Proposal's Category	Category II
Proposal's Title	GraphTempo: Exploring the History of Temporal Graphs
Proposal's Acronym	GraphTempo
Scientific Area	Mathematics and Information Sciences
Scientific Field 1	Computer Sciences, Informatics and Bioinformatics
Scientific Field 2	–
Project's Duration (in months)	36
Total Budget (€)	200,000
Principal Investigator (PI)	Evaggelia Pitoura
Host Institution	University of Ioannina
List of Cooperative Organizations	University of Warwick, UK
	Boston University, USA

Part 6.A: Research Proposal

1. Excellence

1.1. Proposal's main goals, objectives and challenges

Graphs offer a natural model for representing entities and the interactions and relationships between them. In collaborative networks, graph edges capture the cooperation between actors in movies, authors of scientific articles, or co-workers in teams. In social networks, edges express the relationship (e.g., friend, follower) as well as the interactions and reactions (e.g., retweets, likes) between users. In communication networks, edges indicate email and phone exchanges between people, in transportation networks, roads and flights between cities, and in biological networks, interactions between proteins. Other examples include knowledge graphs, program dependency graphs, computer networks, and the web to name just a few more.

Most real-world graphs are not static but evolve through time. New interactions and relationships are created, while existing ones may no longer be valid. In addition, new entities are created, while old ones leave the network. Furthermore, the content associated with the graph structure, such as labels on vertices, or weights on edges, also changes with time.

The ***long term vision*** and ***overall goal*** of the *GraphTempo* project is to provide a declarative framework for exploring the past history of a graph that will act as a tool for predicting and potentially affecting the future evolution of the graph. *GraphTempo* aims at improving our understanding of real world graphs, such as social and collaboration networks, and thus enabling us to foresee and improve their future operation.

There is abundant information in the evolution of the graph not to be found in just the current snapshot of the graph. As a simple example, consider the fundamental task of selecting important nodes in a graph, based, for example, on their centrality value (e.g., degree, or PageRank). Looking only at the current snapshot of the graph, two vertices with the same centrality value will be considered equally important. However, centrality values fluctuate over time, and the popularity of one of the two vertices may have been constantly dropping, while the popularity of the other one may have been constantly rising. Furthermore, there is rich information in the co-evolution of content and structure. For example, do the properties of neighboring nodes tend to converge to similar values over time, thus exhibiting a phenomenon known as homophily?

Looking into the whole evolution of the graph may reveal hidden properties, alert us in cases of unexpected changes and help us identify spurious behavior. Considering the past history of the graph improves our understanding of the network, enhances the analysis for the data scientists and it is instrumental in making predictions for the future evolution of the graph.

Towards achieving its overall goal, *GraphTempo* sets the following ***specific goals:***

1. to offer a general framework for the exploration of the full history of a graph through time including its structure (vertices and edges) and content (e.g., labels, properties, weights) associated with this structure. The framework will offer a declarative approach to temporal graph processing consisting of a set of novel exploration queries.

2. to reveal hidden properties of real-world networks through the application of the framework, and

3. to improve the evolution of networks by proposing novel link recommendation algorithms.

Many ***challenges*** need to be addressed to achieve these goals. Such challenges include all 3Vs of big data, that is volume (most real world graphs have billion, even trillions, of vertices), velocity (the amount and speed of change in some networks may be huge) and variety (graphs vary both in structure and content). Another challenge is how to concretely explore, analyze and interpret the nature of change. Yet another challenge is capturing and understanding the interplay between structural and content change.

To achieve the first goal and address these challenges, the project will propose two general families of queries, namely graph evolution queries and graph journey queries. In the following, we use the term graph element to refer to an edge, node, value, or, subgraph. *Graph evolution* queries will focus on (1) transformations, that is, identifying sets of graph elements that follow specific evolution patterns through time, and (2) change, that is, discerning graph elements based on the volume of change. *Graph journey* queries will concentrate on journeys, defined as an *ordered set* of graph elements, where a graph element with order i must appear earlier in time than a graph element with order $j > i$. For example, a time-ordered set of edges may be used to model the path that information flow (e.g, a rumor) follows in a network. We will look into both strict and partial orders on sets of graph elements.

To address the volume, velocity and variety of graph data, *GraphTempo* will propose appropriate query processing algorithms for all types of queries. Novel index structures will be introduced based on temporal graph embeddings. Finally, the *GraphTempo* system will extend one of the current graph processing systems with the proposed novel queries.

To achieve the second goal, *GraphTempo* will apply the framework to specific real-world networks. Envisioned applications of graph evolution queries include (1) improving team formation in network by identifying stable cooperations, (2) spotting homophily in social and cooperation networks, and (3) anomaly detection (such as spam behavior) in social networks. Potential applications of graph journey queries include (1) identifying information cascades in social networks, (2) locating spurious transaction in financial networks such as in bitcoin, and (3) characterizing forged information propagation (e.g., fake news).

To achieve the third goal, *GraphTempo* will develop a novel link recommendation algorithm. The porposed algorithm will be based on temporal graph embeddings that will take advantage of the novel queries proposed by *GraphTempo*. Besides recommendation accuracy, the link recommendation algorithm will consider additional constraints towards improving specific properties of the network, such as diversity, to mitigate the effects of homophily.

In a nutshell, the ***specific objectives*** of the *GraphTempo* for materializing its three specific goals are summarized as follows:

- introduce novel graph evolution queries, query processing algorithms, index structures and real-world applications,
- introduce novel graph journey queries, query processing algorithms, index structures and real-world applications,
- extend the functionality of current graph processing systems with these novel queries through the *GraphTempo* system, and
- explore the novel queries to propose a network-aware link recommendation algorithm.

1.2. State-of-the-art

Current data management research related to historical graphs mainly focuses on two axes, namely, on efficiently storing and retrieving graph snapshots (where each snapshot corresponds to a state of the graph at some instant in time) (Axis (1)) and on applying graph queries to one of more graph snapshot (Axis (2)). A recent survey on the topic can be found in [46].

Along *Axis (1)*, the main focus is on how to maintain the whole history of a graph through time so that to minimize both the required amount of storage and the time needed for retrieving any of the snapshots. To this end, DeltaGraph [21, 22] proposes a hierarchical index structure. The leaves of the DeltaGraph index structure correspond to (not explicitly stored) graph snapshots, whereas the internal nodes to graphs constructed by combining the lower level graphs. Partial reconstruction, that is, constructing only the required subgraphs of past graph snapshots, is discussed in [25, 24, 27]. The authors in [48, 49] introduce a Find-Verify-and-Fix (FVF) approach where graph snapshots are clustered and representative graphs are stored for each cluster. A query involving multiple snapshots is initially executed only on the representatives. The results of the query are then verified, and the query is re-executed on any of the individual snapshots for which the verification failed.

Numerous systems have been proposed for managing graphs (see, for example, [18] and [67] for related surveys). We can distinguish graph systems into two general categories, namely, graph databases and general graph systems, where graph databases provide out of core storage of graphs, offer OLTP queries and transaction, while general graph systems are tailored to OLAP type of graph analysis. There has been some research on building new [13, 38] or extending existing [37, 17] parallel graph processing systems. Previous research also considers storing snapshots in native graph databases [54, 53, 8, 9], in relational graph databases [40, 41], and building new multi-snapshot graph databases [29].

Along *Axis (2)*, the research focus is on how to apply efficiently existing types of graph queries to a single past snapshot, or to multiple past snapshots. For example, in the case of a shortest path distance query between two vertices, the focus is on how to apply this query efficiently on multiple snapshots. Then, the query aggregates the results at each snapshot by returning for example, the average, or minimum distance value of the values computed at the snapshots involved in the query. Graph queries considered so far include reachability queries [56], graph patterns queries [52, 55, 70] and shortest path and distance queries [14, 1, 15]. In addition the authors of [48, 49] explore the FVF framework for distance queries [48, 49], while the authors of [61] propose a general framework for the efficient application of the steps of any algorithm in multiple snapshots.

However, all previous work on both axes focuses on applying known graph queries on one or more past snapshot. *GraphTempo* will go beyond applying graph queries to past snapshots and introduce novel interactive queries to explore the evolution of a graph through time.

Extracting knowledge from historical graphs has been studied from a data mining perspective, for example, there is previous work on finding frequent subgraphs and clusters. In this context, research includes among others, a particle-and-density based evolutionary clustering method [23], an optimization-based approach for modeling dynamic community structure [60], extending frequent item-set mining to graph patterns (eg., [4]), finding heavy subgraphs that maximize edge weights [6], identifying dense subgraphs over time [57], and extending temporal association rules from items to graphs [42].

There is also work on generative model for evolving graphs (e.g., the influential work of [33]), studies of graph metrics and graph-theoretic problems (e.g., connectivity [20]) and various analytics (e.g, Pagerank [50]).

As opposed to such data mining approaches, *GraphTempo* proposes a *declarative query-based* exploration of the history of a graph.

Sometimes historical graphs are called *temporal* graphs. Here, we shall use the term "*temporal* graph" for graphs where edges model interactions that have a duration, in particular, a starting time t_s and duration time δ is associated with each edge between two vertices u and v [65]. A *temporal path* or *journey* in a temporal graph is a sequence of vertices $u_1 u_2 \dots u_k$ where $t_s(u_i, u_{i+1}) \geq t_s(u_{i-1}, u_i) + \delta(u_{i-1}, u_i)$ for $1 < i < k$.

Previous research on querying temporal graphs includes minimum distance [65] and reachability queries [66] for different definitions of minimum temporal paths [65]. Recent work on temporal graphs also include finding motifs in temporal graphs [43] including motifs with flow constraints [28]. A motif is a subgraph that appears significantly more often in a real network than in a randomized network with similar characteristics [39].

In *GraphTempo*, we will support declarative querying of the evolution of temporal graphs. To our knowledge, there is no related research on the topic.

Finally, there is a long line of research on temporal databases (e.g., [16]). *GraphTempo* will build on such research to provide different notions and semantics of time.

1.3. Scientific methodology & novel aspects

There is rich information in the history of a graph. *GraphTempo* goes beyond maintaining versions, or, snapshots of the graph and applying queries on them. *GraphTempo* aims at offering a declarative framework to support the exploration of the evolution of the graph through time and the extraction of interesting information from this evolution.

In the following, we first provide a description of the proposed research work and methodology and highlight its novel aspect. Then, we present an overview of our scientific methodology.

1.3.1 Proposed scientific work and novel aspects. Graphs are ubiquitous as a modeling abstraction as they ofter a natural model for representing connected entities and their interactions and relationships. Most graphs are not static, but evolve with time. As reported in [51], one of the future challenges in graph data management is maintaining and querying the history of a graph. In general, exploring change is envisioned to advance data science in many ways [5]. *GraphTempo* addresses this exact need and vision.

A *graph* is typically represented as an ordered pair $G = (V, E)$ of a set V of *vertices* and a set $E \subseteq V \times V$ of *edges*. In some cases, *data*, or, *content* is attached to the graph structure. The simplest form of content is a set of values, called *labels*, associated with vertices, edges or both. A special case of a labeled graph is a *weighted* graph where labels take numerical values. Often, labels are used to associate semantics with vertices and edges, for example, to give vertices and edges a *type*. Finally, content may take the form of (attribute, value) pairs called *properties*. In the *GraphTempo* project, we will consider all these different forms of content. In the following, we use the term graph element to refer to an edge, node, piece of content, or, subgraph.

For simplicity, we assume a linearly-ordered discrete time domain, use consecutive integers to represent consecutive time instants and now to represent the current time. We use the term historical graph to refer to the sequence $\mathcal{G}_{\mathcal{G}}$ = $\{G_1, G_2, \ldots, G_{now}\}$ of graph snapshots, where each G_t captures the state of the graph (i.e., vertices, edges and content) at time instant t.

One of the main goals of *GraphTempo* project is the development of a general framework for the exploration of the evolution of a graph through time. The framework will offer a declarative approach to temporal graph processing consisting of a set of exploration queries. This is a novel aspect, since most previous work focuses on applying queries on one, or more, specific snapshots (e.g., see [46] for a survey). In the course of the project, we will propose and formally define two innovative families of queries, namely graph evolution queries and graph journey queries.

Graph evolution queries will focus on (1) transformations, that is, identifying sets of graph elements that follow specific evolution patterns through time, and (2) change, that is, identifying graph elements based on the volume of change.

Through a *transformation query*, we would like, for example, to track a set of graph elements that start from an initial formation and then evolve to a new formation through time. An example such query could be specified, for example, by using what we can call a *before-and-after query*, or, *B&A*-query, for short. For instance, a *B&A*-query Q may be defined as a pair (P_{before}, P_{after}) of graph patterns. The result of Q when applied to a graph history is a set of vertices such that their induced subgraph matches P_{before} in some time instance t_i and P_{after} in some time instance t_j with $t_j > t_i$. For example, a special case is looking for three connected vertices that latter form a triangle. *B&A*-queries can be generalized, so that, instead of just two graph patterns, they take as input an ordered set of $m \geq 2$ graph patterns, and look for matches of all, or some, of the m patterns in a corresponding sequence of graph snapshots.

Through a *change query*, we would like to identify elements of the graph based on the amount of change they have experience during the evolution of the graph. For instance, we may want to identify volatile graph elements, that is, graph elements with a high-rate of updates, and stable graph elements, that is graph elements with a low-rate of updates.

Graph journey queries will concentrate on *ordered set* of graph elements, where a graph element with order i must appear earlier in time than a graph element with order $j > i$. Journeys are interesting since they possibly reveal dependencies, or causalities. Take for example a graph that reports information propagation, e.g., a graph whose vertices represent users and there is a directed edge from user u to user v at graph snapshot G_t, if u has sent a piece of information to v at time instant t. Such graphs are also known as *interaction graphs*. As an example, take a *diffusion* query taking as input an ordered set of labels $(L_1, L_2, \ldots L_m)$ and asking for a set of m vertices $(u_1, u_2, \ldots u_m)$ such that each u_i has label L_i, for $1 \leq i \leq m$, and there is an edge (u_{j-1}, u_j) that appears before an edge (u_j, u_{j+1}), for $1 < j < m$. The output of such queries is for instance useful for modeling how information flows in a network.

We will study both *strict* and *partial* order. This will allow us to consider both "ties" in the network, that is events happening at the same time, as well as, indifference, that is, events whose order is not relevant for the problem at hand. Furthermore, we will look into cases where there is delay associated with the edges.

In defining the formal model of our graph evolution and graph journey queries, additional research issues include:

- aggregation: users should be able to interactively change (a) the time granularity (e.g., by aggregating snapshots), and (b) the structural granularity (e.g., by restricting queries to sub-graphs) and (c) content granularity (e.g., by expressing constraints on the values associated with vertices and edges), of their queries.

- query relaxation by providing different graph pattern matching semantics (e.g., using isomorphism, or simulation [10, 2]).

- continuous query semantics when the graph updates arrive as a stream.

To efficiently process graph evolution and graph journey queries, appropriate index structures and algorithms will be introduced. Another novel aspect of our approach will be the exploration of graph embeddings for building appropriate graph indexes.

Graph embeddings aim at representing large-scale graphs by mapping their vertices to low-dimensional spaces, e.g., by constructing and embedding the affinity graph into a low dimensional space, or by applying matrix factorization to find the low-dimensional embedding (see, for example [7] for a recent survey). Graph embeddings have been widely used in a variety of graph mining tasks including link prediction, node classification and community detection. However, their use as indexes for query processing is very limited.

One novelty of our approach is that we plan to explore graph embedding for indexing. There has been some previous work to this end, but this is very limited. For example, the authors of [69] use a technique based on selecting landmark vertices, and computing distances from them, while the authors of [68] encode the labels of vertices within distance r of a vertice into a signature for this vertice. Furthermore, most existing work in graph embeddings considers static graphs. In *GraphTempo* project, we will build on very recent ongoing work on embeddings for temporal graphs, such as [71] that uses the concept of a neighborhood formation sequence to describe the evolution of a vertex, where temporal excitation effects exist between neighbors in the sequence.

We also expect various novel research results from the application of the *GraphTempo* to real-world networks. These applications will be studied along the development of the *GraphTempo* framework and implemented in real-world networks. In addition, these case studies will be used to validate the effectiveness of our queries.

Case studies of the graph evolution queries to be explored include:

(1) *team formation* in network by identifying stable cooperations, that is, sets of vertices (or, graph elements in general) with a small amount of change,

(2) spotting *homophily* in social and cooperation networks, for example, by locating co-evolving content and structural transformations, and

(3) *anomaly detection* (such as spam behavior) in social networks, for example, by finding unexpected transformations, or volumes of change.

Case studies of the graph journey queries to be explored include:

(1) identifying *information cascades* in social networks, for example, by finding the most common journeys that a piece of information (e.g., a hashtag) follows, and characterizing *forged information propagation* (e.g., fake news), for instance by looking at the journeys followed through time.

(2) locating *spurious transaction* in financial networks such as in bitcoin, where the order between the vertices participating in the interactions is important, and

(3) finding *interesting routes* in transportation networks, for example, by looking at places of interest commonly visited in a sequence.

Numerous systems have been proposed for managing graphs (see, for example, [18, 67] for recent surveys). As part of the project, we shall comparatively evaluate alternative graph processing systems. The ***GraphTempo system*** will implement the proposed framework as an extension of the system that is found most appropriate after the evaluation. We shall also consider extending the Gremlin graph traversal language, part of the TinkerPop graph processing framework [62], towards supporting graph evolution and graph journey queries. Note that Gremlin is identified as the prevalent graph query processing in a recent survey [51].

During the development of *GraphTempo system*, we expect a number of novel research contributions to be made centered around two axes:

- the integration of our formal model of graph evolution and graph journey queries with existing graph query languages, and
- system-related issues based on the design choices that will be made with respect to implementation aspects, for example, regarding the storage layout, index structures, and the potential use of parallelism.

Finally, *GraphTempo* will introduce a new ***qeury-evolution based link recommendation*** algorithm. Link recommendations are a very well-studied research topic (e.g., see [34] for a survey). Link recommendations are critical for both improving the utility and expediting the growth of networks. Most previous approaches focus on suggesting links based on expected accuracy by recommending links that are highly likely to be adopted. Recent research uses link recommendations as a tool to control the evolution of a network towards achieving desirable properties. Along this line, in our previous research, we have considered link recommendations from the network perspective by aiming at recommending links that if adopted by the users will improve properties of the network such as the network centrality [45].

In the *GraphTempo* project, we shall consider enhancing this form of link recommendation algorithms by using historical information. The goal is to increase the likelihood of the user accepting the recommended links and thus the likelihood of having networks with improved properties. Besides improving accuracy, we will also consider recommending diverse links to counteract homogeneity.

Yet another novelty of our approach is that we plan to explore embeddings not only towards recommending single links but also on making predictions about queries. This is a complete new area of research with many open problems. Some initial work to this end is the very recent work of the authors in [12] where the focus is on making predictions about conjunctive logical queries.

1.3.2 Overall scientific methodology. To pursue our goals and targeted objectives, the proposed research work is divided into workpackages (as detailed in Section 3), each one corresponding to our specific research objectives (as defined in Section 1.1). Dividing the overall proposed research work in distinguishable parts, allows us to monitor progress, set milestones and take corrective actions if necessary.

The development of new theory in each workpackage builds upon a systematic analysis of the *state-of-art* from different areas to identify key problems, models and techniques. These will form the basis to advance our *model* and *algorithms*. *Validation* of our models and algorithms will proceed incrementally, using both synthetic and real datasets, thus creating a feedback loop involving theory and algorithm design, tool development and experimental validation.

Real data will drive the development of theory, algorithms and systems, which in turn will enable us to tackle new applied problems in practice. Specifically, for the evaluation of both the efficiency and the effectiveness of our models and algorithms, data from large collections of real graph datasets, such as from SNAP [59] and konect [26] will be used. Real data will also be collected using the API provided by many social networks. Furthermore, for evaluating the efficiency of the framework, in addition to real datasets, synthetic graph sequences will be generated using appropriate graph evolution models (e.g., [33]).

Developed software will be open-source and freely available in order to involve a larger research community. The same holds for any data collected for the purposes of the *GraphTempo* project.

In more detail, the first two workpackages (WP1 and WP2) are devoted to the foundations of our declarative framework which consists of a set of novel queries. Queries are divided into evolution (WP1) and journey queries (WP2), where journey queries focus on time-order. The scientific methodology detailed above will be followed, that is, conducting a state-of-the-art survey followed by theory development and validation in a feedback loop. We note especially that in terms of validation, besides efficiency, in these two workpackages, particular emphasis will be placed in effectiveness. In particular, the effectiveness of our approach will be demonstrated through specific case studies involving important problems (such as team formation, homophily, information cascades, etc) in real-world networks (e.g., social networks, bitcoin).

The third workpackage (WP3) considers the integration of our framework in a graph management system, again following our overall scientific methodology. We note that in this workpackage, the state-of-the art phase will also involve a comparative evaluation of existing graph management systems (including native graph databases and relational ones). The fourth workpackage (WP4) considers the development of a link recommendation algorithm that uses knowledge extracted from the graph evolution.

2. Impact

2.1. Scientific, economic, and/or social impact

Graphs are ubiquitous as a modeling abstraction as they ofter a very natural model for representing connected entities and the interactions and relationships between them. They are prevalent in applications such in social networks, the web, the social web, in transportation and communication networks, in biology and finance, to name just a few. Graph data management has been the focus of much current research and there is also a surge on related commercial and research software.

Most graphs used to represent real-world entities and their connections are not static but evolve with time. As reported in [51], one of the future challenges is graph data management is being able to maintain and query this history. There is rich information in the history of graphs. In general, exploring change is envisioned to advance data science in many ways in the future [5]. *GraphTempo* addresses this exact need and vision by proposing a declarative framework for the interactive exploration of the evolution of graphs through time. Being both relevant and timely, we expect the research contributions of *GraphTempo* to have important *scientific impact*.

GraphTempo has also the potential to create *social* and *economic* impact. Graphs model important real-world networks. Exploring the evolution of such networks through time will improve our understanding of the mechanisms underlying their evolution. Querying the history of a graph may reveal hidden properties, alert us in cases of unexpected changes and help us identify suspicious behavior.

To this end, in *GraphTempo*, we will apply our framework to a number of dynamic real-world networks towards capturing, understanding and addressing specific social and economic problems. In particular, we will study homophily, the tendency of individuals to connect with others similar to them, or being influenced by their connections. Homoplily often leads to information bubbles and echo chambers. We will also look into information cascades in networks to address important social problems such us the propagation of fake news. In addition, we will look into online financial transactions to identify spurious interactions.

Taking a step further, *GraphTempo* will look into predicting and potentially influencing the evolution of real-world networks. This will be achieved by link recommendation algorithms that propose links such that their addition to the network will improve desired properties of the network. As an example, consider a social network. By recommending to users appropriate links to follow, we can affect the evolution of the network and improve the quality of its influence on its member. For example, we can recommend links to diverse users to counteract homophily, or, discourage the propagation of abusive behavior. By providing the mechanisms to improve real world networks, *GraphTempo* has the potential of making an important societal and economic impact.

As explained, many of the proposed methods can serve as novel approaches to addressing societal challenges. Furthermore, the proposed framework and its result can be *easily adopted and integrated* by related national bodies and high growth companies. Additional examples include forming appropriate teams for solving problems and detecting anomalous behavior, such as spam communication, as showcased in the related case studies.

Furthermore, *GraphTempo* promotes the scientific and technological development of the country in a high tech field such as graph data management. The proposed research topics have also a high degree of *innovation*. It is expected to contribute to the *competitiveness* of research both in national and international level.

Developed software will be open-source and freely available as well as any data collected to maximize among others the *exploitation and dissemination* of the proposed research. Moreover, the research results of the project will be published and presented in highly competitive international conferences and journals.

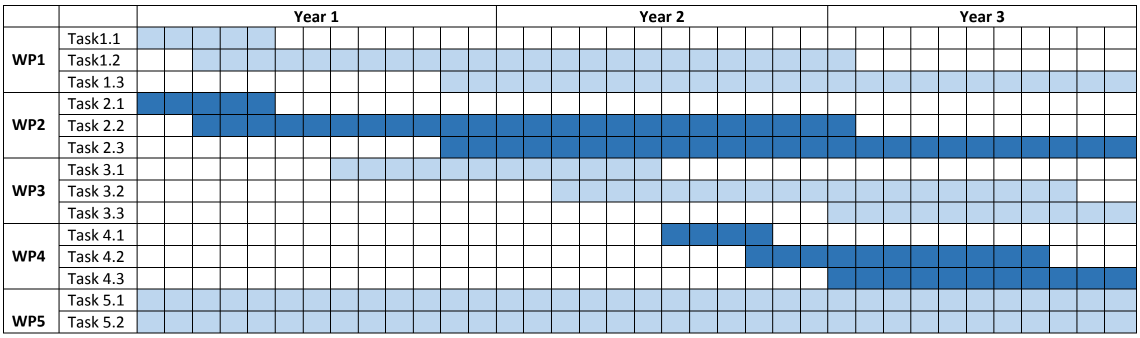
Finally, since the topic of the project involves real applications of interest to the general public (such as social and financial networks), we intend to target a wider coverage of the project. For example, we intend to present selected results in social media and the press.

3. Implementation

3.1. Work plan – Deliverables – Milestones

The proposed research work is divided into four technical workpackages (WP1-WP4), each one of them corresponding to our specific research objectives (as defined in Section 1.1). Dividing the overall proposed research work in distinguishable parts, allows us to define specific tasks per workpackage, monitor progress, set milestones and take corrective actions if necessary. Work in each technical workpackage starts with a systematic analysis of the *state-of-art* followed by the development of *models* and *algorithms* and their *validation* which proceeds incrementally with the theoretical development, thus creating a feedback loop. There is one additional workpackage (WP5) devoted to coordination and dissemination.

Below is the time shedule of the workpackages (Gantt chart).



The first two workpackages (WP1 and WP2) will focus on the development of the declarative framework for querying the evolution of graphs. The first workpackage (WP1) is devoted to graph evolution queries.

<table>
<tr><td colspan="2">WP Number: 1</td><td>WP Title: Graph Evolution Queries</td></tr>
<tr><td>Starting Month: 1</td><td>Ending Month: 36</td><td>Person Months: 38</td></tr>
<tr><td colspan="3">Objectives: Introduce novel graph evolution queries, query processing algorithms, index structures and real-world applications
Description of Work: Survey related work, formally define transformation and change queries, propose appropriate indexes and query processing algorithms using dynamic graph embeddings, evaluate efficiency using real and synthetic datasets, design and implement use cases to evaluate effectiveness
Tasks: Task 1.1: State-of-the-art survey on evolving graph management, Task 1.2: Formal models and algorithms, Task 1.3: Evaluation and Use cases (team formation, homophily, anomaly detection)
Deliverables: D1.1, D1.2, D1.3
Milestones: M1, M5.1, M5.2</td></tr>
</table>

The second workpackage (WP2) is devoted to graph journey queries.

WP Number: 2		**WP Title: Graph Journey Queries**
Starting Month: 1	**Ending Month: 36**	**Person Months:** 38
Objectives: Introduce novel graph journey queries, query processing algorithms, index structures and real-world applications **Description of Work:** Survey related work, formally define journey queries with both strict and partial order, propose appropriate indexes and query processing algorithms using dynamic graph embeddings, evaluate efficiency using real and synthetic datasets, design and implement use cases to evaluate effectiveness **Tasks: Task 2.1:** State-of-the-art survey on interaction graphs and diffusion, **Task 2.2:** Formal models and algorithms, **Task 2.3:** Evaluation and Use cases (financial transactions, information cascades, routes) **Deliverables:** D2.1, D2.2, D2.3 **Milestones:** M2, M5.1, M5.2		

The third workpackage (WP3) concerns the integration of our framework in a graph data management system.

WP Number: 3		**WP Title: The *GraphTempo* System**
Starting Month: 8	**Ending Month: 36**	Person Months: 32
Objectives: Extend the functionality of a graph processing system with the *GraphTempo* queries **Description of Work:** Comparatively evaluate different graph processing systems, implement the proposed queries, evaluate efficiency and effectiveness using real and synthetic datasets **Tasks: Task 3.1:** Comparative evaluation of graph processing systems, **Task 3.2:** Implementation of the proposed queries in a graph processing system, **Task 3.3:** System evaluation **Deliverables:** D3.1, D3.2, D3.3 **Milestones:** M3, M5.2		

The fourth workpackage (WP4) refers to the link recommendation algorithm.

WP Number: 4		**WP Title: Graph-history Driven Recommendations**
Starting Month: 18	**Ending Month: 36**	**Person Months:** 12
Objectives: Explore the novel queries to propose a network-aware link recommendation algorithm **Description of Work:** Survey related work, design a link recommendation algorithm, evaluate efficiency and effectiveness using real and synthetic datasets **Tasks: Task 4.1:** State-of-the-art survey on link recommendations, **Task 4.2:** Design of the link recommendation algorithm, **Task 4.3:** Evaluation of efficiency and effectiveness **Deliverables:** D4.1, D4.2 **Milestones:** M4, M5.2		

The last workpackage (WP5) is devoted to coordination and dissemination.

WP Number: 5		**WP Title: Coordination and Dissemination**
Starting Month: 1	**Ending Month: 36**	**Person Months:** 6
Objectives: Coordinate the project and disseminate its results. **Description of Work:** Coordinate the project, schedule meetings, oversee the timely delivery of reports, quality control, create and maintain the project web page and social media account project result presentation **Tasks: Task 5.1:** Project coordination, **Task 5.2:** Result dissemination **Deliverables:** D5.1, D5.2, D5.3 **Milestones:** M5.1, M5.2		

Below is a list of the project deliverables.

Deliverable Number	Deliverable Name	Related WP	Dissemination Level	Delivery Date
D1.1	Scientific publication (conference) on transformation queries	WP1	Public	M15
D1.2	Scientific publication (conference) on change queries	WP1	Public	M30
D1.3	Scientific publication (journal) on evolution queries	WP1	Public	M36
D2.1	Scientific publication (conference) on journey queries (total order)	WP2	Public	M15
D2.2	Scientific publication (conference) on journey queries (partial order)	WP2	Public	M30
D2.3	Scientific publication (journal) on journey queries	WP2	Public	M36
D3.1	Scientific publication (demo) of the *GraphTempo* system	WP3	Public	M26
D3.2	Scientific publication (journal) on the *GraphTempo* system	WP3	Public	M36
D3.3	Open source software of the *GraphTempo* system	WP3	Public	M36
D4.1	Scientific publication (conference) on the link recommendation algorithm	WP4	Public	M30
D4.2	Scientific publication (journal) on the link recommendation algorithm	WP3	Public	M36
D5.1	Project web page	WP5	Public	M3 (+updates)
D5.2	Project social media (twitter)	WP5	Public	M3 (+updates)
D5.3	Article in popular press	WP5	Public	M26

GraphTempo is a research project, ***milestones*** are set one per technical workpackage to evaluate progress. Specifically, for WP1 and WP2, the milestones M1 and M2 respectively refer to attaining the first experimental results regarding the efficiency and effectiveness of the proposed models and algorithms for the evolution and journey queries respectively. For WP3, milestone M3 is the first release of the *GraphTempo* system implementing a specific subset of the proposed queries to be determined during the course of the project. For WP4, milestone M4 refers to attaining the first experimental results regarding the efficiency and effectiveness of the link recommendation algorithm.

There are also two general milestones set on Month 12 and Month 24 (milestones M5.1 and M5.2) where all members of the research team will meet, evaluate the research results so far and take overall corrective actions if necessary.

The milestones are detailed in the following table.

Milestone Number	Milestone Name	Related WP	Delivery Date	Means of Verification
M1	Evolution Queries	WP1	M16	Experimental validation of efficiency and effectiveness
M2	Journey Queries	WP2	M16	Experimental validation of efficiency and effectiveness
M3	*GraphTempo* system	WP3	M24	First version of the *GraphTempo* system implementing a specific subset of the queries
M4	Link recommendation	WP4	M30	Experimental validation of efficiency and effectiveness
M5.1	1st year progress	WP5	M13	Presentation of 1st year project results
M5.2	2nd year progress	WP5	M25	Presentation of 1st year project results

Finally, below we present a number of potential risks and related mitigation measures.

Description of Risk	WPs Involved	Proposed Risk-mitigation Measures
Validation through case studies shows proposed model of evolution queries to be ineffective (medium level)	WP1	Redesign the models based on the feedback from the case studies
Validation through case studies shows proposed model of journey queries to be ineffective (medium level)	WP2	Redesign the models based on the feedback from the case studies
Chosen graph data management system proves not appropriate (medium level)	WP3	Choose alternative system
Data collection problem (medium level)	WP1, WP2 WP3, WP4	Use real data from repositories, use synthetic data, consider alternative systems for data collection
Problems in student recruitment (medium level)	WP1, WP2, WP3, WP4	Upon acceptance, the PI will make a national call for students with the maximum possible publicity, also potential candidates among undegraduate students working under the supervision of PI and Prof. Tsaparas

4. Research team and the cooperating/collaborating organizations selection rationale

The research team at the University of Ioannina consists of two faculty members, namely *Evaggelia Pitoura* (principal investigator) and *Panayiotis Tsaparas*, two PhD students and 4 graduate students (to be hired).

Professor *Evaggelia Pitoura* has more than twenty years of experience in data management research. She is currently pursuing research on graph data management and topics directly related to the proposal [56, 52, 54, 53, 24, 25, 27]. She will work on all the workpackages of the project. She has extensive experience in supervising students. She has graduated 4 PhD students, served in the committee of more than 20 PhD students (in Greece and abroad), graduated more than 20 MSc students and more than 40 BSc students. She has successfully coordinated several national and international projects.

Associate Professor *Panayiotis Tsaparas* is well-known for his research in graph mining and social network analysis with prominent contributions in web link analysis and in clustering. Prof. Tsaparas and Prof. Pitoura have been actively collaborating on various issues relating to social networks [11, 63, 44, 32] and recommendations [58, 45, 47]. In the *GraphTempo* project, they will continue and extend their cooperation. Prof. Tsaparas will contribute in particular in link recommendations (WP4) and in designing and evaluating the case studies of WP1 towards extracting important knowledge about the social network dynamics.

The two *PhD students* will work on the two types of queries: one student will work on graph evolution queries (WP1) and the other one on graph journey queries (WP2). They will cooperate in building the *GraphTempo* system (WP3) and the link recommendation algorithm (WP4).

The *graduate students* will work as part of their theses on topics related to the proposal. In particular, one graduate student will work on the case studies in WP1 and another one on the case studies in WP2. Two graduate students will work in the implementation of the *GraphTempo* system in WP3.

There are two additional members in the team from two collaborating organizations, namely *Peter Triantafillou* from the University of Warwick, UK and *Evimaria Terzi* from Boston University, USA.

Professor *Peter Triantafillou* is a world expert on big data systems and large-scale data infrastructures. His research has a strong experimental flavor, implementing and testing the developed (sub)systems. He has contributed a number of vision papers; most recently he has been invited to author and present a vision paper at ICDCS 2018 and a lightning talk at ICDE 2018. He is a co-designer of several innovative systems, such as Minerva [3], one of the first decentralized search engines and eXO [36], a decentralized social networking system. He is currently working on graph processing systems and their performance benchmarking (e.g., [19, 64]). He has co-operated in several projects with Evaggelia Pitoura (FET projects DBGlobe, Aeolos, Thalis project, Cloud9). He will be actively involved in the design of the *GraphTempo* system (WP3).

Associate Professor *Evimaria Terzi* is an authority on algorithmic data mining with influential work on the formulation of various graph related concepts (e.g., social teams [30], effectors [31], graph anonymization [35]). She has already started a collaboration with Evaggelia Pitoura and Panayiotis Tsaparas on data mining in historical graphs [57]. In the *GraphTempo*, she will work in particular in the theoretical formulation of journey-based queries and in designing the related case studies contributing to the social network analysis (WP2).

References

[1] Takuya Akiba, Yoichi Iwata, and Yuichi Yoshida. Dynamic and historical shortest-path distance queries on large evolving networks by pruned landmark labeling. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 237–248, 2014.

[2] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan L. Reutter, and Domagoj Vrgoc. Foundations of modern query languages for graph databases. *ACM Comput. Surv.*, 50(5):68:1–68:40, 2017.

[3] Matthias Bender, Sebastian Michel, Peter Triantafillou, Gerhard Weikum, and Christian Zimmer. MINERVA: collaborative P2P search. In *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005*, pages 1263–1266, 2005.

[4] Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. Mining graph evolution rules. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part I*, pages 115–130, 2009.

[5] Tobias Bleifuß, Leon Bornemann, Theodore Johnson, Dmitri V. Kalashnikov, Felix Naumann, and Divesh Srivastava. Exploring change - A new dimension of data analytics. *PVLDB*, 12(2):85–98, 2018.

[6] Petko Bogdanov, Misael Mongiovì, and Ambuj K. Singh. Mining heavy subgraphs in time-evolving networks. In *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*, pages 81–90, 2011.

[7] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.*, 30(9):1616–1637, 2018.

[8] Ciro Cattuto, Marco Quaggiotto, André Panisson, and Alex Averbuch. Time-varying social networks in a graph database: a neo4j use case. In *First International Workshop on Graph Data Management Experiences and Systems, GRADES 2013, co-loated with SIGMOD/PODS 2013, New York, NY, USA, June 24, 2013*, page 11, 2013.

[9] Gabriel Campero Durand, Marcus Pinnecke, David Broneske, and Gunter Saake. Backlogs and interval timestamps: Building blocks for supporting temporal queries in graph databases. In *Proceedings of the Workshops of the EDBT/ICDT 2017 Joint Conference (EDBT/ICDT 2017), Venice, Italy, March 21-24, 2017.*, 2017.

[10] Wenfei Fan, Jianzhong Li, Shuai Ma, Nan Tang, Yinghui Wu, and Yunpeng Wu. Graph pattern matching: From intractable to polynomial time. *PVLDB*, 3(1):264–275, 2010.

[11] Nathalie T. H. Gayraud, Evaggelia Pitoura, and Panayiotis Tsaparas. Diffusion maximization in evolving social networks. In *Proceedings of the 2015 ACM on Conference on Online Social Networks, COSN 2015, Palo Alto, California, USA, November 2-3, 2015*, pages 125–135, 2015.

[12] William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. Embedding logical queries on knowledge graphs. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 2030–2041, 2018.

[13] Wentao Han, Youshan Miao, Kaiwei Li, Ming Wu, Fan Yang, Lidong Zhou, Vijayan Prabhakaran, Wenguang Chen, and Enhong Chen. Chronos: a graph engine for temporal graph analysis. In *Ninth Eurosys Conference 2014, EuroSys 2014, Amsterdam, The Netherlands, April 13-16, 2014*, pages 1:1–1:14, 2014.

[14] Takanori Hayashi, Takuya Akiba, and Ken-ichi Kawarabayashi. Fully dynamic shortest-path distance query acceleration on massive networks. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 1533–1542, 2016.

[15] Wenyu Huo and Vassilis J. Tsotras. Efficient temporal shortest path queries on evolving social graphs. In *Conference on Scientific and Statistical Database Management, SSDBM '14, Aalborg, Denmark, June 30 - July 02, 2014*, pages 38:1–38:4, 2014.

[16] Christian S. Jensen and Richard T. Snodgrass. Temporal data management. *IEEE Trans. Knowl. Data Eng.*, 11(1):36–44, 1999.

[17] Xiaoen Ju, Dan Williams, Hani Jamjoom, and Kang G. Shin. Version traveler: Fast and memory-efficient version switching in graph processing systems. In *2016 USENIX Annual Technical Conference, USENIX ATC 2016, Denver, CO, USA, June 22-24, 2016.*, pages 523–536, 2016.

[18] Martin Junghanns, André Petermann, Martin Neumann, and Erhard Rahm. Management and analysis of big graph data: Current systems and open challenges. In *Handbook of Big Data Technologies*, pages 457–505. 2017.

[19] Foteini Katsarou, Nikos Ntarmos, and Peter Triantafillou. Performance and scalability of indexed subgraph query processing methods. *PVLDB*, 8(12):1566–1577, 2015.

[20] David Kempe, Jon M. Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *J. Comput. Syst. Sci.*, 64(4):820–842, 2002.

[21] Udayan Khurana and Amol Deshpande. Efficient snapshot retrieval over historical graph data. In *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, pages 997–1008, 2013.

[22] Udayan Khurana and Amol Deshpande. Storing and analyzing historical graph data at scale. In *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016.*, pages 65–76, 2016.

[23] Min-Soo Kim and Jiawei Han. A particle-and-density based evolutionary clustering method for dynamic networks. *PVLDB*, 2(1):622–633, 2009.

[24] Georgia Koloniari and Evaggelia Pitoura. Partial view selection for evolving social graphs. In *First International Workshop on Graph Data Management Experiences and Systems, GRADES 2013, co-loated with SIGMOD/PODS 2013, New York, NY, USA, June 24, 2013*, page 9, 2013.

[25] Georgia Koloniari, Dimitris Souravlias, and Evaggelia Pitoura. On graph deltas for historical queries. *CoRR*, abs/1302.5549, 2013. Proceedings of 1st Workshop on Online Social Systems (WOSS) 2012, in conjunction with VLDB 2012.

[26] Konect. *The Koblenz Network Collection*, 2018 (accessed January). `http://konect.uni-koblenz.de/`.

[27] Andreas Kosmatopoulos, Kostas Tsichlas, Anastasios Gounaris, Spyros Sioutas, and Evaggelia Pitoura. Hinode: an asymptotically space-optimal storage model for historical queries on graphs. *Distributed and Parallel Databases*, 35(3-4):249–285, 2017.

[28] C. Kosyfaki, N. Mamoulis, E. Pitoura, and P. Tsaparas. Flow motifs in interaction networks. In *EDBT*, page To appear, 2019.

[29] Alan G. Labouseur, Jeremy Birnbaum, Paul W. Olsen, Sean R. Spillane, Jayadevan Vijayan, Jeong-Hyon Hwang, and Wook-Shin Han. The g* graph database: efficiently managing large distributed dynamic graphs. *Distributed and Parallel Databases*, 33(4):479–514, 2015.

[30] Theodoros Lappas, Kun Liu, and Evimaria Terzi. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 467–476, 2009.

[31] Theodoros Lappas, Evimaria Terzi, Dimitrios Gunopulos, and Heikki Mannila. Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 1059–1068, 2010.

[32] Konstantina Lazaridou, Konstantinos Semertzidis, Evaggelia Pitoura, and Panayiotis Tsaparas. Identifying converging pairs of nodes on a budget. In *Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015, Brussels, Belgium, March 23-27, 2015.*, pages 313–324, 2015.

[33] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *TKDD*, 1(1):2, 2007.

[34] David Liben-Nowell and Jon M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[35] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 93–106, 2008.

[36] Andreas Loupasakis, Nikos Ntarmos, and Peter Triantafillou. exo: Decentralized autonomous scalable social networking. In *CIDR 2011, Fifth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 9-12, 2011, Online Proceedings*, pages 85–95, 2011.

[37] Peter Macko, Virendra J. Marathe, Daniel W. Margo, and Margo I. Seltzer. LLAMA: efficient graph analytics using large multiversioned arrays. In *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, pages 363–374, 2015.

[38] Youshan Miao, Wentao Han, Kaiwei Li, Ming Wu, Fan Yang, Lidong Zhou, Vijayan Prabhakaran, Enhong Chen, and Wenguang Chen. Immortalgraph: A system for storage and analysis of temporal graphs. *TOS*, 11(3):14:1–14:34, 2015.

[39] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2004.

[40] Vera Zaychik Moffitt and Julia Stoyanovich. Temporal graph algebra. In *Proceedings of The 16th International Symposium on Database Programming Languages, DBPL 2017, Munich, Germany, September 1, 2017*, pages 10:1–10:12, 2017.

[41] Vera Zaychik Moffitt and Julia Stoyanovich. Towards sequenced semantics for evolving graphs. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017.*, pages 446–449, 2017.

[42] Mohammad Hossein Namaki, Yinghui Wu, Qi Song, Peng Lin, and Tingjian Ge. Discovering graph temporal association rules. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 1697–1706, 2017.

[43] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. In *WSDM*, pages 601–610, 2017.

[44] Nikos Parotsidis, Evaggelia Pitoura, and Panayiotis Tsaparas. Selecting shortcuts for a smaller world. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 28–36, 2015.

[45] Nikos Parotsidis, Evaggelia Pitoura, and Panayiotis Tsaparas. Centrality-aware link recommendations. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, pages 503–512, 2016.

[46] Evaggelia Pitoura. Historical graphs: Models, storage, processing. Springer.

[47] Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. Recommending packages to groups. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, pages 449–458, 2016.

[48] Chenghui Ren, Eric Lo, Ben Kao, Xinjie Zhu, and Reynold Cheng. On querying historical evolving graph sequences. *PVLDB*, 4(11):726–737, 2011.

[49] Chenghui Ren, Eric Lo, Ben Kao, Xinjie Zhu, Reynold Cheng, and David W. Cheung. Efficient processing of shortest path queries in evolving graph sequences. *Inf. Syst.*, 70:18–31, 2017.

[50] Polina Rozenshtein and Aristides Gionis. Temporal pagerank. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II*, pages 674–689, 2016.

[51] Siddhartha Sahu, Amine Mhedhbi, Semih Salihoglu, Jimmy Lin, and M. Tamer Özsu. The ubiquity of large graphs and surprising challenges of graph processing. *PVLDB*, 11(4):420–431, 2017.

[52] Konstantinos Semertzidis and Evaggelia Pitoura. Durable graph pattern queries on historical graphs. In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pages 541–552, 2016.

[53] Konstantinos Semertzidis and Evaggelia Pitoura. Time traveling in graphs using a graph database. In *Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference, EDBT/ICDT Workshops 2016, Bordeaux, France, March 15, 2016.*, 2016.

[54] Konstantinos Semertzidis and Evaggelia Pitoura. Historical traversals in native graph databases. In *Advances in Databases and Information Systems - 21st European Conference, ADBIS 2017, Nicosia, Cyprus, September 24-27, 2017, Proceedings*, pages 167–181, 2017.

[55] Konstantinos Semertzidis and Evaggelia Pitoura. Top-k durable graph pattern queries on temporal graphs. *IEEE Trans. Knowl. Data Eng.*, 31(1):181–194, 2019.

[56] Konstantinos Semertzidis, Evaggelia Pitoura, and Kostas Lillis. Timereach: Historical reachability queries on evolving graphs. In *Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015, Brussels, Belgium, March 23-27, 2015.*, pages 121–132, 2015.

[57] Konstantinos Semertzidis, Evaggelia Pitoura, Evimaria Terzi, and Panayiotis Tsaparas. Best friends forever (BFF): finding lasting dense subgraphs. *Data Mining and Knowledge Discovery*, (Journal track of PKDD 2019), 2019.

[58] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. Fairness in package-to-group recommendations. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 371–379, 2017.

[59] SNAP. *Large Network Dataset Collection*, 2018 (accessed January). `https://snap.stanford.edu/data/`.

[60] Chayant Tantipathananandh, Tanya Y. Berger-Wolf, and David Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, pages 717–726, 2007.

[61] Manuel Then, Timo Kersten, Stephan Günnemann, Alfons Kemper, and Thomas Neumann. Automatic algorithm transformation for efficient multi-snapshot analytics on temporal graphs. *PVLDB*, 10(8):877–888, 2017.

[62] Apache TinkerPop. 2018 (accessed January). `http://tinkerpop.apache.org/`.

[63] Paraskevas Tsantarliotis, Evaggelia Pitoura, and Panayiotis Tsaparas. Defining and predicting troll vulnerability in online social media. *Social Netw. Analys. Mining*, 7(1):26:1–26:15, 2017.

[64] Jing Wang, Nikos Ntarmos, and Peter Triantafillou. Graphcache: A caching system for graph queries. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017.*, pages 13–24, 2017.

[65] Huanhuan Wu, James Cheng, Silu Huang, Yiping Ke, Yi Lu, and Yanyan Xu. Path problems in temporal graphs. *PVLDB*, 7(9):721–732, 2014.

[66] Huanhuan Wu, Yuzhen Huang, James Cheng, Jinfeng Li, and Yiping Ke. Reachability and time-based path queries in temporal graphs. In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pages 145–156, 2016.

[67] Da Yan, Yingyi Bu, Yuanyuan Tian, and Amol Deshpande. Big graph analytics platforms. *Foundations and Trends in Databases*, 7(1-2):1–195, 2017.

[68] Peixiang Zhao and Jiawei Han. On graph query optimization in large networks. *PVLDB*, 3(1):340–351, 2010.

[69] Lei Zou, Lei Chen, M. Tamer Özsu, and Dongyan Zhao. Answering pattern match queries in large graph databases via graph embedding. *VLDB J.*, 21(1):97–120, 2012.

[70] Andreas Züfle, Matthias Renz, Tobias Emrich, and Maximilian Franzke. Pattern search in temporal social networks. In *EDBT*, pages 289–300, 2018.

[71] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. Embedding temporal network via neighborhood formation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 2857–2866, 2018.