j}$ and time range $[t_1^{G_i}, t_2^{G_j}] \cap [t_1^{G_j}, t_2^{G_j}] = [t_1^{R}, t_2^{R}]$ for creating spatial sub-networks.

4.1.2 Creating Spatial Sub-Networks: Now, we generate spatial sub-networks based on the qualifying nodes within the spatial (i.e., $\text{Ellipse}_{G_i} \cap \text{Ellipse}_{G_j}$) and temporal ($[t_1^{R}, t_2^{R}]$) constraints of the trajectory gap pairs $G_i, G_j$ as described in Section 4.1.1. To create a spatial network (SN), we perform a linear scan of (N) nodes and edges ($E) \in \text{Ellipse}_{G_i} \cap \text{Ellipse}_{G_j}$. We then generate edge weight $E_w$ by considering historic location traces residing within the time range of $[t_1^{R}, t_2^{R}]$ and calculate the time cost by taking average speed $\mu$ (MS) and dividing it by total edge distance between $N_i$ and $N_j$.

During time slicing, we capture the dynamic edge weights within $[t_1^{R}, t_2^{R}]$. First, we perform uniform sampling for a given temporal range into $N$ samples where $[t_1^{G_i}, t_2^{G_j}] \in [t_1^{R}, t_2^{R}]$. We then compute edge weights $E_w$ with given time range $[t_1^{R}, t_2^{R}]$ and filter out nodes which qualify within the Lens$_i$ where $N_i \leq N$. Hence for each time frame we generate subnetworks $SN_i$ with nodes $N_i$ edge weights $E_w \forall i \in [t_1^{R}, t_2^{R}]$.

Algorithm 1 summarizes the process of gathering trajectory gap pairs and creation of spatial networks in steps as follows:

**Step 1:** For a given trajectory gap $G_i$ in a Non-Oberved list, we first check if $G_i$ intersects with $G_j$ in the Observed List $\neq \emptyset$. If the
Algorithm 1 Spatial Sub-Networks with Rendezvous Gap Pairs

Input:
- Historic Trajectory Data (HTD)
- A set of Trajectory Gaps \{G_1,...,G_N\}
- A Spatial Network \(N\)
- A Sampling Rate \(K\)

Output:
- Spatial Sub-Networks List

1: procedure Sub-Networks Map
2: if \(G_j\) is Non-Oberved List do
3: for each: \(G_i \in \) Observed List do
4: if \(G_i \in \) Observed List do
5: \([S_{N_0}, S_{N_1}, ..., S_{N_N}] \leftarrow \) Sub-networks \((G_i \cap G_j, \) HTD, N, K\)
6: Sub-Networks Map \([-G_i, G_j]\) \leftarrow [S_{N_0}, S_{N_1}, ..., S_{N_N}]
7: \return Sub-Networks Map

Step 2: For a given trajectory gap pair \(G_i, G_j\) and sampling rate \(K\), we sample time interval \([t_R^j, t_R^i]\) in \(K\) samples such that each sub-network \(S_{N_0}, S_{N_1}, ..., S_{N_N}\) has its corresponding time-stamp \([t_R^1, t_R^2, ..., t_R^N]\). Using historic trajectory data HTD, we calculate edge weights \(E_w\) for each gap pair \(G_i, G_j\) where \(SN_i \in \{E_w, t^R\}\). We then filter out a Sub-Network Map which is later returned as the output.

Algorithm 2 Time Slicing Gap-Aware Rendezvous Detection (TGRAD)

The proposed TGRAD algorithm captures an object’s more realistic movements by considering parameters such as traffic congestion, which later affect the shortest path computation needed to calculate the earliest arrival and latest departure time of the given sampled sub-network \(S_{N_0}\) where each sub-network \(S_{N_0} \in [t^R, t^L]\) and \(L_{en}\). Intermediate nodes \(N_u\) is defined within \(L_{en}\) towards which the shortest path between the start and end nodes is computed to calculate \(\Delta t_{ed}\) and \(L_{en}\). However, we need to perform shortest path computation for every \(N_u\) of every sampled sub-network \(S_{N_0}\) and edge weights \(E_w\). In Figure 6 each edge represents time series edge weights \(E_{w_i}, E_{w_k}\), and \(E^t_{w_k} \in P_1 \) and \(E^t_{w_k} \in P_2\), which gives the shortest path from \(N_0\) to \(N_6\).

Figure 6: An illustration of Time Dependent Shortest Path

In Figure 6, \(P_1\) takes the path \(N_0, N_1, N_2, N_3, N_5, and N_6\) at \(t_0\) (since \(2 + 1 + 1 \geq 3\)). In contrast, \(P_2\) uses \(N_0, N_1, N_2, N_3, N_4, N_5\) and \(N_6\) since the path from \(N_1\) and \(N_2\) at \(t_1 = 5 \geq 2 + 1 + 1\) (i.e., \(N_1, N_2, N_3, N_4, N_5,\) and \(N_6\)). In this case, node \(N_4\) gets affected and the shortest path needs to be recomputed towards \(N_6\) at time \(t_2\). To capture such changes in edge weights in \(E_{w_i} \in t_i \) and \(E^t_{w_k} \in t_i + 1\), we use a difference metric \(\delta(E^{t_i}_{w_k})\) and compare it with a threshold \(\tau\) (i.e., whether \(\delta(E^{t_i}_{w_k}) \geq \tau\). If it is, then we recompute the shortest path towards that node. In Algorithm 2, we denote \(\delta(E^{t_i}_{w_k})\) as \(\delta(E^t_{w_k})\). For instance, in Figure 6, node \(N_6\) gets affected at time \(t = 2\) given \(\tau = 2\) such that \(\delta(E^t_{w_k}) \geq \tau\). Hence we recompute the shortest path at \(P_2\) from \(N_0\) to \(N_6\) via \(N_2\) and \(N_5\). If not, the current path remains the same for future computations and thereby retains its optimal sub-structure property for shortest path, stated as follows:

**Theorem 4.1.** Given \(G = (V, E)\) with edge weights \(E_w\). Let \(P = \{N_1, N_2, N_3, ..., N_k\}\) be the shortest path from \(N_1\) to \(N_k\) such that \(1 \leq i \leq j \leq k\). If \(P_{ij} = \{N_i, N_{i+1}, ..., N_j\}\) is the sub-path of \(P\) from \(N_i\) to \(N_j\), then \(P_{ij}\) is the shortest path from \(P_i\) and \(P_j\).

**Proof.** The proof is straightforward. If we decompose \(P_i, j\) into \(P_{ij}, P_{jk}\), where \(E_w(P_{ij}) + E_w(P_{jk}) + E_w(P_{jk})\) and assume a new path \(P_{ij}\) from \(N_i\) to \(N_j\) such that \(E_w(P_{ij}) \leq E_w(P_{ij})\). Then, \(E_w(P_i, j) + E_w(P_{ij}) + E_w(P_{jk})\) which contradicts the assumption that \(P\) is shortest path from \(N_i\) to \(N_j\).

**Algorithm 2 Time Slicing Gap-Aware Rendezvous Detection**

Input:
- Time Overlap Threshold \(TO\)
- Rest same as Algorithm 1

Output:
- Possible Rendevous Nodes Hmap \(N_u\)

1: procedure
2: if \(G_i, G_j\) and \(\delta(E^t_{w_k}) \geq \tau\) do
3: for each Sub-Network \(S_{N_0}\) in Trajectory Gap Pairs \(G_i, G_j\) do
4: Construct \(\text{Len}_{en}\) and \(\text{Len}_{en}\) for \(t^R, t^L\)
5: if \(\text{Len}_{en} \cap \text{Len}_{en}\) do
6: if \(\Delta t_{ed} \geq \tau\) do
7: \(N_{en} \leftarrow \text{Len}_{en} \cap \text{Len}_{en}\)
8: if \(N_{en} \) is reachable by Sub-Network \(S_{N_0}\) and \(\text{Len}_{en}\) then
9: Compute \(N_{en}^{DA}\) and \(N_{en}^{LDA}\) for both \(\alpha(N_{en}^{DA})\) and \(\alpha(N_{en}^{LDA})\)
10: if \(\alpha(N_{en}^{DA}) \cap \alpha(N_{en}^{LDA}) \rightleftharpoons \emptyset\) and \(\text{TO}\) then
11: \(\text{Rendezvous Node Map} \text{RMap}[N_{en}] \leftarrow N_{en} \in [t^R, t^L]\)

The TGRAD algorithm steps are as follows:

**Step 1:** First we compute a sub-network (SN) map for each gap pair \(G_i, G_j\) using Algorithm 1 and initialize a Rendezvous Node Map RMap \(\text{RMap} = \emptyset\). For each gap pair \(G_i, G_j\), we have \(k\) sub-networks \(S_{N_0}\) which generate \(\text{Len}_{en}\) from both \(G_i\) and \(G_j\) at \(t^R\). We then filter out \(N_{en}\) residing within the spatial boundary of \(\text{Len}_{en} \cap \text{Len}_{en}\). In addition, we also check if edge weights \(\delta(E^t_{w_k})\) for sub-network \(S_{N_0}\) have changed considerably by comparing them with threshold \(\tau\). If yes, then we consider all the affected nodes within \(\text{Len}_{en} \cap \text{Len}_{en}\) for computing early arrival and late departure times to preserve completeness. If not, we use the previous shortest path calculations for early arrival and late departure at \(t^L\) adhering to the optimal sub-structure property for a shortest path stated in Theorem 4.1.

**Step 2:** After gathering all \(N_{en} \in \text{Len}_{en} \cap \text{Len}_{en}\), we check if \(N_{en}\) is reachable (i.e., \(\alpha(N_{en}) \rightleftharpoons \emptyset\)) by checking the availability
interval $[N_{EA}^k, N_{LD}^k] \not\subset [t_k, t_{k+1}]$ for gap $G_i$ and $G_j$. If valid, we then check whether both the availability intervals of $G_i$ and $G_j$ and their intersection is $\neq \emptyset$ and $\geq T$. If yes, then $N_{EA}^k$ qualifies as $N_k$ and is saved in rendezvous node (RN) Map which later returned as output.

4.3 A Dual Convergence Approach (DC-TGARD)

Since the time slicing operation is computationally expensive, the Dual Convergence method leverages the symmetric property of the ellipse to efficiently reduce the iterations of time slicing used by TGARD while preserving correctness and completeness. We first define an early termination condition based on areal coverage of Lens, where $k \in [0, n]$ for a given time frame $t$, where $t \in [0, n]$. We then perform bi-directional pruning and compute $EA$ and $LD$ from both tail-end of the ellipse in parallel to improve computational efficiency.

**Ellipse Symmetry Property**: Given an ellipse with major and minor axis and center $(0,0)$, then its foci $(\pm c, 0)$ are equidistant from it’s origin at c. Hence, the areal coverage drawn from the lenses centered at $(\pm c, 0)$ will be the same, permitting an equal number of nodes within the spatial bounds of the lenses. For instance, Figure 7 (b) shows an equal number of nodes residing within the spatial bounds of $Lens_k$ and $Lens_{n-k}$ and using Lemma 3.3, time slicing does not leave out any other interpolated nodes (green) bounded within the ellipse, preserving the completeness of DC-TGARD.

![Figure 7: An illustration of TGARD vs Dual Convergence Method](image)

**Early Stopping Criterion**: The early stopping criterion holds when the property of nonmonotonicity (i.e., non-increasing) is violated during estimation of the areal coverage of lenses throughout every time slicing operation from $t_k$ to $t_{k+1}$. For instance, Figure 7 (a), shows increasing areal-coverage i.e., $A(Lens_{k}) \leq A(Lens_{k+1})$ (given $A(Lens_{k})$ as the areal coverage of lens $k$) at $t_k$ and $t_{k+1}$ respectively (where, $t_{k+1} \geq t_k$). This is due to the increasing (or non-decreasing) length of the minor axis which preserves the monotonicity until $Lens_{k+1}$ of the ellipse. After $Lens_{k+1}$, the areal coverage start decreasing, thereby violating the monotonic property i.e., ($A(Lens_{k}) \leq A(Lens_{k+1}) \geq A(Lens_{k+2})$) and, resulting in early stopping of the time slicing operation.

**Dual Convergence Approach**: We need to consider each time-slice operation to preserve completeness of the algorithm. The dual convergence approach introduces bi-directional pruning in conjunction to efficiently converge towards the early stopping criterion. Hence, a time-slicing operation is performed from both tail ends of the ellipse focii in parallel such that both lenses generated at $t$ and $t_{k+1}$ follows the property of ellipse symmetry. While performing bi-directional pruning, we check the early stopping criterion if $A(Lens_{k}) \geq A(Lens_{k+1})$ such that the monotonic property is violated. Figure 7 shows an example of dual convergence where baseline TGARD in Figure 7 (a) performs a linear time slicing operation from $Lens_k$, $Lens_{k+1}$ to $Lens_n$ covering all the interpolated nodes. Figure 7 (b) shows bi-directional pruning at $Lens_k$ and $Lens_{n-k}$ where $k \in [1,n]$. The areal coverage of $Lens_k$ i.e., $A(Lens_k)$ and $A(Lens_{n-k})$ is computed in parallel using the property of ellipse symmetry. If $A(Lens_k) \leq A(Lens_{k+1})$, we have an early stopping condition. For instance, in Figure 7 DC-TGARD terminates when $A(Lens_{n-k+1}) \geq A(Lens_{n-k})$ at $t=2$ as compared to TGARD in Figure 7 (a) which terminates at $t=3$. Hence, areal coverage pruning is more efficient in DC-TGARD as compared to TGARD.

**Algorithm 3 Dual Convergence Time Slicing Gap-Aware Rendezvous Detection (DC-TGARD)**

```
Input:
Same as Algorithm 2
Output:
Possible Rendezvous Nodes List $Nr$
1: procedure:
2: Use Algorithm 1 to construct Subnetwork Map $SN$ Map
3: Rendezvous Node Map $RNMap \leftarrow \emptyset$ and Max Overlap $\leftarrow 0$
4: Construct Sub Network $SN_k$ and $SN_{n-k}$ for $r^k$ and $r^{n-k}$ respectively
5: for each: Sub Network $SN_k$ and $SN_{n-k}$ in Trajectory Gap Pairs $<G_i, G_j>$ do
6: Construct $Lens^{G_i}_k, Lens^{G_i}_{n-k} \in r^k$ and $Lens^{G_j}_{k}, Lens^{G_j}_{n-k} \in r^{n-k}$
7: while Max Overlap $\neq \emptyset$ Area($Lens^{G_i}_k \cap Lens^{G_i}_{n-k}$) do
8: if $Lens^{G_i}_k \cap Lens^{G_i}_{n-k} \neq \emptyset$ and $\delta(E_w^k, E_{w-k}^k) \leq r$ then
9: $N_{EA}^k \leftarrow Lens^{G_i}_k \cap Lens^{G_i}_{n-k}$ and $N_{LD}^{n-k} \leftarrow Lens^{G_j}_{n-k} \cap Lens^{G_j}_{k}$
10: if $N_{EA}^k \in SN_k$ and $N_{LD}^{n-k} \in SN_{n-k}$ reachable and if $RNMap$ then:
11: Compute $N_{EA}^k, N_{LD}^{n-k} \in r^k$ and $r^{n-k} \in SN_k$ and SN_{n-k} respectively
12: if $\delta(E_w^k, E_{w-k}^k) \neq 0$ and $\delta(E_w^k, E_{w-k}^k)$ with threshold $r$. Then, rest of the steps are similar to TGARD except we simultaneously calculate all the variable for the early arrival, late departure and availability intervals.
13: Rendezvous Node Map $RNmap |N_{EA}^k| \leftarrow N_{EA}^k$ and $N_{LD}^{n-k}$
14: return Rendezvous Node Map $RNMap$
```

The DC-TGARD algorithm steps are as follows:

**Step 1**: After generating a Sub-Network Map (SN Map) via Algorithm 1, we first initialize the Rendezvous Node Map $RNMap$ and Max Overlap variable as $0$. We then generate sub-networks $SN_k$ and $SN_{n-k}$ for each $r^k$ and $r^{n-k}$ and generate $Lens_k$ and $Lens_{n-k} \subseteq SN_k$ and $SN_{n-k}$ respectively. We then check if $Lens^{G_i}_k \cap Lens^{G_j}_{n-k} \neq \emptyset$ and simultaneously check $\delta(E_w^k, E_{w-k})$ with threshold $r$. Then, rest of the steps are similar to TGARD except we simultaneously calculate all the variable for the early arrival, late departure and availability intervals.

**Step 2**: We update the Max Overlap variable with Area of $Lens^{G_i}_k \cap Lens^{G_j}_{n-k}$. Finally, we check the early termination condition (i.e., current $Lens^{G_i}_k \cap Lens^{G_j}_{n-k}$ such that $Lens^{G_j}_{n-k} \geq Max$ Overlap. If yes, then the loop has reached to the peak element and the algorithm terminates.

5 THEORETICAL EVALUATION

5.1 Correctness and Completeness

In this section, we provide a theoretical analysis of the correctness and completeness of the proposed algorithms.

**Lemma 5.1.** TGARD and DC-TGARD algorithms are correct.
We also compared the execution time of TGARD and DC-TGARD while computing the time slice operations and the early termination. The time complexity for each algorithm are as follows:

\[ O(\text{worse case}) \]

The correctness of the algorithm also depends on the extent of the overlap of the two availability intervals \( a(1) \in G_1 \) and \( a(2) \in G_2 \). For instance, a low overlap threshold results in multiple false positive rendezvous areas whereas a high overlap threshold results in more false negatives. Hence, both TGARD and DC-TGARD algorithms are correct for a given overlap threshold. □

**Lemma 5.2.** The TGARD and DC-TGARD algorithms are complete.

**Proof.** Assume a finite set of \([K]\) trajectories with maximum \(N\) number of finite points, resulting in a maximum bound of \([K] \times [N]\) finite trajectory points. Hence, a finite number of trajectory gaps \(G_i\) is generated during pre-processing, resulting in a finite number of operations will be performed by TGARD and DC-TGARD algorithms that terminate at a finite time. The correctness of the algorithm also depends on the extent of the overlap of the two availability intervals \(a(1) \in G_1\) and \(a(2) \in G_2\). For instance, a low overlap threshold results in multiple false positive rendezvous areas whereas a high overlap threshold results in more false negatives. Hence, both TGARD and DC-TGARD algorithms are correct for a given overlap threshold. □

### 5.2 Asymptotic Analysis

The time complexity for each algorithm are as follows:

**Exacting Candidate Gap Pairs:** For generating candidate pairs, \((\frac{K}{2})\) trajectories must be selected where each such pair will have \((N - 1) \times [N - 1]\) comparisons. Hence, the total number of comparisons will be \(\binom{K}{2} \times [N - 1] \times [N - 1]\), which results in an asymptotic worst case of \(O(\binom{K}{2} \times [N]^2)\).

**Creating Sub-Networks (SNs):** For a given subgraph \(SN\) with \(|E| \leq N \times N\) edges, the edge weights need to be calculated \([T]\) times where \(E_w\) is calculated at constant time. Hence the overall time complexity for calculating \(E_w\) is \(O(|E| \times [T])\).

**TGARD Algorithm:** Given a sub-network \(SN\) for \(N\) nodes, we perform a linear scan for each time slice \([T]\) to gather intermediate nodes \(N_{N_{max}} \leq N\) resulting in \(O(|K| \times [T])\) operations. For computing Early Arrival and Late Departure for \(N_{i}\), we run a bi-directional Dijkstra’s algorithm with \(O(|N_{i}| + |E|)\). Hence, the algorithm performs \(O(|K| \times [T]) + O(|N_{i}| \times [E])\) operations.

**DC-TGARD Algorithm:** The worst case of DC-TGARD will be similar to TGARD except we perform bi-directional searches while computing the time slice operations and the early termination condition guarantees the algorithm stops in \(T/2\) steps. This in turn reduces shortest path computations, which enhances computational efficiency in high density regions.

### 6 EXPERIMENTAL EVALUATION

The goal of the experiments was to validate the benefit of the proposed time slicing approach for reducing the search space of rendezvous detection. We evaluated the solution quality by comparing the proposed DC-TGARD against space-time prism methods [25]. We also compared the execution time of TGARD and DC-TGARD under different parameters. Details shown in Fig. 8.

**Real World Data:** We used the Geolife [30] dataset based on Beijing Road Networks where each location has latitude, longitude, and height with a variety of travel modes (e.g., walking, driving, etc.). In this paper, we limited our evaluation to driving patterns due to their accordance with road network topology (e.g., road segment and intersection). In addition, we further simulated certain trajectories by adding more objects and scenarios of rendezvous patterns to test effectiveness and scalability of the proposed methods.

**Synthetic Data Generation:** For the solution quality experiment, we lacked ground truth data (i.e., information on whether a node was involved in a rendezvous or not). Therefore, we evaluated the proposed algorithms on synthetic data derived from the Geolife dataset. First, we gathered trajectories on a fixed study area of the Beijing road network with mobility data. We then pre-processed the trajectory points with gap durations greater than 30 mins and randomly classified each gap based on whether \(N_i\) was reachable or was not using the proposed methods.

**Computing Resources:** 2.6 GHz 6-Core Intel Core i7 processor and 16 GB 2667 MHz DDR4 RAM.

#### 6.1 Solution Quality

To assess solution quality, we compared TGARD against baseline space-time prism approach on bounding efficiency and accuracy. We developed an efficiency metric called node pruning efficiency (NPE), to measure the tightness of the proposed filter in the trajectory gap pairs. We defined NPE as the ratio of the nodes of a total study area to the nodes within a bounded region (Eq. 17):

\[
NPE = \frac{\text{Nodes in Total Study Area}}{\text{Nodes in the Bounded Region}}
\]  

**Nodes Pruning Efficiency (NPE):** We fixed the study area and varied the nodes \((N)\), effective missing period (EMP), speed (MS) and number of objects \((O)\) within each trajectory gap \(G_i\). Figure 9 shows the results for the space-time prism approach and the proposed time slicing approach. As can be seen in Figure 9 (a), DC-TGARD has better node pruning efficiency (NPE) as we increase the number of nodes (since node density increases). A similar trend is seen in Figure 9 (b) except the effectiveness of time slicing is not as significant as we increase the number of objects of a fixed study area. Figure 9 (c) and 9 (d) again show the time slicing effectiveness is superior as compared the baseline approach as we increase the duration of the effective missing period (EMP) of gap.

**Accuracy:** We used synthetic data with manually labelled ground truth data about whether or not the objects were involved in rendezvous for a fixed number of objects (i.e., 2000) within a fixed network shape of 5000 nodes. More details can be found in Appendix C.

**1) Number of Objects:** We set the EMP to 4 hours, MS to 30 m/s, Time Overlap Threshold (T) to 30 mins and we varied the number of objects from 500 to 1250. Figure 10 (a) shows DC-TGARD gives a more accurate representation of the minimum travel time from the start node and end node of the trajectory gap compared to the space-time prism based method. This results in more accurate estimates.
Towards a Tighter Bound on Possible-Rendezvous Areas: Preliminary Results

SIGSPATIAL ’22, November 1–4, 2022, Seattle, WA, USA

6.2 Computational Efficiency

Figure 9: DC-TGARD performs better than a space-time prism method under different parameters of $N_r$ and reduces the number of false negatives, which are known to be quite common in space-time prism based approaches [25].

(2) Effective Missing Period (EMP): Next, we set the number of objects at 500, the number of nodes at $5 \times 10^4$, MS at 30 m/s, TO threshold to 30 mins, and varied the EMP threshold from 2 to 10 hours. As shown in Figure 10 (b), we find that DC-TGARD outperforms space time prisms. Higher EMP estimates increase the size of the geo-ellipses resulting in a higher rate of false positives.

(3) Speed: We kept EMP and number of objects and EMP constant at 4 and 1000 hours respectively, the TO threshold at 30 mins and increased the maximum speed MS from 10 to 50 in meter/second units. Figure 10 (c) shows DC-TGARD is more accurate as compared to ST-Prisms. The reason is that high speed outputs a greater number of potential rendezvous. This increases false positives but DC-TGARD filters out non-reachable nodes.

(4) Time Overlap Threshold (TO): We set the EMP at 4 hours, MS at 30 m/s and varied the TO threshold from 20 min to 100 mins. Again, DC-TGARD outperforms space-time prism as we increase the TO threshold (Figure 10 (d)). We also see that a greater TO threshold results in less $N_r$, resulting in more false negatives which are more accurately filtered out by DC-TGARD algorithm.

6.2 Computational Efficiency

We then compared the proposed DC-TGARD against the baseline TGARD algorithm on computation efficiency using number of nodes, number of objects, effective missing period, Speed MS and time overlap threshold (TO).

(1) Number of Objects: We set the EMP to 4 hours, MS to 30 m/s, Time Overlap Threshold (T) to 30 mins and we varied the number of objects from 500 to 2500. Figure 11 (a) shows that DC-TGARD always runs faster than the TGARD algorithm via dual-convergence and early stopping criterion.

(2) Effective Missing Period (EMP): We set the number of objects at 500, the number of nodes at $5 \times 10^4$, MS at 30 m/s, TO threshold at 30 mins, and varied the EMP from 30 to 90 minutes. As shown in Figure 11 (b), we find that DC-TGARD outperforms TGARD with increasing EMP. The bi-directional pruning of the dual convergence approach reduces the time slicing operations.

(3) Speed: We kept EMP and number of gaps constant at 4 hours and 1000 respectively, TO threshold 30 mins and increased the maximum speed MS from 10 to 50 in m/s units. Figure 11 (c) shows DC-TGARD is faster than TGARD. Hence, the dual convergence of DC-TGARD helps but not that significantly if we increase the MS. The reason is similar to EMP results since higher maximum speed produces larger ellipses except non-reachable nodes the non-reachable nodes decreases for both the algorithms which results in more shortest path computations for both algorithms.

(4) Time Overlap Threshold (TO): We kept the number of objects at 1000, the EMP at 4 hours and MS at 30 m/s, but this time we increased the TO threshold from 20 min to 100 mins. Again, Figure 11 (d) shows DC-TGARD outperforms TGARD as we increase TO threshold i.e., the dual convergence nature of DC-TGARD efficiently filters out more $N_r$ with fewer shortest path computations.

(5) Number of Nodes: First, we did a high density test by setting the number of objects to 1000 and varied the number of nodes from 500 to 2500. Then we did a low density test, where we set the number of objects to 100 and varied the number of nodes from 200 to 1000. For both experiments, EMP was set to 4 hours, MS to 30 m/s and Time Overlap Threshold (TO) to 40 mins for a fixed study area. Figure 5 (e) and (f) shows that DC-TGARD always outperforms the TGARD algorithm. It’s time slicing operations are effective for both high and low density in spatial networks including road networks.

7 CONCLUSION AND FUTURE WORK

We study the problem of identifying a set of possible rendezvous locations within a trajectory gap in a given spatial network. We theoretically study a time slicing model which provides tighter bounds
We also want to thank Kim Koffolt and the spatial computing research group for their helpful comments and refinements.

ACKNOWLEDGMENTS

This research is funded by an academic grant from the National Geospatial-Intelligence Agency (Award No. HM0476-20-1-0009, Project Title: Identifying Aberration Patterns in Multi-attribute Trajectory Data with Gaps). Approved for public release, 22-536. We also want to thank Kim Koffolt and the spatial computing research group for their helpful comments and refinements.

REFERENCES

[1] BBC. 2017. North Korea: South seizes ship amid row over illegal oil transfer. https://www.bbc.com/news/world-asia-42507883.
[2] Su Chen et al. 2008. ST2B-tree: a self-tunable spatio-temporal B+-tree index for moving objects. In Proceedings of the ACM SIGMOD 28–42.
[3] Ugur Demiryurek et al. 2011. Online computation of fastest path in time-dependent spatial networks. In SSDT Springer, 92–111.
[4] Bolin Ding, Jeffrey Xu Yu, and Lu Qin. 2008. Finding time-dependent shortest paths over large graphs. In Proceedings of the 11th international conference on Extending database technology: Advances in database technology 205–216.
[5] Hui Ding, Gocce Trajevski, and Peter Scheuermann. 2008. Efficient similarity join of large sets of moving object trajectories. (2008), 79–87.
[6] S. Dodge, R. Weibel, and A. Lautenschütz. 2008. Towards a taxonomy of movement patterns. Information visualization 7, 3–4 (2008), 240–252.
[7] Joshua S Greenfield. 2002. Matching GPS observations to locations on a digital map. In Transportation Research Board 81st Annual Meeting, Vol. 22.
[8] Venkata Gunturi et al. 2011. A critical-time-point approach to all-start-time lagrangian shortest paths: A summary of results. In International Symposium on Spatial and Temporal Databases. Springer, 74–91.
[9] Hyun-Mi Kim and Mei-Po Kwan. 2003. Space-time accessibility measures: A geocomputational algorithm with a focus on the feasible opportunity set and possible activity duration. Journal of geographical Systems 5, 1 (2003), 71–91.
[10] George Kollors et al. 1999. On indexing mobile objects. In Proceedings of the eighteenth Symposium on Principles of Database Systems. 261–272.
[11] John Krumm, Robert Gruen, and Daniel Delling. 2013. From destination prediction to route prediction. Journal of Location Based Services 7, 2 (2013), 98–120.
[12] John Krumm and Eric Horvitz. 2006. Predestination: Inferring destinations from partial trajectories. In Ubicomp Springer, 243–260.
[13] Bart Kuipers, Rafael Grimson, and Walied Othman. 2011. An analytic solution to the alibi query in the space–time prisms model for moving object data. International Journal of Geographical Information Science 25, 2 (2011), 293–322.
[14] R. Kuipers, H. J. Miller, and W. Othman. 2017. Kinetic prisms: incorporating acceleration limits into space–time prisms. FGIS 31, 11 (2017), 2164–2194.
[15] Bart Kuipers and Walied Othman. 2009. Modeling uncertainty of moving objects on road networks via space–time prisms. FGIS 23, 9 (2009), 1095–1117.
[16] Yin Lou et al. 2009. Map-matching for low-sampling-rate GPS trajectories. In 17th ACM SIGSPATIAL. 352–361.
[17] H. J. Miller. 1991. Modelling accessibility using space–time prism concepts within geographical information systems. FGIS 5, 5 (1991), 287–301.
[18] Jignesh M Patel, Yun Chen, and V Prasad Chakka. 2004. STRIPES: an efficient index for predicted trajectories. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data. 635–646.
[19] A Sharma et al. 2020. Analyzing trajectory gaps for possible rendezvous: A summary of results. In 11th International Conference on Geographic Information Science (GIScience 2021)-Part I Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
[20] A. Sharma, J. Gupta, and S Shekhar. 2022. Abnormal Trajectory-Gap Detection: A Summary (Short Paper). In 15th International Conference on Spatial Information Theory (COSIT 2022) Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
[21] A Sharma and S Shekhar. 2022. Analyzing Trajectory Gaps to Find Possible Rendezvous Region. ACM Transactions on Intelligent Systems and Technology (TIST) 13, 2 (2022), 1–23.
[22] Gocce Trajevski. 2003. Probabilistic range queries in moving objects databases with uncertainty. In Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access. 39–45.
[23] Gocce Trajevski et al. 2004. Managing uncertainty in moving objects databases. ACM TDEIS 29, 3 (2004), 463–467.
[24] Gocce Trajevski et al. 2010. Uncertain range queries for necklaces. In 2010 Eleventh International Conference on Mobile Data Management. IEEE, 199–208.
[25] Rez Uddin, Michel N Rice, Chintu Y Ravishankar, and Vassilis J Tsotras. 2017. Assembly queries: Planning and discovering assemblies of moving objects using partial information. In Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 1–10.
[26] Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, and Guang-Zhong Sun. 2010. An interactive-voting based map matching algorithm. In 2010 Eleventh international conference on mobile data management. IEEE, 43–52.
[27] Kai Zheng, Gocce Trajevski, Xiaofang Zhou, and Peter Scheuermann. 2011. Probabilistic range queries for uncertain trajectories on road networks. In Proceedings of the 14th International Conference on Extending Database Technology. 283–294.
[28] Kai Zheng, Yu Zheng, Xing Xie, and Xiaofang Zhou. 2012. Reducing uncertainty of low-sampling-rate trajectories. In 2012 IEEE 28th ICDE IEEE, 1144–1155.
[29] Y. Zhang. 2015. Trajectory data mining; an overview: ACVM Transactions on Intelligent Systems and Technology (TIST) 6, 3 (2015), 1–41.
[30] Yu Zheng et al. 2010. GeoLife: A collaborative social networking service among user, location and trajectory. IEEE Data Eng. Bull. 33, 2 (2010), 32–39.
A TABLE OF NOTATIONS

| Notations | Description |
|-----------|-------------|
| N         | Nodes (or vertices) representing road intersections |
| E         | Edges representing road segments |
| $E_u$     | Edge weights (i.e., minimum time to traverse $E$) |
| $\theta$ | Minimum Effective Missing Period threshold |
| $O_i$     | Moving object |
| $G_t$     | Trajectory gap with Minimum Effective Missing Period $\geq \theta$ |
| $MS$      | Maximum Speed attained within the geometric gap $G_t \in G_i$ |
| $t$       | Time stamp within a given trajectory gap $G_t$ |
| $s_1$     | Start point of a trajectory gap $G_t$ (i.e., $x_s$, $y_s$, and $t_1$) |
| $e$       | End point of a trajectory gap $G_t$ (i.e., $x_e$, $y_e$, and $t_2$) |
| $N_p$     | Candidate interpolated node within a trajectory gap $G_t$ |
| $a(N_p)$  | Availability Interval of a given candidate interpolated node $N_p$ |
| $R$       | A set of Possible Rendezvous Node where $a(N_p) \neq \emptyset$ |
| $\Delta W_{ij}$ | Change in Edge Weight $E_{ij}$ within time interval $[t_i, t_{i+1}]$ |
| $\tau$   | Aggregated Edge Weight Threshold |

B OTHER RELATED WORK

The literature of trajectory mining includes a broad overview of movement patterns [6] and a taxonomy of spatial mining methods used in various application domains [29]. In trajectory data management, specific frameworks have analyzed trajectory gaps via indexing methods (e.g., hierarchical trees [2, 10] and grids [18]) but they are not designed to detect movement patterns. Other works have modelled regions of uncertainty [22, 23] via snapshot models etc. More realistic solutions are based on geometric models such as cylindrical [24] and space-time prism models [9, 13, 17] that construct an areal interpolation of the gaps using coordinates and a maximum speed of the objects. However, no studies have addressed rendezvous behavior patterns within trajectory gaps.

In spatial network research, map matching techniques are prevalent while modeling uncertainty in trajectories. For instance, in [16], the authors consider uncertainty by proposing ST-matching to map-match potential routes for low-sampling-rate GPS trajectories. The authors further incorporate spatial and temporal analysis, which are not considered by traditional methods [7]. A similar study captures uncertainty for low-sampling rate GPS trajectories using an interactive voting system [26] that incorporates geometric, topological structure, and speed constraints. The algorithm first retrieves a set of candidate gaps for each sampling point, then defines a similarity function for every two sequential sampling points and their candidate points.

Other works capture uncertainty without map-matching by using certain deterministic methods based on geographic properties of space-time in road network topology. For instance, [12] performs range queries for uncertain trajectories using the probability distribution function as a function of speed and time. In addition, [11] extended their work towards low sample trajectories using a traversed graph-based approach and a nearest neighbor-based approach. Two recent studies [27, 28] used route prediction and inference-based methods to model uncertainties within the trajectory where location signals are missing for a long period of time. However, none of the above methods consider rendezvous patterns within spatial networks.

Recent papers consider rendezvous detection in a euclidean space [19, 21] and static-spatial network [25] by modeling the uncertainty region using the geo-ellipse property of space-time prisms [20]. In [25], the authors defined the geo-ellipse region as type-2 uncertainty where the foci of the ellipse denote the start time and end time of the trajectory gap. However, such type-2 uncertainty provides a rather loose bound, resulting in many possible rendezvous nodes for human analysts to post-process for ground-truth verification. In addition, [25] was also limited to static-spatial networks and does not consider dynamic travel time estimates on edge weights which are more common in a real-world setting (e.g., traffic congestion). Figure 1b shows how the traditional space-time prism provides a loose bound as compared to the time slicing method (shown in Figure 1c), which includes a tighter bound by including a finer temporal sampling rate. Time slicing also captures dynamic time-dependent edge weights for more accurate computations of possible rendezvous nodes resulting in better solution quality.

Figure 12: A Comparison of Contribution with Related Work

In this work, we proposed a time-slicing based method to provide a tighter bound over traditional geo-ellipse estimation methods [25] and provide theoretical justification. This results in relatively fewer possible rendezvous nodes for the post-processing phase. In addition, we consider time-dependent edge weights to capture real-world settings, providing a more accurate estimation of possible rendezvous nodes and thereby improving solution quality. We propose a Time Slicing-based Gap-Aware Rendezvous Detection (TGARD) algorithm to effectively detect rendezvous nodes and a Dual Convergence TGARD (DC-TGARD) algorithm using bi-directional pruning to improve the computational efficiency. We provide theoretical evaluation for both algorithms based on correctness, completeness, and time complexity. Finally, we validate both algorithms based on solution quality and computation efficiency on both synthetic and real-world datasets.

C ACCURACY

The term accuracy is defined in Equation 18 as ratio of total number of actual objects involved in rendezvous by total number of rendezvous nodes output from the proposed algorithm.

$$\text{Accuracy} = \frac{\text{Actual Nodes involved in Rendezvous}}{\text{Total Nodes involved in Rendezvous}}$$ (18)