IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 2, 2020, accepted August 15, 2020, date of publication August 21, 2020, date of current version September 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3018497

A Framework for Subgraph Detection in Interdependent Networks via Graph Block-Structured Optimization

FEI JIE[1,2,5], CHUNPAI WANG[3], FENG CHEN[4], (Member, IEEE), LEI LI[1,2], (Senior Member, IEEE), AND XINDONG WU[1,2,5], (Fellow, IEEE)

[1]Key Laboratory of Knowledge Engineering with Big Data (Ministry of Education), Hefei University of Technology, Hefei 230601, China
[2]School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China
[3]Department of Computer Science, University at Albany – SUNY, Albany, NY 12222, USA
[4]Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas, Richardson, TX 75080, USA
[5]Mininglamp Academy of Sciences, Mininglamp Technology, Beijing 100084, China

Corresponding author: Xindong Wu (xwu@hfut.edu.cn)

The work of Fei Jie was supported in part by the scholarship from China Scholarship Council (CSC). This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000901, in part by the National Natural Science Foundation of China under Grant 91746209, in part by the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education of China under Grant IRT17R32, and in part by the US National Science Foundation under Grant IIS-1815696 and Grant IIS-1750911.

ABSTRACT As the backbone of many real-world complex systems, networks interact with others in nontrivial ways from time to time. It is a challenging problem to detect subgraphs that have dependencies on each other across multiple networks. Instead of devising a method for a specific scenario, we propose a generic framework to discover subgraphs in multiple interdependent networks, which generalizes the classical subgraph detection problem in a single network and can be applied to more practical applications. Specifically, we propose the **G**raph **B**lock-structured **G**radient **H**ard **T**hresholding **P**ursuit (**GB-GHTP**) framework to optimize interdependent networks with block-structured constraints, which enjoys 1) a theoretical guarantee and 2) a nearly linear time complexity on the network size. It is demonstrated how our framework can be applied to three practical applications: 1) evolving anomalous subgraph detection in dynamic networks, 2) anomalous subgraph detection in networks of networks, and 3) connected dense subgraph detection in dual networks. We evaluate our framework on large-scale datasets with comprehensive experiments, which validate our framework's effectiveness and efficiency.

INDEX TERMS Subgraph detection, sparse optimization, interdependent networks.

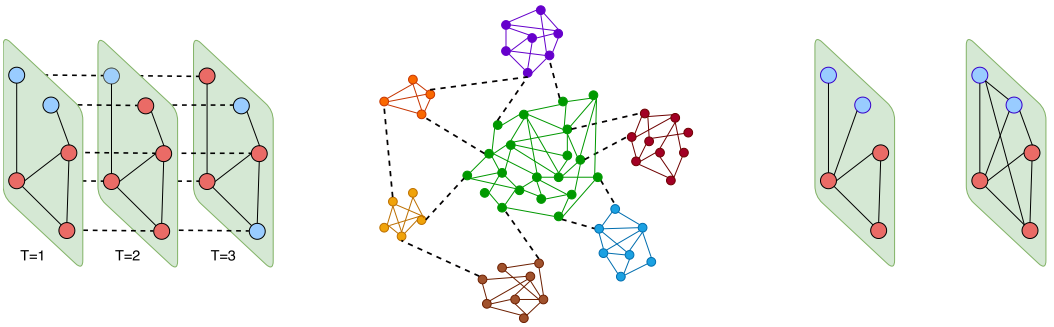
I. INTRODUCTION

A graph[1] $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ refers to a set of entities, denoted as nodes $\mathbb{V}$, along with some connections between node pairs, represented by edges $\mathbb{E}$. Due to its generic structure, graphs have the capability to model various applications, including the natural sciences, social sciences, and engineering [1]–[3]. A canonical challenging problem in graph analytics is the detection of subgraphs. Subgraph detection is useful in many fields, such as intrusion detection in computer networks [4], [5], disease outbreak detection [6], event detection in activity networks [7], [8], and traffic congestion detection [9], [10]. In this paper, we focus on subgraph detection in attributed networks, in which nodes in a graph are associated with attributes. Assume a node i is associated with an attribute vector $w_i \in \mathbb{R}^P$; then the attribute matrix defined on the whole graph can be denoted as $\mathrm{W} \in \mathbb{R}^{P\times N}$, where $|\mathbb{V}| = N$. Subgraph detection in attributed networks usually refers to a problem that finds a subset of nodes whose attributes are anomalous or significant compared to those nodes outside of the subset [11]. The problem simultaneously deals with the structure of a network (e.g. connectivity, density, compactness or isomorphism) and its attributes on nodes. Generally, the typical subgraph detection problem in isolated attributed networks can be formulated as a combinatorial optimization problem as follows:

$$\begin{aligned} &\min_{S\subseteq\mathbb{V}} F(S), \\ &\text{s.t.} \quad S \text{ satisfies some predefined topological constraints} \end{aligned} \tag{1}$$

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif.

[1]In this article, the terms graph and network are used interchangeably.



(a) Dynamic networks (b) Network of networks (c) Dual networks

FIGURE 1. Examples of interdependent networks. (a) Dynamic Networks: black dashed lines describe implicit temporal dependencies or consistencies. (b) Network of networks: black dashed lines are bridges across networks; social networks can be viewed as a network of networks with explicit connections between communities. (c) Dual networks: two networks share the same nodes but have different edge sets that represent different relationships.

where $F(\cdot)$ is a user-specified objective function depending on applications, and S is any subset of $\mathbb{V}$ of a network. The subgraph to be found is determined by the definition of $F(\cdot)$, which has different definitions in different applications. However, in reality, networks rarely appear in isolation. It is common that real-world systems interact with each other. For example, diverse critical infrastructures are coupled together, such as systems of food and water supplies, fuel, communications, financial transactions and power generation and transmission [12]. Specifically, thermal power stations forming the nodes of a power grid require fuel supplied via road or pipe networks and are also controlled by the nodes in a communication network. Although the transportation network does not depend on the power grid to function, the communication network does. Thus cascading failures of a system can originate from the deactivation of a critical number of nodes in either the power grid or the communication network. If the two networks are treated in isolation, this important feedback effect cannot be seen, which further affects the location of malfunctioning nodes. However, jointly detecting significant nodes in both power grid and communication networks can provide deep insight into the functionality dependencies between the two networks.

In the following, we introduce interdependent networks to model multiple networks with dependencies across different networks. Interdependent networks are comprised of multiple networks $\{\mathbb{G}^1, \ldots, \mathbb{G}^k, \ldots\}$ and dependency edges $\mathbb{E}^0$, where $\mathbb{G}^k = (\mathbb{V}^k, \mathbb{E}^k)$ and $\mathbb{E}^0$ is the set of dependencies between networks. The elements in $\mathbb{E}^0$ are determined by a specific application. For example, in the example of power grid, the edge set $\mathbb{E}^0$ refers to connections between thermal power stations and road or pipe network. $\mathbb{V}^k$ and $\mathbb{E}^k$ refer to nodes and edges of the k^{th} network. Multiple networks interacting with each other appear in almost every aspect of science and technology. For instance, a dynamic network can be viewed as multiple networks with implicit node-level temporal dependency, in which each network represents a snapshot of the dynamic network at a specific timestamp. In such networks, every node's attributes in the current timestamp implicitly depend on attributes in the previous timestamp (as shown in Fig. 1a). Another trivial example of interdependent networks is the web-scale social network consisting of many communities. In such networks, communities can be viewed as small networks or blocks that have explicit connections between each other (as shown in Fig. 1b), which technically form a network of networks. A nontrivial example of interdependent networks is the dual networks built from the citation dataset, where one network builds the coauthorships among researchers, and the other network models the research interests between researchers. Both networks have identical node sets (researchers), whereas the edge sets represent the coauthorships and research interest similarities. An example of dual network is shown in Fig. 1c.

Because of the ubiquity of attributed interdependent networks, it is useful to propose generic methods to solve the problem of subgraph detection in interdependent networks. Correspondingly, subgraph detection in multiple interdependent networks can be framed as a block-structured optimization problem with multiple topological constraints on blocks:

$$\min_{S_k \subseteq \mathbb{V}^k} F(S_1, \ldots, S_K) = f(S_1, \ldots, S_K) + \sum_{i \neq j} g(\mathbb{V}^i, \mathbb{V}^j),$$
$$\text{s.t.} \quad S_k \text{ satisfies some predefined topological constraints} \tag{2}$$

where $f(\cdot)$ is a user-specified function to capture signals on nodes of interdependent networks, and $g(\mathbb{V}^i, \mathbb{V}^j)$ models dependencies between network $\mathbb{G}^i$ and network $\mathbb{G}^j$. S_k is a subset of nodes in the k^{th} network $\mathbb{G}^k, k = 1, \ldots, K$. K refers to the number of networks in the interdependent networks. For example, subgraph detection in a dynamic network finds a sequence of subsets of nodes in a sequence of blocks, where the detected subgraphs in each block must satisfy a predefined topological constraint and subgraphs at two consecutive timestamps share some consistency on attributes [4]. K denotes the number of timestamps in this scenario. It is readily concluded that the vanilla subgraph detection problem (1) is a special case of problem (2) when the number of networks (blocks) is 1.

Problem (2) is built on discrete space. Since the solutions for combinatorial optimization with topological constraints are undeveloped, we naturally turn to nonconvex optimization

(optimization techniques for continuous space) for help. To make nonconvex optimization suitable for our scenario, the relaxation of problem (2) from discrete space to continuous space is needed. Then, we can apply a series of nonconvex optimization techniques, such as stochastic gradient descent and Adam [13]. However, due to exponentially many solutions of subgraph detection in interdependent networks, it is infeasible to search the exact solution for a large network (e.g. $|\mathbb{V}| \geq 10,000$) with brute force in an acceptable time [4], [14]. Hence, most existing methods for addressing this problem find an approximation or suboptimal solution heuristically within an acceptable runtime, which attempts to balance effectiveness and efficiency. To the best of our knowledge, most related studies on subgraph detection in interdependent networks focus on a specific application and lack generality. Furthermore, they are heuristic-driven without any theoretical guarantee. In this paper, we explore possible solutions for graph block-structured optimization by leveraging sparse optimization theories and approximate projections for graph-structured sparsity, aiming to provide a generic framework for subgraph detection problems in interdependent networks with tractable computation as well as provable theoretical guarantees. The contributions of our research can be summarized as follows:

- **Design of a framework for graph block-structured optimization**. We propose a novel generic framework, named **G**raph **B**lock-structured **G**radient **H**ard **T**hresholding **P**ursuit (GB-GHTP), for the graph block-structured optimization problem, which is efficient and useful for approximately solving a broad of class of subgraph detection problems in interdependent networks in nearly linear time on the network size.
- **Theoretical guarantees**. We analyze the theoretical properties of our proposed framework and prove that the framework enjoys a good convergence rate and a tight error bound on the quality of the results. The time complexity of our algorithm is also be analyzed, which is nearly linear with the network size and has provably good efficiency.
- **Comprehensive experiments in multiple practical applications**. We demonstrate that our framework can be applied to three practical applications: 1) evolving anomalous subgraph detection in dynamic networks, 2) anomalous subgraph detection in networks of networks, and 3) connected dense subgraph detection in dual networks. Comprehensive experiments are conducted on both synthetic and real-world datasets to validate the effectiveness and efficiency of our framework.

The remaining parts of this paper are organized as follows. Section II discusses related work relevant to our research. Section III introduces the relaxation and formulation of our problem. Section IV presents an efficient framework for general graph block-structured optimization and its theoretical properties. Section V shows how to model three practical applications as graph block-structured optimization problems and solve them with our framework. Comprehensive experiments on synthetic and real-world datasets are presented in Section VI. Section VII concludes the paper and describes future work.

II. RELATED WORK

A. SUBGRAPH DETECTION IN ATTRIBUTED NETWORKS

Subgraph detection in attributed networks often refers to finding those nodes and edges whose behaviors are significantly different from the behaviors of those outside the subgraphs [11]. The detected subgraphs are usually supposed to satisfy some constraints, such as connected subgraphs, dense subgraphs, compact subgraphs, subgraphs with regular shapes (e.g., circles and rectangles), and subgraphs that are isomorphic to a query graph. There are a large number of applications or problems concerned with subgraph detection. They can be listed as follows: detection of subnetwork biomarkers [15], detection of road traffic congestion events, detection of abnormally high breakage in a distribution network [16], detection of disease outbreaks [6], [17], and event detection in social networks [6], [7]. According to the dynamics of attributes, attributed graphs fall into two categories: static graphs and dynamic graphs. A typical example of a static graph is the molecular structure of proteins and nanomaterials, whose molecular structure does not change over time [2], [3]. Social networks are a good example of dynamic graphs, in which friendship links are added or removed at any time; thus, the graph changes over time. For subgraph detection in static networks, those methods can be further divided into two parts, which handle spatial networks and complex networks. For spatial networks like a street network, most studies are statistical approaches. These methods typically assume that the attributes (e.g., traffic volumes in a street intersection) follow some distribution in Euclidean space. The goal is then to detect whether there exists a subarea where the attributes are in the same distribution but with a higher density parameter. For example, expectation-based statistics can be used to scan subsets and find anomalous space areas [18], while these methods do not consider any topological constraints. Others consider graph structures and propose the graph scan method [19] to detect connected subgraphs. Some studies also extend the graph scan method and introduce soft constraints on temporal consistency to find dynamic patterns [20]. The graph scan methods and their variants use the LTSS property, which rules out subgraphs that are suboptimal and dramatically reduces the search space [19]. For subgraph detection in static complex networks, the nonparametric heterogeneous graph scan was proposed to detect events in heterogeneous social networks [6] and can be used for civil unrest prediction, rare disease outbreak detection, and early detection of human rights events. All of the above methods are statistical approaches and cannot provide any theoretical guarantee. Alternatively, a class of subgraph detection problems can be framed as a general submodular (but not monotone) maximization problem and used to detect activity in networks [7]. Another work relaxes the nonconvex constraints to convex and introduces the constraints as a regularization

term, which provides a performance bound [17]. However, the method is not scalable to large graphs ($\geq 1,000$ nodes). For subgraph detection in dynamic networks research, topological constraints on graphs and attributes' dependencies between different timestamps are considered. Meden [21] mines the heaviest dynamic subgraph (region) with the maximum score defined on nodes and edges. NetSpot [14] defines smoothness between different timestamps. Both methods prune most instances to be searched and make themselves scalable. Dynamic GraphScan [20] uses expectation-based statistics and soft consistency constraints, which are efficent and can scale with the instance size. These methods for dynamic networks are statistical approaches without theoretical guarantees and are designed for some specific scenarios, which restricts their applications. The aforementioned algorithms [4]–[6], [9], [10] mainly leverage statistical theories, which cannot optimize on raw data and thus rarely give any theoretical guarantee for optimization.

B. STRUCTURED SPARSE OPTIMIZATION

In the past decade, sparsity has arisen as an important tool in many fields, such as statistics, signal processing, and machine learning. In many settings, sparsity is useful because it enables us to identify structures in high-dimensional data while still remaining a mathematically tractable concept [22]. Structured sparsity models refer to a class of sparsity models that discover patterns in high-dimensional data with prior knowledge about their structures. Recently, a number of structured sparsity models defined through trees [23], groups, clusters, and paths [24] have been proposed. Generally, an optimization problem based on a structured sparsity model can be defined as

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{s.t.} \quad \text{supp}(x) \in \mathbb{M} \tag{3}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable objective function, the support set $\text{supp}(x)$ denotes the set of indices of nonzero entries in x, and $\mathbb{M}$ is the sparsity model and represents a family of structured supports, i.e., $\mathbb{M} = \{S_1, S_2, \ldots, S_L\}$, where $S_i \subseteq [n] = \{1, 2, \ldots, n\}$ satisfies a certain structure constraint (e.g., groups, trees or clusters). For example, a k-sparsity model is defined as $\mathbb{M} = \{S \subseteq [n] || S| \leq k\}$.

Structured sparse optimization can be implemented by two main methods: 1) encoding the structured sparsity model as an induced norm and embedding it into the objective function, where the induced norm is usually non-Euclidean and nonsmooth. Or we can 2) leverage a projection oracle on $\mathbb{M}$, which is defined as

$$\text{P}(b) = \underset{x\in\mathbb{R}^n}{\text{argmin}} \|b - x\|_2^2 \quad \text{s.t.} \quad \text{supp}(x) \in \mathbb{M} \tag{4}$$

and decompose the problem into two subproblems, including the optimization of $f(x)$ independent of the structured sparsity constraints and the projection problem (4). Most methods via a projection oracle require exact solutions to the projection problem (4), which are usually unavailable. For instance, if we require the connectivity of the $\text{supp}(x)$, the projection oracle is reduced from prize collection Steiner tree problem that is NP-hard. However, when we use an approximate projection, the theoretical guarantees of those methods no longer hold. A recent approach named Graph-CoSaMP [22], [25], attempts to introduce an approximation framework for sparsity structured models defined via graphs and provide a theoretical guarantee, in which an efficient approximate projection algorithm that runs in nearly linear time is proposed. There are two components in the approximate projection algorithm, including head and tail approximate projections, which provide a theoretical guarantee as long as they are utilized jointly. Although Graph-CoSaMP shows good performance for finding trees or clusters in data with graph structures, it is only applicable in linear regression or compressive sensing. Some works have generalized Graph-CoSaMP and proposed algorithms for graph-structured sparsity optimization problems [24], [26], [27]. They evaluate their methods in the problems of connected subgraph detection and interesting subspace detection. Reference [24] proposed Graph-IHT and Graph-GHTP to solve the structured optimization problem on a single graph. These methods are variants of iterative hard thresholding (IHT) and gradient hard thresholding pursuit (GHTP) [28], respectively, in which the projection oracle is approximately solved by the head and tail approximations. References [26], [27] used the same idea as [24] on match pursuit (MP) and designed the Graph-MP and SG-Pursuit methods. Graph-MP aims for subgraph detection, while the task of SG-Pursuit detects subspace. Our work generalizes the aforementioned ideas in [24], [26], [27] in that we can solve combinatorial optimization problems with topological constraints via structured sparsity optimization. Other works such as [5], [10] have been designed for uncovering specific-shape subgraphs via nonparametric statistics, which do not possess the ability to run on raw data. More importantly, those works only handle structural constraints defined on an isolated network (i.e. $\mathbb{M}$ defined on a single graph in problem (3)). We propose a generic framework to solve general optimization problems on multiple networks with interdependency. Thus, previous methods [24], [26], [27] are special cases of our framework when constraints are defined on a single network.

The aforementioned methods, IHT and MP, are two classical algorithms for general sparse optimization problems. GHTP is an improved method based on IHT and GHTP, which iterates between 1) a standard gradient descent step and 2) a hard thresholding step [28]. Our framework follows the iterative scheme of GHTP and integrates the approximate projection oracles head and tail in Graph-CoSaMP. Then we generalize GHTP from the trivial sparse optimization problem to a generic problem setup of graph block-structured optimization, and propose a framework to optimize structured data with multiple blocks, which can be deployed to detect subgraphs across interdependent networks. In this setting, structured optimization on single graph is a special case of our framework for multiple-block-structured optimization when the number of blocks is 1.

C. NETWORK ANALYSIS

Complex systems can be modeled as complex networks, which usually encompass many subsystems that interact with or depend on each other. These networks composed of multiple interdependent networks are also known as multilayer networks, networks of networks or multiplex networks [29]. Studies in this field mostly study static and evolving statistical characteristics of networks, which ignore attributes on nodes and edges. Recent works [12], [30], [31] have studied the percolation properties of networks with interdependency on each other, which can be utilized to analyze the robustness of networks. In particular, [31] discussed the vulnerability of interdependent spatially embedded networks. Furthermore, epidemics in interdependent networks, which can depict disease transmission, were studied in [32], [33]. Apart from the percolation properties of networks, another network application is data transmission, which is critical in the era of the Internet. The references [34], [35] studied the fractional factor problem on fractional critical deleted graphs, which can help make better decisions for dividing large data packets into small packets and improve the digital communication efficiency. The molecular structures in microbiology or nanomaterials can be expressed as a network, where genes, proteins, cells or atoms are denoted as nodes, and the connected elements are regarded as edges. Then, researchers could calculate the topological indices of molecular structures, which are definitions from graph theory, to test the chemical, physical [3], biological [2], [15], and pharmaceutical [2] properties of various materials. These conclusions have promising application prospects in bioengineering and nanoscience.

There are many other research lines in this field. For example, a framework has been established to study the community structure of time-dependent networks, which handles various types of links (multiplexity) and multiple scales [36]. As a special network structure, ontology has attracted much attention. Researchers presented an efficient partial multi-dividing ontology algorithm to obtain a semantic matching set of concepts and rank them according to their similarities [37]. It must be noted that all of the studies in this part do not involve features on nodes or edges, which is the largest difference from our work.

III. PROBLEM FORMULATION

The problem (2) is nonlinear, combinatorial, and nonconvex. To make use of advanced numerical optimization techniques, such as the stochastic gradient descent and coordinate descent method, which have been proven to be impressively simple, efficient, and effective in nonconvex problems (e.g. deep learning) [13], we first reformulate the original combinatorial problem (2) as an equivalent 0-1 integer programming problem:

$$\min_{x^k \in \{0,1\}^{N_k}} f(x^1, x^2, \ldots, x^k) + \sum_{i \neq j} g(x^i, x^j),$$
$$\text{supp}(x^k) \in \mathbb{M}_k, \quad k = 1, \ldots, K \tag{5}$$

where x^k denotes binary variables of nodes in the k^{th} block, $\text{supp}(x^k)$ refers to the set of indices of nonzero entries in x^k, which represents a subset of nodes in block k, and $\mathbb{M}_k$ represents all possible subsets of nodes that satisfy the topological constraint on graph $\mathbb{G}^k$. The set composed of the supports $\text{supp}(x^k)$ refers to a subgraph whose corresponding variables x have values of 1 s and minimize the objective function. To make it easy to solve the problem (5) and take advantage of existing advanced numerical optimization methods, the domain can be further relaxed from $x \in \{0, 1\}$ to $x \in [0, 1]$ (i.e., from integer to continuous), and then the problem becomes a ***numerical optimization*** problem with graph-structured sparsity constraints that are nonconvex and combinatorial. We detail the formal problem setting in the following.

Given a network $\mathbb{G} = (\mathbb{V}, \mathbb{E}, W)$, where $\mathbb{V} = \{1, \cdots, N\}$ is the ground set of nodes, $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ is the ground set of edges, $W = [w_1, \cdots, w_N] \in \mathbb{R}^{P \times N}$ is the feature matrix defined on nodes, and $w_i \in \mathbb{R}^P$ is the feature vector of node i, the node set $\mathbb{V}$ has a multiple-block structure and can be decomposed to K disjoint subsets (blocks): $\mathbb{V} = \mathbb{V}^1 \cup \cdots \cup \mathbb{V}^K$, where $N_k = |\mathbb{V}^k|$ refers to the size of the node set of block $\mathbb{V}^k$. After relaxation of the domain from 0, 1 to [0, 1], the subgraph detection problem with multiple blocks can be formulated as the following general graph block-structured optimization problem:

$$\min_{x=(x^1,\ldots,x^K)} F(x) = f(x^1, \cdots, x^K) + \sum_{i \neq j}^{K} g(x^i, x^j),$$
$$\text{s.t. } \text{supp}(x^k) \in \mathbb{M}_k(\mathbb{G}, s_k), \quad k = 1, \cdots, K \tag{6}$$

where the vector $x \in \mathbb{R}^N$ is partitioned into multiple disjoint blocks $x^1 \in \mathbb{R}^{N_1}, \cdots, x^K \in \mathbb{R}^{N_K}$, $F(\cdot)$ is a continuous differentiable and convex function, $\text{supp}(x^k)$ denotes the support set of vector x^k, $\mathbb{M}_k(\mathbb{G}, s_k)$ denotes all possible subsets of vertices in $\mathbb{G}$ that satisfy a certain predefined topological constraint on block k. The functions $f(\cdot)$ and $g(\cdot)$ are defined based on the feature matrix W, and can be used to formulate the cost function and dependencies among blocks respectively.

One example of topological constraints for defining $\mathbb{M}_k(\mathbb{G}, s_k)$ is a connected subgraph, and we can formally define it as follows:

$$\mathbb{M}_k(\mathbb{G}, s_k) = \{S | S \subseteq \mathbb{V}^k; |S| \leq s_k; \mathbb{G}_S \text{ is connected.}\} \tag{7}$$

where s_k is an upper bound of the number of S and defined by users, $S \subseteq \mathbb{V}^k$, and $\mathbb{G}_S$ refers to the induced subgraph by S. The topological constraints can be any graph-structured sparsity constraints on $\mathbb{G}_S$, such as connected subgraphs, dense subgraphs, and compact subgraphs [27]. Moreover, we do not restrict all $\text{supp}(x^1), \cdots, \text{supp}(x^K)$ satisfying an identical topological constraint. An illustration of the problem formulation for connected subgraph detection in interdependent networks can be found in Fig. 2.

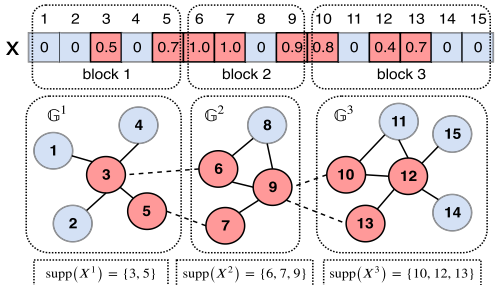


FIGURE 2. Illustration of connected subgraph detection in interdependent networks. In this example, the interdependent networks with 3 blocks $\{\mathbb{G}^1, \mathbb{G}^2, \mathbb{G}^3\}$ are given. The dashed lines represent the connections across different networks, and solid lines represent connections between nodes in the same network. These two types of lines may characterize different relationships in practical applications. Those red nodes are what we are interested in, and their corresponding entries in vector x should be nonzero, while others should be 0 s once the $F(x)$ is minimized.

IV. METHODOLOGY

A. PRELIMINARIES

The relaxed problem (6) is hard to solve due to its nonconvex topological constraints. Intuitively, we could apply projected gradient descent to find an approximate solution, in which we first 1) optimize the objective function independent of the topological constraints, and then 2) project the intermediate solution to the feasible space that satisfies the topological constraints. The projection can be defined as (4). However, this trivial projection oracle is NP-hard for popular network-structural constraints. For example, for connected subgraphs, $\mathrm{P}(x)$ can be reduced from the prize collecting Steiner tree (PCST) problem, which is known to be NP-hard [25]. If we consider an approximation of the projection oracle $\mathrm{P}(x)$, the projected gradient descent algorithm becomes a heuristic algorithm with a slow convergence rate [25]. Fortunately, there exist some approximation methods for this NP-hard projection problem that provide the possibility to perform theoretical analysis. In the following, we introduce the approximate projection method that our method depends on. Note that any other approximate projection methods can also be applied to our framework as long as they provide a theoretical guarantee.

1) APPROXIMATE ALGORITHMS FOR THE PROJECTION ORACLE P(x)

There are two major components related to the support of the topological constraint "$\mathrm{supp}(x) \in \mathbb{M}(\mathbb{G}, s)$", including head and tail projections [22]. The key idea is that, suppose we can find a good intermediate solution x that does not satisfy these topological constraints, these two types of projections can be used to find good *approximations* of x in the feasible space defined by $\mathbb{M}(\mathbb{G}, s)$.

- **Tail approximation** ($\mathrm{T}(x)$): Find an $S \subseteq \mathbb{V}$ such that

$$\|x - x_S\|_2 \leq c_{\mathrm{T}} \cdot \min_{S' \in \mathbb{M}(\mathbb{G}, s_{\mathrm{T}})} \|x - x_{S'}\|_2 \tag{8}$$

  where $c_{\mathrm{T}} \geq 1$, $s_{\mathrm{T}} = 5s$, and x_S is the restriction of x to indices in S: we have $(x_S)_i = x_i$ for $i \in S$ and $(x_S)_i = 0$ otherwise. When $c_{\mathrm{T}} = 1$, $\mathrm{T}(x)$ returns an optimal solution to the problem: $\min_{S' \in \mathbb{M}(\mathbb{G}, s)} \|x - x_{S'}\|_2$. When $c_{\mathrm{T}} > 1$, $\mathrm{T}(x)$ returns an approximate solution to this problem with the approximate factor c_{T}.

- **Head approximation** ($\mathrm{H}(x)$): Find an $S \subseteq \mathbb{V}$ such that

$$\|x_S\| \geq c_{\mathrm{H}} \cdot \max_{S' \in \mathbb{M}(\mathbb{G}, s_{\mathrm{H}})} \|x_{S'}\|_2 \tag{9}$$

  where $c_{\mathrm{H}} \leq 1$ and $s_{\mathrm{H}} = 2s$. When $c_{\mathrm{H}} = 1$, $\mathrm{H}(x)$ returns an optimal solution to the problem: $\max_{S' \in \mathbb{M}(\mathbb{G}, s)} \|x_{S'}\|_2$. When $c_{\mathrm{H}} < 1$, $\mathrm{H}(x)$ returns an approximate solution to this problem with the approximate factor c_{H}.

It can be readily proven that $\mathrm{T}(x) = \mathrm{H}(x) = \mathrm{P}(x)$ when $c_{\mathrm{T}} = c_{\mathrm{H}} = 1$. Although the head and tail projections are NP-hard when we restrict $c_{\mathrm{T}} = 1$ and $c_{\mathrm{H}} = 1$, these two projections can still be implemented in nearly linear time when approximate solutions with $c_{\mathrm{T}} > 1$ and $c_{\mathrm{H}} < 1$ are allowed. Moreover, the joint utilization of both head and tail projections is critical in the design of approximate algorithms for network topology-related optimization problems [22], [24]–[26]. It is claimed that these two approximations can be generalized to graph-structured sparsity models that are defined on different graph topological constraints, such as density, k-core, radius, cut, or various others, as long as their corresponding head and tail projections are available [26].

2) GRADIENT HARD THRESHOLDING PURSUIT

As mentioned in Section II, the gradient hard thresholding pursuit is an iterative method for the sparsity constrained convex optimization problem, which is defined as

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad \|x\|_0 \leq s$$

In the iterative procedures, a sequence of intermediate s-sparse vectors $x^0, x^1, \ldots$ from an initial sparse approximation x^0 (typically $x^0 = 0$) are generated. At the i^{th} iteration, the GHTP can be divided into three steps

1) $\tilde{x}^i = x^{i-1} - \eta \nabla f(x^{i-1})$, this step applies the gradient descent at the point x^{i-1} with step size η;
2) $\Omega^i = \mathrm{P}(\mathrm{supp}(\tilde{x}^i))$, the s coordinates of the vector x^i that have the largest magnitude are selected as the support;
3) $x^i = \underset{x}{\mathrm{argmin}} f(x) \quad \mathrm{supp}(x) \in \Omega^i$, a vector that minimizes the objective function is returned.

The first step of GHTP is a standard gradient descent; the second step gives a direction in which pursuing the minimization will be most effective; and the third step, often referred to as debiasing, has been shown to improve the performance in some algorithms [28]. These steps continue until the algorithm reaches a terminating condition, e.g., on the change in the objective function or the change in the estimated minimum from the previous iteration.

GHTP has proven its performance in optimization for the vanilla s-sparsity model. However this algorithm cannot handle optimization problems with graph block-structured constraints that are described in problem (6) due to the unavailability of the exact solution for the projection oracle in step 2) of GHTP. Meanwhile, the aforementioned head and tail approximations provide us with an effective and efficient method for achieving the approximation for the graph-structured sparsity model. Inspired by these

Algorithm 1 Graph Block-Structured Gradient Hard Thresholding Pursuit (GB-GHTP)

Input: Input graph $\mathbb{G}$, maximum subgraph size s_k on each block, and step size η.
Output: The estimated vector $\hat{x}$ and the corresponding connected subgraph $\mathbb{S}$.
Initialization, $i = 0$, $x^i = (x^{1,i}, \ldots, x^{K,i}) = 0$, $b^i = (b^{1,i}, \ldots, b^{K,i})$, $\Psi = \bigcup_{k=1}^{K} \Psi^k$, $k = 1, \ldots, K$.

1: **repeat**
2: **for** $k = 1, \ldots, K$ **do**
3: $\Omega^k \leftarrow \mathrm{H}(\nabla_{x^k} F(x^i))$ ▷ Head projection
4: $\Gamma^k \leftarrow \mathrm{supp}(x^{k,i} - \eta \cdot (\nabla_{x^k} F(x^i))_{\Omega^k})$
5: **end for**
6: $b^i \leftarrow \mathrm{argmin}_{x \in \mathbb{R}^n} F(x)$ s.t. $\mathrm{supp}(x^k) \subseteq \Gamma^k$ (10)
7: **for** $k = 1, \ldots, K$ **do**
8: $\Psi^k \leftarrow \mathrm{T}(b^{k,i})$ ▷ Tail projection
9: $x^{k,i+1} \leftarrow (b^{k,i})_{\Psi^k}$
10: **end for**
11: $i \leftarrow i + 1$
12: **until** halting condition holds
13: **return** $\hat{x} = x^i$ and $\mathbb{S} = \mathbb{G}_\Psi$

two algorithms, we generalize the GHTP algorithm to graph block-structured optimization by integrating head and tail approximate projections, and propose a novel framework named as **G**raph **B**lock-structured **G**radient **H**ard **T**hresholding **P**ursuit. We claim that GB-GHTP maintains the good optimization power as the GHTP method, efficiency and effectiveness of head and tail approximate projections, and provides a theoretical guarantee. The key idea of GB-GHTP is to alternatively search for a close-to-optimal solution by solving easier subproblems for graph $\mathbb{G}$ with K blocks in each iteration i until convergence. We will describe the GB-GHTP algorithm in detail and its theoretical properties in the rest of this section.

B. GRAPH BLOCK-STRUCTURED GRADIENT HARD THRESHOLDING PURSUIT

The GB-GHTP algorithm generalizes the graident hard thresholding pursuit algorithm to our problem, where multiple graph block-structured constraints are imposed on the variables. We outline the procedures of GB-GHTP in Algorithm 1. This algorithm also follows the scheme described in the gradient hard thresholding pursuit. We decompose it into three steps:

1) (Lines 2 $\sim$ 5) alternatively use head projection for partial derivative on each block and find a tentative gradient update to obtain a potentially good direction, in which pursuing optimization will be most effective;
2) (Line 6) optimizes the objective function in the union of sets Γ^k;
3) (Lines 7 $\sim$ 10) alternatively apply tail projection to project the intermediate results to a feasible space, where the final results satisfy some topological constraints.

Furthermore, we utilize the block-coordinate descent method with proximal linear update [38], [39] to solve the problem (10). This method has been analyzed and applied to both convex and nonconvex problems [40]–[42] and shows good performance empirically. Block-coordinate descent is a generalization of the alternating minimization method that has been applied to a variety of problems, such as the expectation-maximization (EM) algorithm [43]. In addition, we utilize the proximal linear update that ensures the convergence of the algorithm on convex problems with convex constraints "$\mathrm{supp}(x^k) \subseteq \Gamma^k$". The proximal linear update in our scenario is defined by

$$x^{k,t+1} = \mathrm{argmin}_{x^k} F(\hat{x}^t) + \langle \nabla_{x^k} F(\hat{x}^{k,t}, \hat{x}^{\neq k,t}), x^k - \hat{x}^{k,t} \rangle + \frac{1}{2\alpha^{k,t}} \|x^k - \hat{x}^{k,t}\|_2 \quad \text{s.t.} \quad \mathrm{supp}(x^k) \subseteq \Gamma^k \tag{11}$$

where $\alpha^{k,t}$ serves as a step size and can be set as the reciprocal of the Lipschiz constant of $\nabla_{x^k} F(\hat{x}^{k,t}, \hat{x}^{\neq k,t})$, and $\hat{x}^{k,t}$ is an extrapolated point that helps accelerate the convergence of the proximal point update scheme:

$$\hat{x}^{k,t} = x^{k,t} + \omega_t (x^{k,t} - x^{k,t-1}) \tag{12}$$

where $\omega_t \geq 0$ is an extrapolation weight. [44] suggests setting hyperparameters $\omega_{t+1} = (\rho_{t+1} - 1)/\rho_{t+1}$, with $\rho_0 = 1, \omega_0 = 0$, and $\rho_{t+1} = (1 + \sqrt{1 + 4\rho_t^2})/2$, to speed up the algorithm.

The objective function in problem (11) is simply the second-order Taylor approximation of function $F(\cdot)$ with the Hessian matrix replaced by the identity matrix. We can easily derive and implement the closed-form solution of the objective function in problem (11) and then project it to the feasible space, which is convex. Mathematically, the solution of problem (11) is:

$$x^{k,t+1} = \mathrm{P}\left(x^{k,t} - \alpha^{k,t} \cdot \nabla_{x^k} F(\hat{x}^{k,t}, \hat{x}^{\neq k,t})\right) \tag{13}$$

where $\mathrm{P}(b) = \mathrm{argmin}_{x^k \in \mathbb{R}^n} \|b - x^k\|_2^2$ s.t. $\mathrm{supp}(x^k) \subseteq \Gamma^k$.

Note that the feasible set Γ^k is convex. The overall block-coordinate gradient projection method on a convex function with convex constraints (Algorithm 2) has a sublinear rate of convergence [39], [40].

C. THEORETICAL ANALYSIS

In this section, we analyze the theoretical properties of GB-GHTP. To guarantee the convergence of our framework and the accuracy of estimates, we require the objective function $F(\cdot)$ satisfying the weak restricted strong concvexity (WRSC) condition, which is a variant of the restricted strong convexity (RSC) [28] and defined as

Definition 1 (Weak Restricted Strong Convexity (WRSC)): For some $\xi > 0$ and $0 < \delta < 1$, a function $F(\cdot)$ has the weak restricted strong convexity if for any $x, y \in \mathbb{R}^N$ and $S \in \mathbb{M}$ with $\mathrm{supp}(x) \cup \mathrm{supp}(y) \subseteq S$, the following inequality holds:

$$\|x - y - \xi \nabla_S F(x) + \xi \nabla_S F(y)\|_2 \leq \delta \|x - y\|_2 \tag{14}$$

where $x = (x^1, \ldots, x^K), y = (y^1, \ldots, y^K), x^k, y^k \in \mathbb{R}^{N_k}$, $k = 1, \ldots, K$, topological constraint $\mathbb{M}$ can be expressed as $\mathbb{M}(\mathbb{G}, s) = \bigcup_{k=1}^{K} \mathbb{M}_k(\mathbb{G}, s_k), s = \sum_{k=1}^{K} s_k$, and the subgraph in the k^{th} block (i.e., $\mathbb{G}^k$) is S_k, which satisfies $|S_k| \leq s_k, S_k \subseteq \mathbb{V}^k, S = \bigcup_{k=1}^{K} S_k, |S| \leq s$. Here, since constraints on blocks are independent, we use the union sign "$\bigcup$" to denote the combined model $\mathbb{M}$, in which $x \in \mathbb{M} = \{x | x^k \in \mathbb{M}_k(\mathbb{G}, s_k), k = 1, \ldots, K\}$.

Algorithm 2 Block-Coordinate Descent Method With Proximal Linear Update to Solve Problem (10)

Input: $\{\mathbb{G}^1, \ldots, \mathbb{G}^K\}$
Output: $x^{1,t}, \cdots, x^{K,t}$
Initialization: $t = 0, \epsilon = 10^{-3}, \rho_0 = 1., \omega_0 = 0.$
1: **repeat**
2: Choose index $k \in \{1, \cdots, K\}$
3: $\hat{x}^{k,t} = x^{k,t} + \omega_t(x^{k,t} - x^{k,t-1})$
4: Update $x^{k,t+1} \leftarrow \hat{x}^{k,t} - \frac{1}{\alpha^{k,t}} \nabla_{x^k} F(\hat{x}^{k,t}, \hat{x}^{\neq k,t})$
5: Project $x^{k,t+1}$ to feasible space by setting entries of $x^{k,t+1}$ to zero if the index of entry is not in set Γ^k.
6: Keep $x^{j,t+1} = x^{j,t}$, for all $j \neq k$
7: $\rho_{t+1} = (1 + \sqrt{1 + 4\rho_t^2})/2,$
8: $\omega_{t+1} = (\rho_{t+1} - 1)/\rho_{t+1},$
9: Let $t = t + 1$
10: **until** $\sum_{k=1}^{K} \|x^{k,t} - x^{k,t-1}\| \leq \epsilon$
11: **return** $\{x^{1,t}, \cdots, x^{K,t}\}$

Remark 1: We can set different s_k values for the k^{th} block. In our applications, $s_1 = \cdots = s_k = s'$, i.e., we use the same upper bound of subgraph size for all blocks.

Theorem 1: Given a graph block-structured constraint with K blocks, $\mathbb{M}(\mathbb{G}, s) = \bigcup_{k=1}^{K} \mathbb{M}_k(\mathbb{G}, s_k)$ and an objective function $F : \mathbb{R}^N \to \mathbb{R}$ satisfying the $(\xi, \delta, \mathbb{M}(\mathbb{G}, 5s))$-WRSC condition. If $\eta > 0$, then for any $x \in \mathbb{R}^N$ that satisfies some topological constraints, i.e., $\text{supp}(x) \in \mathbb{M}(\mathbb{G}, s)$, the following inequality holds in the iterations of GB-GHTP (Algorithm 1)

$$\|x^{i+1} - x\|_2 \leq \alpha \|x^i - x\|_2 + \beta \|(\nabla F(x))_{\mathrm{I}}\|_2, \tag{15}$$

where

$$\alpha_0 = c_{\mathrm{H}}(1 - \delta) - \delta, \beta_0 = \delta(1 + c_{\mathrm{H}}),$$
$$\alpha = \frac{\sqrt{2}(1 + c_{\mathrm{T}})}{1 - \delta} \left(\sqrt{1 - \alpha_0^2} + \left(1 - \frac{\eta}{\xi} + \left(2 - \frac{\eta}{\xi}\right)\delta\right)\right),$$
$$\beta = \frac{1 + c_{\mathrm{T}}}{1 - \delta} \left(2\eta + \xi + \frac{\sqrt{2}\beta_0}{\alpha_0} + \frac{\sqrt{2}\alpha_0\beta_0}{\sqrt{1 - \alpha_0^2}} \right),$$

$c_{\mathrm{H}} = \min_{k=1,\ldots,K} c_{\mathrm{H}_k}$, $c_{\mathrm{T}} = \max_{k=1,\ldots,K} c_{\mathrm{T}_k}$, *and* $\mathrm{I} = \underset{S \in \mathbb{M}(\mathbb{G}, 8s)}{\operatorname{argmax}} \|(\nabla F(x))_S\|_2$.

Proof: The proof is provided in Appendix B □

Remark 2: 1) The convergence of the GB-GHTP algorithm is controlled by the shrinkage rate $\alpha < 1$, which is satisfied if and only if $c_{\mathrm{H}}^2 > 1 - 1/(1 + c_{\mathrm{T}})^2$ when δ is small. As proven in [25], the approximation factor c_{H} of any given head approximation algorithm can be boosted to any arbitrary constant close to 1, such that the above condition is satisfied. 2) According to the WRSC condition, the function is Lipshcitz continuous, which further implies that $\|(\nabla F(x))_{\mathrm{I}}\|_2$ must be bounded. In summary, Theorem 1 indicates the convergence of our algorithms.

Theorem 2: Suppose that $x \in \mathbb{R}^N$ such that $\text{supp}(x) \in \mathbb{M}(\mathbb{G}, s)$, and $F : \mathbb{R}^N \to \mathbb{R}$ is an objective function satisfying the $(\xi, \delta, \mathbb{M}(\mathbb{G}, 8s))$-WRSC condition. If $\alpha < 1$, GB-GHTP returns an $\hat{x}$ such that $\text{supp}(\hat{x}) \in \mathbb{M}(\mathbb{G}, 5s)$ and $\|\hat{x} - x\|_2 \leq c\|(\nabla F(x))_{\mathrm{I}}\|_2$, where $c = 1 + \frac{\beta}{1-\alpha}$ is a fixed constant. In addition, GB-GHTP runs in time

$$O\left(\left(T + \sum_{k=1}^{K} |\mathbb{E}^k| \log^3 N_k\right) \log\left(\frac{\|x\|_2}{\|(\nabla F(x))_{\mathrm{I}}\|}\right)\right), \tag{16}$$

where T is the time of solving the subproblem in Line 6 of the GB-GHTP. Furthermore, if T scales linearly with N and $|E|$, then GB-GHTP scales nearly linearly with the network size N and $|E|$.

Proof: The proof is provided in Appendix C □

*Remark 3: We can run head and tail projections on blocks **in parallel**, which reduces the time cost of each iteration to $(T + |\mathbb{E}'| \log^3 N'), |\mathbb{E}'| \log^3 N' = \max_{k=1,\ldots,K} |\mathbb{E}^k| \log^3 N_k$. As mentioned in the previous subsection, which was proved in [40], the sublinear convergence rate $O(1/t)$ can be established for Algorithm 2. Hence, the iteration number of Algorithm 2 is $O(\lceil 1/\epsilon \rceil)$ when error bound is ϵ. Then, it concludes that the time complexity of Algorithm 2 (i.e., T) is $O(\lceil 1/\epsilon \rceil N)$, which further implies the GB-GHTP algorithm scales nearly linearly with N and $|\mathbb{E}|$.*

V. EXAMPLE APPLICATIONS

In this section, we show three applications of our framework: 1) evolving anomalous subgraph detection in dynamic networks, 2) anomalous subgraph detection in networks of networks, and 3) connected dense subgraph detection in dual networks. To apply our framework in these applications, we need to formulate these applications to the context of the graph block-structured optimization problem. Thus, we leverage the **elevated mean scan** (EMS) statistic to discover subgraphs whose features on vertices are anomalous/significant compared to those background vertices. The EMS statistic is used widely for detecting signals among node-level numerical features on graphs [17], [45] and is defined as

$$\frac{w^{\intercal} x}{\sqrt{x^{\intercal} 1}} \tag{17}$$

where $x \in \{0, 1\}^N$, w denotes the attribute vector of all nodes (here, we assume that each node only has one attribute). $w_i \in \mathbb{R}$ refers to the univariable feature of node i. Empirically, the EMS statistic can be maximized to precisely detect significant subset of nodes in a network. To embed the EMS statistic into our optimization framework, relaxation on its input domain from $\{0, 1\}$ to continuous space $[0, 1]$ can be introduced. Then we can detect subgraphs by minimizing the negative relaxed EMS statistic, which is defined as

$$-\frac{(w^{\intercal} x)^2}{x^{\intercal} 1} + \frac{1}{2}\|x\|_2, \quad \text{where } x \in [0, 1]^N \tag{18}$$

Note that the negative relaxed EMS statistic satisfies the RSC condition, which implies the WRSC condition [26], [28]. Then Theorem 1 is established in our applications, i.e., the deployment of the negative relaxed EMS statistic guarantees the convergence of our framework.

A. EVOLVING ANOMALOUS SUBGRAPH DETECTION IN DYNAMIC NETWORKS

Dynamic networks arise in many applications and it is an important and challenging problem to detect subgraphs in dynamic networks, which is usually NP-hard [4]. Generally, attributes on nodes of a network evolve with time, and these evolving phenomena can be characterized by three phrases: emerging, spreading, and receding over a period of time. The phrases usually represent an evolving event on networks (e.g., Fig. 1a). Assume a dynamic network spreads over K timestamps, i.e., we have K attributed networks $(\mathbb{G}^1, \ldots, \mathbb{G}^K)$, $k = 1, \ldots, K$. The evolving subgraphs refer to a consecutive sequence of subgraphs $(S_i \sim S_j, i \geq 1, j \leq K,$

$i \le k \le j, S_k \subseteq \mathbb{G}^k$), which are connected (denoted as *local connectivity constraint*) at each timestamp and two subgraphs at consecutive timestamps share some overlap vertices (denoted as *temporal consistency constraint*) [4]. Then, the evolving anomalous subgraph detection in dynamic networks problem can be formulated as a nonconvex optimization problem with a convex objective function and block-structured constraints:

$$\min_{x^1,\ldots,x^K} \sum_{k=1}^{K} \left(-\frac{(w^{k\mathsf{T}} x^k)^2}{x^{k\mathsf{T}} 1} + \frac{1}{2}\|x\|_2 \right) + \lambda \cdot \sum_{k=2}^{K} \|x^k - x^{k-1}\|_2,$$
$$\text{s.t.} \quad \text{supp}(x^k) \in \mathbb{M}_k(\mathbb{G}, s) \tag{19}$$

where the first term is the summation of the negative relaxed EMS, the second term is a soft constraint to ensure the temporal consistency of detected subgraphs between timestamps k and $k-1$, and $\lambda > 0$ is a tradeoff parameter. The final evolving subgraphs can be obtained from the support of x^k, i.e., $S_k = \text{supp}(x^k)$.

B. ANOMALOUS SUBGRAPH DETECTION IN NETWORK OF NETWORKS

Our framework can also be applied to detect anomalous subgraphs in networks. A large-scale static network with many communities can be viewed as one instance of a network of networks (trivial interdependent networks), where each community is a small block of the network. When an event is widespread in such large networks, it becomes very challenging and time consuming to apply effective models to detect the significant subgraphs. For example, one application is rumor detection and tracking in social networks. It is interesting and useful to identify a connected subgraph that depicts how a rumor spreads across different communities in a social network.

The most traditional approach is to partition a large network into several small blocks and then process them individually and independently. However, this approach affects the detection performance if those blocks are highly interdependent. By encoding the dependencies in our proposed framework, we can detect subgraphs in each individual partition of networks more efficiently without sacrificing performance.

Specifically, our proposed framework provides a feasible solution for this scenario where node dependencies among a large-scale network cannot be neglected. We can also leverage the relaxed EMS to detect signals on vertices. Then the subgraph detection problem in the network of networks can be formulated as follows:

$$\min_{x^1,\ldots,x^K} \sum_{k=1}^{K} \left(-\frac{(w^{k\mathsf{T}} x^k)^2}{x^{k\mathsf{T}} 1} + \frac{1}{2}\|x\|_2 \right) + \lambda \cdot \sum_{i\in\mathbb{V}^{k_1}, j\in\mathbb{V}^{k_2}, k_1\neq k_2} e_{ij} \cdot (x_i - x_j)^2,$$
$$\text{s.t.} \quad \text{supp}(x^k) \in \mathbb{M}_k(\mathbb{G}, s) \tag{20}$$

where the first term is the summation of the negative relaxed EMS, the second term is a soft constraint on bridge vertices of two networks (blocks) to ensure dependencies: if vertex i and vertex j are connected but in two different networks (i.e., edge (i, j) is a graph cut), $e_{ij} = 1$; otherwise, $e_{ij} = 0$. x^i and x^j are i^{th} and j^{th} entries of x, and $\lambda > 0$ is a tradeoff parameter.

C. CONNECTED DENSE SUBGRAPH DETECTION IN DUAL NETWORKS

In real-life applications, there exist many *dual networks*, in which one network represents the physical world and the other network represents the conceptual world [46]. A typical example of dual networks is citation networks. In the research community, there are many collaborations between different researchers and some of them share similar research interests. Two different networks can be constructed in this context. One network models coauthor relationships, in which vertices are authors and edges represent that two authors coauthored one paper, i.e., physical interactions between authors. Another network models the research interest similarity between authors, in which an edge denotes that two authors have similar research interests (an edge can be constructed by measuring the similarity of publications of two researchers). The research interest network is conceptual. It is interesting to find a group of *active* researchers in which there exist collaborations between those researchers and their research interests are similar. Three goals are considered in this application. First, authors should be *active*, which can be detected by maximizing some metric defined on authors' features. Second, authors should have direct coauthorship or indirect coauthorship via other authors. This goal can be achieved by imposing a *connected* constraint on the coauthorship network. Last, we expect those authors to share similar research interests, which can be reflected by *dense* connections between authors on the research interest network. Here, we give a formal formulation of dual networks. Given two graphs $\mathbb{G}^1(\mathbb{V}, \mathbb{E}^1)$ and $\mathbb{G}^2(\mathbb{V}, \mathbb{E}^2)$ that have the same node set, i.e., $\mathbb{V}$ but have different edge sets, i.e., $\mathbb{E}^1$ and $\mathbb{E}^2$. For brevity, we also use $\mathbb{G}(\mathbb{V}, \mathbb{E}^1, \mathbb{E}^2)$ to represent the dual networks. The subgraphs induced by vertex set $S \subseteq \mathbb{V}$ in the physical network and conceptual network are denoted as $\mathbb{G}^1_S$ and $\mathbb{G}^2_S$, respectively. Therefore, the connected dense subgraph detection problem in dual networks can be defined as follows

Definition 2: Given dual networks $\mathbb{G}(\mathbb{V}, \mathbb{E}^1, \mathbb{E}^2)$, the connected dense subgraph detection in dual networks refers to finding a set of vertices such that their induced subgraphs satisfy:

1) *$\mathbb{G}^1_S$ is connected;*
2) *the density of $\mathbb{G}^2_S$ is larger than a threshold α, denoted as $\rho(\mathbb{G}^2_S) \ge \alpha$;*
3) *the function defined on the network features is maximized.*

Similar to the previous two applications, we use the EMS statistic to detect significant nodes. We can formulate this problem with a mathematical format and express it as an optimization problem with constraints on two networks

$$\min_{x,y} \left(-\frac{(w^{\mathsf{T}} x)^2}{x^{\mathsf{T}} 1} + \frac{1}{2}\|x\|_2 \right) + \left(-\frac{(w^{\mathsf{T}} y)^2}{y^{\mathsf{T}} 1} + \frac{1}{2}\|y\|_2 \right) + \lambda\|x - y\|_2$$
$$\text{s.t.} \quad \mathbb{G}^1(\text{supp}(x)) \text{ is connected}$$
$$\rho(\mathbb{G}^2(\text{supp}(y))) \ge \alpha \tag{21}$$

Algorithm 3 Graph Block-Structured Gradient Hard Thresholding Pursuit With Parallelism

Input: $\{\mathbb{G}^1, \ldots, \mathbb{G}^K\}$
Output: $x^{1,t}, \cdots, x^{K,t}$
Initialization: $i = 0, x^{k,i}$ = initial vectors, $k = 1, \ldots, K$, τ = number of processors, n = number of blocks
1: **repeat**
2: $\Omega_{x^k} = \mathrm{H}\Big(\nabla_{x^k} F(x^{1,i}, \ldots, x^{K,i})\Big), \forall k$
3: $\Gamma^k \leftarrow \mathrm{supp}(x^{k,i} - \eta \cdot (\nabla_{x^k} F(x^i))_{\Omega^k}), \forall k$
4: $t = 0$, choose $x^0 \in \mathbb{R}^N$, $x^0 = z^0$, and $\theta_0 = \frac{\tau}{n}$
5: **repeat**
6: $y^t = (1 - \theta_t)x^t + \theta_t x^t$
7: Generate random set of blocks $S_t \subseteq \{1, \ldots, K\}$
8: $z^{t+1} = z^t$
9: **for** $k \in S_t$ **do**
10: $z^{k,t+1} = \mathrm{argmin}_{z \in \mathbb{R}^{N_k}} \left\langle \nabla_{y^k} F(y^t), z - y^{k,t} \right\rangle + \frac{n\theta_t}{2\tau} \left\| z - z^{k,t} \right\|_2^2 \quad \mathrm{supp}(z^k) \subseteq \Omega_{x^k}$
11: **end for**
12: $x^{t+1} = y^t + \frac{n}{\tau}\theta_t (z^{t+1} - z^t)$
13: $\theta_{t+1} = \frac{\sqrt{\theta_t^4 + 4\theta_t^2} - \theta_t^2}{2}$
14: let $t = t + 1$
15: **until** $\|x^t - x^{t-1}\|_2 \leq \epsilon$
16: $\Psi_{x^k} = \mathrm{T}(x^{k,i}), x^{k,i} = [x^{k,i}]_{\Psi_{x^k}}, \forall k$
17: let $i = i + 1$
18: **until** $\sum_{k=1}^K \|x^{k,i+1} - x^{k,i}\| \leq \epsilon$
19: **return** $\{x^{1,i}, \cdots, x^{K,i}\}$

where x and y are associated coefficients with the physical network and conceptual network respectively. The first and second terms are the negative relaxed EMS statistics of two networks, and the third term is a penalty to make the detected subgraphs in both networks as overlap as possible. $\rho(\mathbb{G}^2(\mathrm{supp}(y)))$ denotes the density of the subgraph induced by $\mathrm{supp}(y)$ in the conceptual network. To reduce the complexity of the problem, we assume that the corresponding nodes on both networks share the same features, i.e., the same w.

In addition, we devise a parallel version of our framework (Algorithm 3) to speed up the computation by integrating the APPROX algorithm, a randomized coordinate descent method proposed in [47]. In Algorithm 3, the head projections (Line 2), tail projections (Line 16), and steps from Line 9 to Line 11 are parallelizable. The block from Line 4 to Line 15 can be run in parallel and boosts the convergence rate of solving problem (10) from $O(1/t)$ to $O(1/t^2)$, and the theoretical proof and more details are provided in [47]. The parallel part starts from $x^0 \in \mathbb{R}^N$ (Line 4) and generates three vector sequences denoted as $x^t, y^t, z^t \geq 0$. In Line 6, y^t is defined as a convex combination of x^t and z^t. In Line 7, a set of random blocks S_t are sampled and then Lines 9–12 are performed in parallel.

VI. EXPERIMENTS

We evaluate our GB-GHTP framework in the aforementioned three practical applications on both synthetic and real datasets. The details of the experiments and the discussion of the results are reported in this section and are organized by applications. For each application, we perform experiments on both synthetic and real-world datasets. These real-world datasets are 1) water pollution data, 2) traffic data, 3) biological data, and 4) citation data, which cover many aspects of daily life and demonstrate that our framework has extensive applications.

TABLE 1. The statistics of datasets for the 1st application.

Datasets	Statistics			
	Nodes	Edges	Timestamps	Resolution
Synthetic	3,000	11,984	7	N/A
Water Pollution	12,527	14,831	8	60 min.
Washington D.C.	1,188	1,323	17	60 min.
Beijing	59,000	70,317	12	10 min.

A. EVOLVING ANOMALOUS SUBGRAPH DETECTION IN DYNAMIC NETWORKS

1) SYNTHETIC DATASETS

We construct network structures via the Barabási-Albert preferential attachment model [48], which is used to generate scale-free networks. In each synthetic instance, 7 networks with the same structure are generated and represent observations of a network at different timestamps. To simulate the dynamic process of attributes on a graph, we generate subgraphs on all networks one by one with random walks. The subgraphs of two consecutive timestamps must share 50% of node overlap, which characterizes the temporal dependency of node attributes at different timestamps. The univariate feature of each node in and not in the subgraph is generated in normal distributions $\mathbb{N}(\mu, 1)$ and $\mathbb{N}(0, 1)$ respectively. Fifty instances are generated for each setting of $\mu = \{3, 4, 5\}$. When μ is small, the signals on nodes have more noise, and it is more difficult to distinguish background nodes and anomalous nodes based on univariate features.

2) REAL-WORLD DATASETS

1) *Water Pollution Dataset*: a real-world network provided in the Battle of the Water Sensor Networks (BWSN) [49]. Among the nodes of the network, there were 25 nodes with chemical contaminant plumes, which are distributed in the network and produced a contaminated subarea. Each of nodes was associated with a sensor, which reports 1 when the node is polluted; otherwise, 0. The spread of pollution was monitored by these sensors for a period of 8 hours and reported for each hour. 2) *Washington D.C. Road Traffic Dataset*: a road traffic dataset collected from June 1, 2013 to March 31, 2014 in Washington D.C. [50]. We use the data from 6AM to 10PM with a time resolution of 60 minutes (17 timestamps). 3) *Beijing Road Traffic Dataset*: the dataset contains the real-time traffic conditions of four days. We use the data between 5PM and 7PM of the first day with a time resolution of 10 minutes (a totally 12 timestamps) [26]. For both traffic datasets, the difference between the reference speed

TABLE 2. The results on synthetic datasets with different μ.

Methods	$\mu = 3$			$\mu = 4$			$\mu = 5$		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Meden	0.7588	0.7342	0.7453	0.8836	0.8591	0.8709	0.9646	0.9145	0.9388
NetSpot	0.6658	0.7267	0.6947	0.7615	0.7922	0.7763	0.7956	0.8185	0.8068
GB-GHTP	0.7796	0.7679	**0.7736**	0.8661	0.9592	**0.9102**	0.9619	0.9891	**0.9753**

TABLE 3. The results on the Washington D.C. and Beijing datasets.

Methods	Washington D.C.			Beijing		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Meden	0.7076	0.7662	0.7342	0.6424	0.7509	0.6882
NetSpot	0.5823	0.7098	0.6367	0.6789	0.7351	0.6973
GB-GHTP	0.9162	0.6462	**0.7551**	0.7440	0.9212	**0.8141**

and current speed is used as the node feature, and the true congested roads are provided. The statistics of all datasets are provided in Table 1. The column ''Nodes'' gives the number of nodes per graph in the dataset, and the column ''Edges'' describes the edge size. The columns ''Timestamps'' and ''Resolution'' describe the number of timestamps and the time resolution of each dataset.

3) COMPARISON METHODS

Our method is compared with two state-of-the-art baselines: *Meden* [21] and *NetSpot* [14]. Both algorithms prune the subinterval space to speed up and are proposed to detect significantly anomalous areas in dynamic networks.

4) PERFORMANCE METRICS

We use precision, recall, and F-measure to evaluate the performance of all approaches. A higher F-measure indicates better overall quality of the detected subgraph. For synthetic datasets, the reported results of each setting are averaged over 50 instances. Those metrics are defined as follows

$$\text{Precison} = \frac{S_A \bigcup S_B}{S_A}$$

$$\text{Recall} = \frac{S_A \bigcup S_B}{S_B}$$

$$\text{F-measure} = \frac{2 \times \text{Precison} \times \text{Recall}}{\text{Precison} + \text{Recall}}$$

where S_A refers to the node subset returned by an algorithm and S_B denotes the ground set of nodes. Precision quantifies the accuracy of detected nodes that actually belong to the true subgraph. Recall reflects the coverage level of detected nodes for the true subgraph. The F-measure provides a single score that balances both the concerns of precision and recall in one number and is able to measure how well a model is.

5) RESULTS

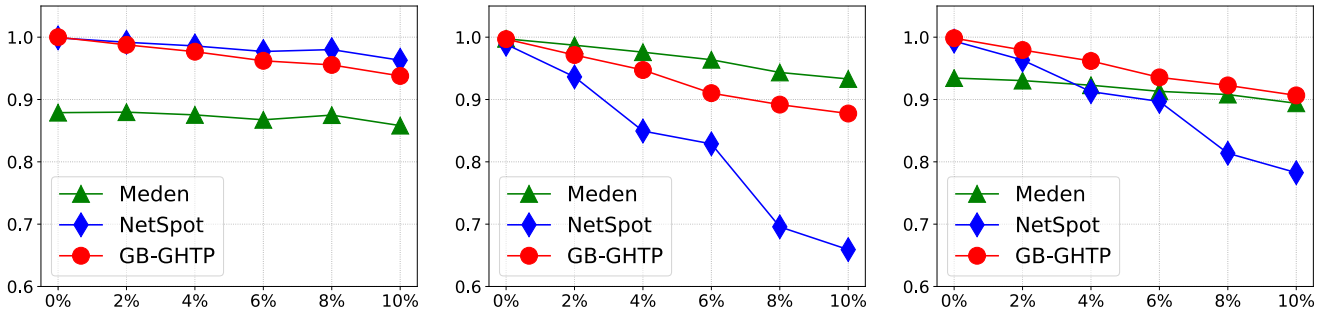
The results of all methods on synthetic and traffic datasets are listed in Table 2 and Table 3. As shown in these two tables, our method outperforms both baselines on synthetic data and real-world data. Our approach has comparable precision to other baselines and outperforms them substantially on all datasets with the aspect of F-measure. Both baselines pruning the search space are heuristic, which cannot guarantee their performance and makes the results worse than ours. We further describe our settings on traffic datasets to make our results more intuitive. In traffic datasets, a road segment corresponds to an edge in the network, and intersections are nodes in the networks. The task is to find the most congested area in the road network, i.e., which road segments and intersections are congested. Here, we use the difference between the average speed and reference speed of a road segment as a feature on an edge. Then, the feature on a node is obtained by taking average of features of the edges connected to the node. Thus, a road segment with smooth traffic has a small speed difference. Cars in a blocked road segment have a low average speed, which makes the speed difference large. Thus, we obtain the traffic condition of a road network by monitoring the speed difference. Specifically, we define an objective function based on the speed difference and find the most congested area by maximizing the score. In such a scenario, a higher precision means that more nodes detected by an algorithm belong to the congested area, and a higher recall means that more nodes in the congested area are found in the results.

6) ROBUSTNESS ANALYSIS

We also analyze the robustness of these methods on the water pollution dataset. Noise with different levels is injected into the water pollution dataset. Specifically, $K \in \{2, 4, 6, 8, 10\}$ percent of nodes are selected randomly and their sensor values are flipped (i.e., 0 to 1, or 1 to 0). The goal of the task is to detect connected subgraphs that correspond to the contaminated subarea at each timestamp. The results of robustness validation are reported in Fig. 3. The precision of Meden is lower than ours. NetSpot has a slightly higher precision performance. However, it decreases drastically on recall when noise increases. Our method has a more stable performance on all metrics, which indicates its better robustness.

7) PARAMETER TUNING

We use strategies recommended by authors of those approaches to tune parameters. For NetSpot, edges are weighted by comparing their p-values to a significance level threshold μ (0.01 recommended by the authors). To fit the input to NetSpot, weights of nodes can be easily averaged to obtain edge weights, which are used in the original paper [14]. It can be shown that problems with node weights and problems with edge weights are equivalent [21]. Our framework in this application has two parameters, 1) sparseness parameter s (an upper bound of the subgraph size on each block) and 2) tradeoff parameter λ. The partial data are



(a) Precision vs. Noise Level (b) Recall vs. Noise Level (c) F-measure vs. Noise Level

FIGURE 3. The results on Water Pollution dataset.

TABLE 4. The statistics of datasets for the 2nd application.

Datasets	Statistics			
	Nodes	Edges	Blocks	Processors
SynNode	1,000∼10,000	3,000∼30,000	10	10
SynEdge	100,000	300,000∼1,000,000	100	25
Beijing	59,000	70,317	100	25
Wikivote	7,115	103,689	10	10
CondMat	23,133	93,497	100	25
DBLP	329,404	1,082,106	100	25

extracted as a training dataset, and a grid search is performed to determine those two parameters. λ is selected in a range with a fixed step size. The range of s varies in different datasets. These two parameters are set to values 150/0.001, 1000/0.001, 200/0.001, 2000/0.001 on the synthetic, water pollution, Washington D.C. and Beijing datasets, respectively. Additionally, it is observed that the best setting for s is approximately the half of the subgraph size (i.e., $s \approx \frac{1}{2}|S|$).

B. ANOMALOUS SUBGRAPH DETECTION IN NETWORK OF NETWORKS

1) SYNTHETIC DATASETS

As in the previous application, we use the Barabási-Albert model to generate multiple networks with different network sizes and then apply the random walk algorithm to select 10% of nodes as the ground-truth subgraphs. The attributes of nodes in true subgraphs conform to a normal distribution $\mathbb{N}(5, 1)$, while the attributes of the background nodes follow normal distribution $\mathbb{N}(0, 1)$. Two synthetic datasets are generated to analyze the scalability in terms of nodes and edges, which are denoted as SynNode and SynEdge.

2) REAL-WORLD DATASETS

1) *Beijing Road Traffic Dataset*: we use static network data per timestamp from 5PM to 7PM in the previous experiment. 2) *Wikivote Dataset*[2]: a dataset contains all administrator elections and vote history data, which is extracted from the Wikipedia page edit history untill January 3, 2008. 3) *CondMat Dataset*[3]: the collaboration network is from the e-print website arXiv and covers collaborations between researchers who submitted papers to condense matter category. For the Wikivote and CondMat datasets, true subgraphs with 1,000 nodes are generated by a random walk, and the node attributes in true subgraphs follow the distribution $\mathbb{N}(5, 1)$; otherwise, $\mathbb{N}(0, 1)$. 4) *DBLP Dataset*[4]: the collaboration graph of authors from DBLP computer science bibliography. An edge between two authors represents at least one collaboration, and the attribute of a node represents the number of publications published by the author. The dataset covers records ranging in time from 1995 to 2005. We apply random walk to obtain subgraphs with 20,000 nodes and inject anomalies as our true subgraph, as suggested by [28]. Table 4 gives the statistics of all datasets in this application. For the SynNode dataset, we test graphs with different sizes, which have nodes from 1,000 to 10,000 and the edges are 3 times of the nodes. In the SynEdge dataset, we keep nodes unchanged, and test different sizes of edges from 300,000 to 1,000,000. The "Blocks" columns tell us how many subnetworks in a network of networks. The "Processors" column gives the number of processors we used in our parallel algorithm. We can see that our algorithm can be scaled up to large networks with hundreds of thousands of nodes and over one million edges.

3) COMPARISON METHODS

We use three baselines to validate the performance of our framework: 1) AdditiveGraphScan [20], 2) EventTree [7], and 3) LinearTimeSubsetScan (LTSS) [19]. AdditiveGraphScan uses the expectation-based binomial (EBB) statistic to detect anomalous subsets in graphs automatically. EventTree reformulates the connected subgraph detection problem in attributed networks as a prize collection Steiner tree (PCST) problem and applies Goemans-Williamson algorithm to solve it. The LTSS method uses the "linear time subset scanning" property of a function (Kulldorff's spatial scan statistic and extensions) to scan subsets and detect events.

4) PERFORMANCE METRICS

We use the same metrics (precision, recall and F-measure) to evaluate the performance of all methods. Additionally, runtimes of different methods are reported here to validate the efficiency of our framework.

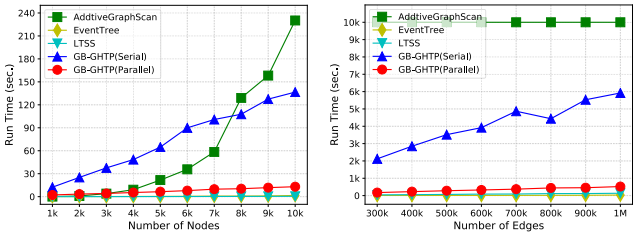
5) RESULTS

Experimental results on all real-world datasets are listed in Table 5. From the comparison of different methods, we can

[2]https://snap.stanford.edu/data/wiki- Vote.html
[3]https://snap.stanford.edu/data/ca- CondMat.html
[4]http://konect.uni-koblenz.de/networks/dblpcoauthor

TABLE 5. The results on the Beijing, Wikivote, CondMat and DBLP datasets. The runtime is measured in seconds.

Methods	Beijing				Wikivote				CondMat				DBLP			
	Precision	Recall	F-measure	Runtime	Precision	Recall	F-measure	Runtime	Precision	Recall	F-measure	Runtime	Precision	Recall	F-measure	Runtime
AddtivegGraphScan	0.4295	0.6884	0.5192	10846.94	0.9543	0.9959	0.9747	249.97	0.9753	0.9900	**0.9826**	1188.33	/	/	/	/
EventTree	0.5547	0.5577	0.5369	90.68	0.9088	0.9654	0.9360	80.99	0.8623	0.9204	0.8902	100.23	0.8213	0.1922	0.3113	1961.58
LTSS	0.5144	0.8333	0.6320	**7.56**	0.9543	0.9959	0.9747	**1.72**	0.5174	1.0000	0.6819	**3.85**	0.3910	1.0000	0.5622	**533.13**
GHTP(Serial)	0.9167	0.7285	**0.8056**	1812.78	0.9698	0.9863	**0.9780**	1552.57	0.9322	0.9832	0.9569	1569.26	0.4556	0.9673	0.6194	11203.23
GHTP(Parallel)	0.7903	0.8127	0.7963	436.42	0.9697	0.9858	0.9777	207.64	0.9506	0.9819	**0.9659**	156.61	0.4683	0.9673	**0.6310**	1291.10



(a) Runtime vs Nodes (b) Runtime vs Edges

FIGURE 4. Comparison of runtimes on synthetic datasets. Fig. (a) shows our framework runs in nearly linear time w.r.t to the network size, where $|\mathbb{E}| = 3|\mathbb{V}|$. Fig. (b) shows that our framework can be easily scaled up to 1, 000, 000 edges, where node size $|\mathbb{V}| = 100,000$; by contrast, AdditiveGraphScan runs over 10, 000 seconds on all cases.

see that our original (serial version) and parallelized methods both outperform all baselines in terms of F-measure except on the CondMat dataset. Although our original method is not efficient enough, it can scale to networks with hundreds of thousands of nodes and over one million edges as heuristic methods after parallelization. The reason why GB-GHTP cannot excel EventTree and LTSS in runtime is that our framework is an iterative method and requires more iterations. Despite more iterations to run, our framework provides a theoretic performance guarantee by compromising a small amount of runtime. The result of AdditiveGraphScan on the DBLP dataset is not reported since the method takes over one day to finish one run and thus is infeasible to tune its parameters.

6) SCALABILITY ANALYSIS

To analyze the scalability of different methods with respect to the number of nodes and edges, we evaluate these methods on synthetic datasets with different sizes. To run our method, we partition the static network into multiple blocks with METIS [51]. The runtimes of our framework compared with other baselines are reported in Fig. 4. In AdditiveGraphScan, a shortest path algorithm is used, which makes it not scalable to very large datasets. Since our method is an iterative algorithm, our serial method takes more time than some heuristic methods. However after it is parallelized, the runtime of our method can be reduced sharply (at least four times faster than the original serial version). Meanwhile, our framework can obtain comparable performance as those algorithms devised for specific applications. It is believed that our method could be more scalable if we utilize the computing resources rationally based on network properties.

7) PARAMETER TUNING

For all parameters used in AdditiveGraphScan, EventTree, and LTSS, we follow the same setting as [4], [19], [24]. To set parameters in our method, the same strategy as the previous application is used. The sparseness parameter in this experiment is set to be half of the size of a block. While we set the trade-off parameter to be 0.0005 in the SynNode, Wikivote and DBLP datasets, 0.001 in the SynEdge and CondMat datasets, and 0.0001 in the Beijing dataset. We run parallelized GB-GHTP on servers with multiple processors to speed up the algorithm, and more details are provided in Table 4, in which columns "Blocks" and "Processors" denote the number of subnetworks and processors used in our experiments.

C. CONNECTED DENSE SUBGRAPH DETECTION IN DUAL NETWORKS

1) SYNTHETIC DATASETS

We construct synthetic dual networks based on the Barabási-Albert model (shorthand for SynDual). Two networks with different densities are generated. In the synthetic dual networks, the edges to attach from a new node to existing nodes are 3 and 10 respectively. The subset of true vertices is selected with a biased random walk algorithm, which is applied on the first network. To simulate the dense area in the second network, the random walk algorithm considers the degree distribution of a node in the second network and accesses neighbor nodes that have a higher degree with higher probabilities. This biased strategy makes the generated subgraphs connected in the first network and dense in the second network. Then the univariate feature values of background nodes and true nodes are randomly generated in $\mathbb{N}(0, 1)$ and $\mathbb{N}(3, 1)$ distributions. The statistics about the generated dual networks can be seen in Table 6, which are averaged on 10 instances.

2) REAL-WORLD DATASETS

1) *Homo Dataset*: We consider different types of genetic interactions for organisms in the Biological General Respository for Interaction Datasets (BioGRID, thebiogrid.org), a public database that archives and disseminates genetic and protein interaction data from humans and model organisms [52]. The Homo dataset concerns Homo sapiens, and this multiplex network makes use of 7 layers[5]. We extracted 2 of those layers as our dual networks, which represent colocalization and direct interactions among genes of Homo sapiens.
2) *DBLP Datasets*: We construct two dual networks from the DBLP dataset [53], one for the data mining research community and one for the database reseach community. For the data mining community, papers published in 5 data mining conferences are included (KDD, ICDM, SDM, PKDD, and CIKM) to construct the dual networks. The dataset contains

[5]https://comunelab.fbk.eu/data.php

TABLE 6. The statistics of datasets for the 3rd application. Edges@# refers to the number of edges in the network #. Density@# refers to the density of network #.

Datasets	Statistics				
	Nodes	Edges@1	Edges@2	Density@1	Density@2
SynDual	1,000	2,991	9,900	0.00599	0.01982
Homo	3,886	13,424	19,044	0.00178	0.00252
DBLP-DM	4,102	10,229	30,000	0.00122	0.00357
DBLP-DB	4,402	13,306	30,000	0.00137	0.00310

4,102 authors and 7,194 papers and is denoted as DBLP-DM. The first network is the collaboration network, in which authors are the nodes and an edge represents that two authors have coauthored a paper. The second network is the research interest similarity network among authors, which is generated based on the similarity of the terms in the titles of their papers. We use the shrunk Pearson correlation coefficient to compute the research similarity between authors [46]. The dual networks for the database community are similarly constructed based on papers published in 3 database conferences: SIGMOD, VLDB, and ICDE. This dataset is denoted as DBLP-DB, in which 4,402 authors and 6,087 papers are included. Note that for both datasets, only the top 30,000 positive correlations are introduced into the second network as edges. The aforementioned prepossessing for the DBLP dataset is the same as in reference [46]. The publications for an author are counted and used as univariate features of the author. Table 6 summarizes the statistics of all datasets.

3) COMPARISON METHODS

Two baselines are compared with our method: 1) DCS [46] and 2) EventAllPair+ [7]. DCS is designed for finding the densest connected subgraph in dual networks. However this method does not consider attributes on nodes at all. EventAllPairs+ algorithm finds a subset of vertices that have large total weights and are sufficiently compact. It considers attributes but only handles a single network and cannot guarantee connected and dense constraints.

4) RESULTS

We report performance of different methods on the synthetic and Homo datasets in Table 7. As can be seen, since our method considers the attributes on nodes and constraints imposed on dual networks, it has much better results on all metrics than other methods. Although the DCS can detect subgraphs that are denser than ours, it only reflects structural information while attributes on nodes are sometimes more important and help find more meaningful patterns.

5) CASE STUDIES

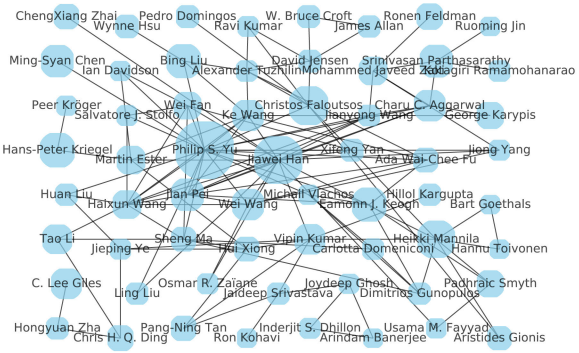
In addition to measuring the performance of various methods, we are also interested in the ability of our framework to infer meaningful patterns. Thus, we analyze some subgraphs detected by our method and DCS baseline on two real datasets DBLP-DM and DBLP-DB. The subgraphs detected by our method in the DBLP-DM dataset are shown in Fig. 5. As you can see, in the collaboration network on the left, the detected subgraph is connected. Most importantly, our method can find some coauthor relationships between some influential researchers in data mining. When drawing the figures, we use a circle to represent an author and the radius of a circle is decided by the number of publications of the associated author. Then we can construct a collaboration network among those most influential researchers in the data mining field via our method. As shown in Fig. 5a, Jiawei Han and Phillip S. Yu are the two most influential researchers in the data mining community and they both published many papers and had a large number of collaborations with other researchers. It can also be seen in Fig. 5b, the more influential a researcher is, the more research interest similarity they share with other researchers, which is reflected by the phenomenon that a node with larger radius has denser connections with other nodes in the network.

The subgraphs detected by our method on DBLP-DB are not drawn since there are many more nodes in the subgraphs and are difficult to visualize. Instead we list some statistics about our found subgraphs in Table 8 and describe some interesting results.

In the DBLP-DM dataset, although the density of the subgraph in the research interest network is higher than ours, the subgraph detected by the DCS method includes many authors whose publications are fewer than 5. From the publication distributions of authors in the detected subgraphs, we can see that all of the authors (Fig. 6a) discovered by our method published more than 10 papers in 5 data mining conferences, while more than 50% of the authors (Fig. 6c) discovered by the DCS method published fewer than 5 papers and nearly 30% of the authors published only 1 paper. The average publications of a researcher in the subgraph by DCS are 8.13, while ours are 25.03. Our method finds collaborations between more influential researchers. Obviously, it is more likely to track a hot research topic from more influential researchers rather than those who only published 1 paper. We observe a similar phenomenon from the results on the DBLP-DB dataset by comparing Fig. 6b with Fig. 6d. The average number of publications of researchers in the

TABLE 7. The results on the Synthetic and Homo datasets. The column "Density@2" refers to the density of the detected subgraph induced from the second network.

Methods	Synthetic				Homo			
	Precision	Recall	F-measure	Density	Precision	Recall	F-measure	Density@2
DCS	0.4620	0.4620	0.4620	0.18212	0.5682	0.3000	0.3927	0.08345
EventAllPair+@1	0.2715	0.7305	0.3955	0.03396	0.3066	0.7071	0.4275	0.00383
EventAllPair+@2	0.2794	0.7295	0.4037	0.03540	0.3226	0.7304	0.4473	0.00661
GB-GHTP	0.8395	0.8680	**0.8528**	0.08627	0.8795	0.8592	**0.8692**	0.01884



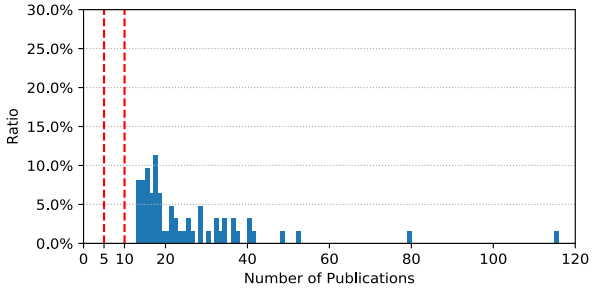
(a) The subgraph extracted from the collaboration network.

(b) The subgraph extracted from the research interest network.

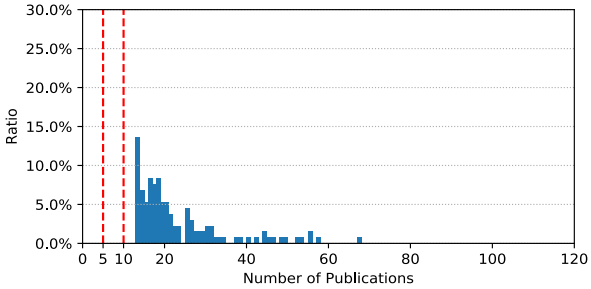
FIGURE 5. The subgraphs detected by our method on the DBLP-DM dataset.

TABLE 8. Some statistics of results by DCS and GB-GHTP. Edges@# refers to number of edges in the # graph. The column "Avg Publications" refers to average number of publications of authors in a detected subgraph.

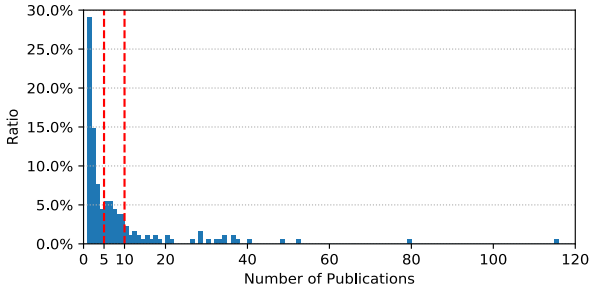
Datasets	Methods	Nodes	Edges@1	Edges@2	Density@2	Avg Publications
DBLP-DM	DCS	182	296	6,599	0.40064	8.13
	GB-GHTP	62	131	2,461	0.24379	25.03
DBLP-DB	DCS	176	339	3,817	0.24786	10.63
	GB-GHTP	132	492	1,071	0.12387	22.71



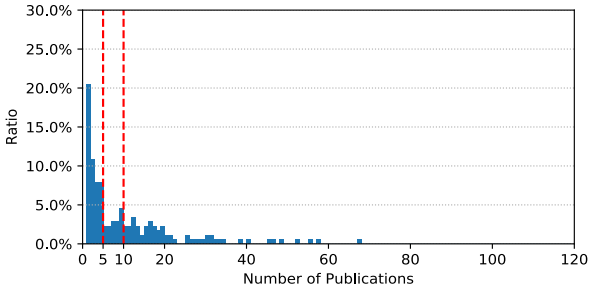
(a) The publication distribution on DBLP-DM.



(b) The publication distribution on DBLP-DB.



(c) The publication distribution on DBLP-DM.



(d) The publication distribution on DBLP-DB.

FIGURE 6. The publication distribution on the DBLP-DM and DBLP-DB datasets. The figures in the top row describe the results by GB-GHTP. The bottom figures describe the results by DCS.

subgraph detected by our method is 22.71. The average number of publications of researchers in the subgraph by DCS is 10.63. These two cases prove that it is not enough to detect subgraphs that are densest and connected and it matters more for us to consider attributes on nodes, rule out those nodes that are not that important, and further detect some more significant patterns.

The results of EventAllPairs+ are not listed because the algorithm cannot guarantee connectivity in the collaboration network and are unexplainable for their implications.

6) METRICS AND PARAMETER TUNING

We use the same strategies as the previous two applications to evaluate the performance of the methods and decide the parameters in our method. The parameters used in DCS are recommended by the authors of the original paper ($\gamma = 1.5$). The parameter λ used in EventAllPair+ is selected from the settings on the training dataset, which evaluates 1,501 and 901 on the SynDual and Homo datasets, respectively As mentioned before, the parameters on DBLP-DM and DBLP-DB are not listed because the results of the

experiments are unexplainable. The sparseness parameter s and tradeoff parameter λ in GB-GHTP are set to be 50/1.0, 250/1.0, 300/10.0 and 300/10.0 on the SynDual, Homo, DBLP-DM and DBLP-DB datasets.

Implementation All experiments were conducted on 64-bit machines with Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40 GHz and 251 GB memory.

VII. CONCLUSIONS AND FUTURE WORK

This paper presents a graph block-structured optimization based framework, to detect subgraphs in attributed interdependent networks, which runs in nearly linear time with respect to the network size and provides a theoretical guarantee. A parallel version of the framework is proposed to improve its scalability. We evaluate our framework on three applications. The results on both synthetic and real-world datasets indicate that our framework enjoys better effectiveness and efficiency than other baselines. Additionally, our framework is not designed for a specific problem and can be applied to more scenarios. For future work, we will deploy our framework to more applications and networks with multiple attributes. It is also worth exploring more powerful objective functions to capture interesting patterns on attributed networks.

APPENDIX A LEMMAS

In the appendix, we first present two necessary lemmas and then give the proof of Theorem 1 and Theorem 2.

Lemma 1: Assume that function $F(\cdot)$ is differentiable, if $F(\cdot)$ satisfies the $(\xi, \delta, \mathbb{M})-$WRSC condition, then for any $x, y \in \mathbb{R}^N$ with $\text{supp}(x) \cup \text{supp}(y) \subseteq S \in \mathbb{M}$, the following inequalities hold [28]

$$\frac{1-\delta}{\xi}\|x-y\|_2 \leq \|\nabla_S F(x) - \nabla_S F(y)\|_2 \leq \frac{1+\delta}{\xi}\|x-y\|_2,$$

$$F(x) \leq F(y) + \langle \nabla F(x), x-y\rangle + \frac{1+\delta}{2\xi}\|x-y\|_2^2.$$

Lemma 2: Assume that $\alpha_0 = c_{\mathrm{H}}(1-\delta) - \delta$, $\beta_0 = \xi(1+c_{\mathrm{H}})$, $r^i = x^i - x$, and $\Omega^k = \mathrm{H}(\nabla_{x^k} F(x^i))$. Then

$$\|r^i_{\Omega^c}\|_2 \leq \sqrt{1-\alpha_0^2}\|r^i\|_2 + \left(\frac{\beta_0}{\alpha_0} + \frac{\alpha_0\beta_0}{\sqrt{1-\alpha_0^2}}\right)\|(\nabla F(x))_{\mathrm{I}}\|_2,$$

where $\mathrm{I} = \underset{S\in\mathbb{M}(\mathbb{G},8s)}{\text{argmax}} \|(\nabla F(x))_S\|_2$. We assume that c_{H} and δ are constants such that $\alpha_0 > 0$.

Proof: Assume that $\Phi = (\Phi^1, \ldots, \Phi^k) = \text{supp}(x), x^k \in \mathbb{M}(\mathbb{G}, s_k), k = 1, \ldots, K$, and $\Gamma = \text{supp}(r^i) \in \mathbb{M}(\mathbb{G}, 6s)$. We first bound the term $\|(\nabla F(x^i))_\Omega\|_2$

$$\begin{aligned}
\|(\nabla F(x^i))_\Omega\|_2 &\geq \sqrt{\sum_{k=1}^{K} c_{\mathrm{H}_k}^2 \|(\nabla_{x^k} F(x^i))_{\Omega^{k*}}\|_2^2} \\
&\geq c_{\mathrm{H}} \sqrt{\sum_{k=1}^{K} \|(\nabla_{x^k} F(x^i))_{\Phi^k}\|_2^2} = c_{\mathrm{H}}\|(\nabla F(x^i))_\Phi\|_2 \\
&\geq c_{\mathrm{H}}\|(\nabla F(x^i))_\Phi - (\nabla F(x))_\Phi\|_2 - c_{\mathrm{H}}\|(\nabla F(x))_\Phi\|_2 \\
&\geq \frac{c_{\mathrm{H}}(1-\delta)}{\xi}\|r^i\|_2 - c_{\mathrm{H}}\|(\nabla F(x))_{\mathrm{I}}\|_2
\end{aligned}$$

in which $c_{\mathrm{H}} = \min_{k=1,\ldots,K} c_{\mathrm{H}_k}$. The first "$\geq$" follows from the head approximation, and the last "$\geq$" follows from Lemma 1. The term $\|(\nabla F(x^i))_\Omega\|_2$ can also be upper bounded by

$$\begin{aligned}
&\|(\nabla F(x^i))_\Omega\|_2 \\
&\quad\leq \frac{1}{\xi}\|\xi(\nabla F(x^i))_\Omega - \xi(\nabla F(x))_\Omega - r^i_\Omega + r^i_\Omega\|_2 + \|(\nabla F(x))_\Omega\|_2 \\
&\quad\leq \frac{1}{\xi}\|\xi(\nabla F(x^i))_{\Gamma\cup\Omega} - \xi(\nabla F(x))_{\Gamma\cup\Omega} - r^i_{\Gamma\cup\Omega}\|_2 + \frac{1}{\xi}\|r^i_\Omega\|_2 \\
&\qquad + \|(\nabla F(x))_\Omega\|_2 \leq \frac{\delta}{\xi}\|r^i\|_2 + \frac{1}{\xi}\|r^i_\Omega\|_2 + \|(\nabla F(x))_{\mathrm{I}}\|_2
\end{aligned}$$

where the last inequality follows the WRSC condition. Combining the two bounds yields the inequality:

$$\|r^i_\Omega\|_2 \geq (c_{\mathrm{H}}(1-\delta) - \delta)\|r^i\|_2 - \xi(1+c_{\mathrm{H}})\|(\nabla F(x))_{\mathrm{I}}\|_2 \quad (22)$$

where $\alpha_0 = c_{\mathrm{H}}(1-\delta) - \delta, \beta_0 = \xi(1+c_{\mathrm{H}})$. Assume $0 < \alpha_0 < 1$. To derive an upper bound of $\|r^i_{\Omega^c}\|_2$, two cases can be considered.

- Case 1: If the right-hand side of (22) ≤ 0, i.e., $\alpha_0\|r^i\|_2 \leq \beta_0\|(\nabla F(x))_{\mathrm{I}}\|_2$, we have
$$\|r^i_{\Omega^c}\|_2 \leq \|r^i\|_2 \leq \frac{\beta_0}{\alpha_0}\|(\nabla F(x))_{\mathrm{I}}\|_2,$$
- Case 2: If the right-hand side of (22) > 0, i.e., $\alpha_0\|r^i\|_2 > \beta_0\|(\nabla F(x))_{\mathrm{I}}\|_2$, we have
$$\|r^i_\Omega\|_2 \geq \left(\alpha_0 - \frac{\beta_0\|(\nabla F(x))_{\mathrm{I}}\|_2}{\|r^i\|_2}\right)\|r^i\|_2,$$

Moreover, since $\|r^i\|_2^2 = \|r^i_\Omega\|_2^2 + \|r^i_{\Omega^c}\|_2^2$, we have

$$\begin{aligned}
\|r^i_{\Omega^c}\|_2^2 &= \|r^i\|_2^2 - \|r^i_\Omega\|_2^2, \\
\|r^i_{\Omega^c}\| &\leq \|r^i\|_2\sqrt{1 - \left(\alpha_0 - \frac{\beta_0\|(\nabla F(x))_{\mathrm{I}}\|_2}{\|r^i\|_2}\right)^2}.
\end{aligned}$$

Denote $\omega_0 = \alpha_0 - \frac{\beta_0\|(\nabla F(x))_{\mathrm{I}}\|_2}{\|r^i\|_2}$. For a given $0 < \omega_0 < 1$ and a free parameter $0 < \omega < 1$, a straightforward calculation yields that $\sqrt{1-\omega_0^2} \leq \frac{1}{\sqrt{1-\omega^2}} - \frac{\omega}{\sqrt{1-\omega^2}}\omega_0$.

$$\sqrt{1-\omega_0^2} \leq \frac{1}{\sqrt{1-\omega^2}} - \frac{\omega}{\sqrt{1-\omega^2}}\omega_0 \Leftrightarrow \omega^2 + \omega_0^2 - 2\omega\omega_0 \geq 0$$

Therefore, substituting into the bound for $\|r^i_{\Omega^c}\|_2$, we obtain

$$\begin{aligned}
\|r^i_{\Omega^c}\|_2 &\leq \|r^i\|_2\left(\frac{1}{\sqrt{1-\omega^2}} - \frac{\omega}{\sqrt{1-\omega^2}}\left(\alpha_0 - \frac{\beta_0\|(\nabla F(x))_{\mathrm{I}}\|_2}{\|r^i\|_2}\right)\right) \\
&= \frac{1-\omega\alpha_0}{\sqrt{1-\omega^2}}\|r^i\|_2 + \frac{\omega\beta_0}{\sqrt{1-\omega^2}}\|(\nabla F(x))_{\mathrm{I}}\|_2
\end{aligned}$$

The coefficient preceding $\|r^i\|_2$ determines the convergence rate of our framework, and we can minimize the value of the coefficient by setting $\omega = \alpha_0$.

Therefore, combining the two cases yields the desired results and proves the lemma. □

APPENDIX B PROOF OF THEOREM 1

Proof: Denote $\Omega^k = \mathrm{H}(\nabla_{x^k} F(x^i))$, $\Gamma^k = \mathrm{supp}(x^{k,i} - \eta \cdot (\nabla_{x^k} F(x^i))_{\Omega^k})$, and $r^{i+1} = x^{i+1} - x$; then, the term r^{i+1} can be bounded as

$$\begin{aligned}\|r^{i+1}\|_2 &= \|x^{i+1} - x\|_2 \le \|x^{i+1} - b^i\|_2 + \|x - b^i\|_2 \\ &= \sqrt{\|x^{1,i+1} - b^{1,i}\|_2^2 + \cdots + \|x^{K,i+1} - b^{K,i}\|_2} \\ &\quad + \|b^i - x\|_2 \\ &\le \sqrt{c_{\mathrm{T}_1}^2 \|(b^{1,i})^* - b^{1,i}\|_2^2 + \cdots + c_{\mathrm{T}_1}^2 \|(b^{K,i})^* - b^{K,i}\|_2^2} \\ &\quad + \|b^i - x\|_2 \\ &\le \sqrt{c_{\mathrm{T}}^2 \|(b^{1,i})^* - b^{1,i}\|_2^2 + \cdots + c_{\mathrm{T}}^2 \|(b^{K,i})^* - b^{K,i}\|_2^2} \\ &\quad + \|b^i - x\|_2 \\ &= (1 + c_{\mathrm{T}})\|b^i - x\|_2 \end{aligned} \tag{23}$$

where the first "$\le$" follows the tail projection and in the second inequality $c_{\mathrm{T}} = \max_{k=1,\ldots,K} c_{\mathrm{T}_k}$. The term $\|(x - b^i)_\Gamma\|_2^2$ is bounded as

$$\begin{aligned}\|(x - b^i)_\Gamma\|_2^2 &= \left\langle b^i - x, (b^i - x)_\Gamma \right\rangle \\ &= \left\langle b^i - x - \xi(\nabla F(b^i))_\Gamma + \xi(\nabla F(x))_\Gamma, (b^i - x)_\Gamma \right\rangle \\ &\quad - \left\langle \xi(\nabla F(x))_\Gamma, (b^i - x)_\Gamma \right\rangle \\ &\le \delta\|b^i - x\|_2 \|(b^i - x)_\Gamma\|_2 + \xi\|(\nabla F(x))_\Gamma\|_2 \|(b^i - x)_\Gamma\|_2, \end{aligned}$$

where the second "$=$" makes sense because $(\nabla F(b^i))_S = 0$, which results from b being the solution to the problem in (10) (Line 6) of GB-GHTP, and the last inequality can be obtained from the WRSC condition. Then, we obtain the inequality

$$\|(x - b^i)_\Gamma\|_2 \le \delta\|x - b^i\|_2 + \xi\|(\nabla F(x))_\Gamma\|_2$$

which further gives the bound

$$\begin{aligned}\|x - b^i\|_2 &= \|(x - b^i)_\Gamma\|_2 + \|(x - b^i)_{\Gamma^c}\|_2 \\ &\le \delta\|x - b^i\|_2 + \xi\|(\nabla F(x))_\Gamma\|_2 + \|(x - b^i)_{\Gamma^c}\|_2 \end{aligned}$$

We obtain the following inequality after rearrangement

$$\|x - b^i\|_2 \le \frac{\|(x - b^i)_{\Gamma^c}\|_2}{1 - \delta} + \frac{\xi\|(\nabla F(x))_\Gamma\|_2}{1 - \delta} \tag{24}$$

Let $\Phi = \mathrm{supp}(x) \in \mathbb{M}(\mathbb{G}, s)$,

$$\|(x^i - \eta(\nabla F(x^i))_\Omega)_\Phi\|_2 \le \|(x^i - \eta(\nabla F(x^i))_\Omega)_\Gamma\|_2$$

as $\Gamma = \mathrm{supp}(x^i - \eta(\nabla F(x^i))_\Omega)$. By eliminating the contribution on $\Phi \cap \Gamma$, we derive

$$\|(x^i - \eta(\nabla F(x^i))_\Omega)_{\Phi\setminus\Gamma}\|_2 \le \|(x^i - \eta(\nabla F(x^i))_\Omega)_{\Gamma\setminus\Phi}\|_2$$

We have the following inequality from the right-hand side

$$\begin{aligned}&\|(x^i - \eta(\nabla F(x^i))_\Omega)_{\Gamma\setminus\Phi}\|_2 \\ &\quad \le \eta\|(\nabla F(x))_{\Omega\cup\Gamma}\|_2 \\ &\qquad + \|(x^i - x - \eta(\nabla F(x^i))_\Omega + \eta(\nabla F(x))_\Omega)_{\Gamma\setminus\Phi}\|_2 \end{aligned} \tag{25}$$

which is derived from the fact that $\Phi = \mathrm{supp}(x)$. For the left-hand side, we have

$$\begin{aligned}&\|(x^i - \eta(\nabla F(x^i))_\Omega)_{\Phi\setminus\Gamma}\|_2 \\ &\quad \ge -\eta\|(\nabla F(x))_{\Omega\cup\Phi}\|_2 \\ &\qquad - \|(x^i - x - \eta(\nabla F(x^i))_\Omega + \eta(\nabla F(x))_\Omega)_{\Phi\setminus\Gamma}\|_2 \\ &\qquad + \|(b^i - x)_{\Gamma^c}\|_2 \end{aligned} \tag{26}$$

where the "$\ge$" follows from the fact that $b^i_{\Gamma^c} = 0$, $x_{\Phi\setminus\Gamma} = x_{\Gamma^c}$, and $-x_{\Phi\setminus\Gamma} + (x - b^i)_{\Gamma^c} = 0$. Assume $\Phi\Delta\Gamma$ is the symmetric difference of the set Φ and Γ. Combining (25) and (26), we obtain

$$\begin{aligned}\|(b^i - x)_{\Gamma^c}\|_2 &\le \sqrt{2}\|(x^i - x - \eta(\nabla F(x^i))_\Omega + \eta(\nabla F(x))_\Omega)_{\Phi\Delta\Gamma}\|_2 \\ &\quad + 2\eta\|(\nabla F(x))_{\mathrm{I}}\|_2 \end{aligned} \tag{27}$$

which follows that

$$\begin{aligned}&\|(b^i - x)_{\Gamma^c}\|_2 \\ &\le \sqrt{2}\|(x^i - x - \eta(\nabla F(x^i))_\Omega + \eta(\nabla F(x))_\Omega)_{\Phi\Delta\Gamma}\|_2 \\ &\quad + 2\eta\|(\nabla F(x))_{\mathrm{I}}\|_2 \\ &\le \sqrt{2}\|(x^i - x - \xi(\nabla F(x^i))_\Omega + \xi(\nabla F(x))_\Omega)_{\Phi\Delta\Gamma}\|_2 \\ &\quad + \sqrt{2}(\xi - \eta)\|((\nabla F(x^i))_\Omega - (\nabla F(x))_\Omega)_{\Phi\Delta\Gamma}\|_2 \\ &\quad + 2\eta\|(\nabla F(x))_{\mathrm{I}}\|_2 \\ &= \sqrt{2}\|(r^i_{\Omega^c} + r^i_\Omega - \xi(\nabla F(x^i))_\Omega + \xi(\nabla F(x))_\Omega)_{\Phi\Delta\Gamma}\|_2 \\ &\quad + \sqrt{2}(\xi - \eta)\|((\nabla F(x^i))_\Omega - (\nabla F(x))_\Omega)_{\Phi\Delta\Gamma}\|_2 \\ &\quad + 2\eta\|(\nabla F(x))_{\mathrm{I}}\|_2 \\ &\le \sqrt{2}\|r^i_{\Omega^c}\|_2 + \sqrt{2}\|(r^i_\Omega - \xi(\nabla F(x^i))_\Omega + \xi(\nabla F(x))_\Omega)_{\Phi\Delta\Gamma}\|_2 \\ &\quad + \sqrt{2}(\xi - \eta)\|((\nabla F(x^i))_\Omega - (\nabla F(x))_\Omega)_{\Phi\Delta\Gamma}\|_2 \\ &\quad + 2\eta\|(\nabla F(x))_{\mathrm{I}}\|_2 \\ &\le \sqrt{2}\|r^i_{\Omega^c}\|_2 + \sqrt{2}\|r^i - \xi(\nabla F(x^i))_{\Omega\cup\Gamma\cup\Phi} + \xi(\nabla F(x))_{\Omega\cup\Gamma\cup\Phi}\|_2 \\ &\quad + \sqrt{2}(\xi - \eta)\|(\nabla F(x^i))_{\Omega\cup\Gamma\cup\Phi} - (\nabla F(x))_{\Omega\cup\Gamma\cup\Phi}\|_2 \\ &\quad + 2\eta\|(\nabla F(x))_{\mathrm{I}}\|_2 \\ &\le \sqrt{2}\|r^i_{\Omega^c}\|_2 + \sqrt{2}\left(1 - \frac{\eta}{\xi} + \left(2 - \frac{\eta}{\xi}\right)\delta\right)\|r^i\|_2 \\ &\quad + 2\eta\|(\nabla F(x))_{\mathrm{I}}\|_2 \end{aligned}$$

where the first "$\le$" follows from the inequality 27. The third "$\le$" follows from the fact that $\|(r^i_{\Omega^c})_{\Phi\Delta\Gamma}\|_2 \le \|r^i_{\Omega^c}\|_2$. The fourth "$\le$" follows from the fact that $\mathrm{supp}(r^i) \subseteq \Omega\cup\Gamma\cup\Phi$. The last "$\le$" follows from the WRSC condition and Lemma 1. Combining (23), (24), (28), and Lemma 2, the theorem is established. □

APPENDIX C PROOF OF THEOREM 2

Proof: The following inequality holds in the i^{th} iteration

$$\|x^i - x\|_2 \le \alpha^i\|x\|_2 + \frac{\beta}{1 - \alpha}\|(\nabla F(x))_{\mathrm{I}}\|_2 \tag{28}$$

After $t = \left\lceil \log\left(\frac{\|x\|_2}{\|(\nabla F(x))_{\mathrm{I}}\|_2}\right) / \log\frac{1}{\alpha} \right\rceil$ iterations, GB-GHTP returns an estimate $\hat{x}$ that satisfies $\|\hat{x} - x\|_2 \le \left(1 + \frac{\beta}{1-\alpha}\right)$ $|(\nabla F(x))_{\mathrm{I}}\|_2$. The time complexities of both head and tail approximations are $O\left(|\mathbb{E}|\log^3 N\right)$. Then the time complexity of one iteration in GB-GHTP is $\left(T + |\mathbb{E}|\log^3 N\right)$. Since

the total number of iterations is $\left\lceil \log\left(\frac{\|x\|_2}{\|(\nabla F(x))_{\mathrm{I}}\|_2}\right) / \log\frac{1}{\alpha}\right\rceil$, the overall time follows Theorem 2. □

REFERENCES

[1] B. Miller, N. Bliss, and P. J. Wolfe, "Subgraph detection using eigenvector L1 norms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1633–1641.

[2] W. Gao, H. Wu, M. K. Siddiqui, and A. Q. Baig, "Study of biological networks using graph theory," *Saudi J. Biol. Sci.*, vol. 25, no. 6, pp. 1212–1219, Sep. 2018.

[3] W. Gao, W. Wang, D. Dimitrov, and Y. Wang, "Nano properties analysis via fourth multiplicative ABC indicator calculating," *Arabian J. Chem.*, vol. 11, no. 6, pp. 793–801, Sep. 2018.

[4] M. Shao, J. Li, F. Chen, and X. Chen, "An efficient framework for detecting evolving anomalous subgraphs in dynamic networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 2258–2266.

[5] N. Wu, W. Wang, F. Chen, J. Li, B. Li, and J. Huai, "Uncovering specific-shape graph anomalies in attributed graphs," in *Proc. 33rd AAAI Conf. Artif. Intell. (AAAI)*, 2019, pp. 5433–5440.

[6] F. Chen and D. B. Neill, "Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1166–1175.

[7] P. Rozenshtein, A. Anagnostopoulos, A. Gionis, and N. Tatti, "Event detection in activity networks," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1176–1185.

[8] B. Zhou, F. Chen, and Y. Ying, "Stochastic iterative hard thresholding for graph-structured sparsity optimization," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 7563–7573.

[9] M. Shao, J. Li, F. Chen, H. Huang, S. Zhang, and X. Chen, "An efficient approach to event detection and forecasting in dynamic multivariate social media networks," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, Apr. 2017, pp. 1631–1639.

[10] N. Wu, F. Chen, J. Li, J. Huai, B. Zhou, B. Li, and N. Ramakrishnan, "A nonparametric approach to uncovering connected anomalies by tree shaped priors," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1849–1862, Oct. 2019.

[11] T. He and K. C. C. Chan, "MISAGA: An algorithm for mining interesting subgraphs in attributed graphs," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1369–1382, May 2018.

[12] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, "Networks formed from interdependent networks," *Nature Phys.*, vol. 8, no. 1, p. 40, 2012.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[14] M. Mongiovì, P. Bogdanov, R. Ranca, E. E. Papalexakis, C. Faloutsos, and A. K. Singh, "NetSpot: Spotting significant anomalous regions on dynamic networks," in *Proc. SIAM Int. Conf. Data Mining*, May 2013, pp. 28–36.

[15] H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, and T. Ideker, "Network-based classification of breast cancer metastasis," *Mol. Syst. Biol.*, vol. 3, no. 1, p. 140, Oct. 2007.

[16] D. P. de Oliveira, D. B. Neill, J. H. Garrett, and L. Soibelman, "Detection of patterns in water distribution pipe breakage using spatial scan statistics for point events in a physical network," *J. Comput. Civil Eng.*, vol. 25, no. 1, pp. 21–30, Jan. 2011.

[17] J. Qian, V. Saligrama, and Y. Chen, "Connected sub-graph detection," in *Proc. 17th Int. Conf. Artif. Intell. Statist.*, in Proceedings of Machine Learning Research, vol. 33, 2014, pp. 796–804.

[18] D. B. Neill and A. W. Moore, "Anomalous spatial cluster detection," in *Proc. Workshop Data Mining Methods Anomaly Detection (KDD)*, 2005. [Online]. Available: https://cs.nyu.edu/~neill/papers/ADKDD-Neill.pdf

[19] D. B. Neill, "Fast subset scan for spatial pattern detection," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 74, no. 2, pp. 337–360, Mar. 2012.

[20] S. Speakman, Y. Zhang, and D. B. Neill, "Dynamic pattern detection with temporal consistency and connectivity constraints," in *Proc. IEEE 13th Int. Conf. Data Mining (ICDM)*, Dec. 2013, pp. 697–706.

[21] P. Bogdanov, M. Mongiovì, and A. K. Singh, "Mining heavy subgraphs in time-evolving networks," in *Proc. IEEE 11th Int. Conf. Data Mining (ICDM)*, Dec. 2011, pp. 81–90.

[22] C. Hegde, P. Indyk, and L. Schmidt, "A nearly-linear time framework for graph-structured sparsity," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 928–937.

[23] C. Hegde, P. Indyk, and L. Schmidt, "Approximation-tolerant model-based compressive sensing," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, Jan. 2014, pp. 1544–1561.

[24] B. Zhou and F. Chen, "Graph-structured sparse optimization for connected subgraph detection," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 709–718.

[25] C. Hegde, P. Indyk, and L. Schmidt, "Approximation algorithms for model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 5129–5147, Sep. 2015.

[26] F. Chen and B. Zhou, "A generalized matching pursuit approach for graph-structured sparsity," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 1389–1395.

[27] F. Chen, B. Zhou, A. Alim, and L. Zhao, "A generic framework for interesting subspace cluster detection in multi-attributed networks," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 41–50.

[28] J. Chen and Q. Gu, "Fast Newton hard thresholding pursuit for sparsity constrained nonconvex optimization," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 127–135.

[29] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *J. Complex Netw.*, vol. 2, no. 3, pp. 203–271, 2014.

[30] J. Gao, S. V. Buldyrev, H. E. Stanley, X. Xu, and S. Havlin, "Percolation of a general network of networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 88, no. 6, Dec. 2013, Art. no. 062816.

[31] A. Bashan, Y. Berezin, S. V. Buldyrev, and S. Havlin, "The extreme vulnerability of interdependent spatially embedded networks," *Nature Phys.*, vol. 9, no. 10, p. 667, 2013.

[32] A. Saumell-Mendiola, M. Á. Serrano, and M. Boguñá, "Epidemic spreading on interconnected networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 86, no. 2, Aug. 2012, Art. no. 026106.

[33] H. Wang, Q. Li, G. D'Agostino, S. Havlin, H. E. Stanley, and P. Van Mieghem, "Effect of the interconnected network structure on the epidemic threshold," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 88, no. 2, Aug. 2013, Art. no. 022801.

[34] W. Gao, D. Dimitrov, and H. Abdo, "Tight independent set neighborhood union condition for fractional critical deleted graphs and ID deleted graphs," *Discrete Continuous Dyn. Syst.-S*, vol. 12, nos. 4–5, p. 711, 2018.

[35] W. Gao, J. L. G. Guirao, M. Abdel-Aty, and W. Xi, "An independent set degree condition for fractional critical deleted graphs," *Discrete Continuous Dyn. Syst.-S*, vol. 12, nos. 4–5, p. 877, 2018.

[36] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *Science*, vol. 328, no. 5980, pp. 876–878, May 2010.

[37] W. Gao, J. L. G. Guirao, B. Basavanagoud, and J. Wu, "Partial multi-dividing ontology learning algorithm," *Inf. Sci.*, vol. 467, pp. 35–58, Oct. 2018.

[38] P. Tseng and S. Yun, "A coordinate gradient descent method for non-smooth separable minimization," *Math. Program.*, vol. 117, nos. 1–2, pp. 387–423, Mar. 2009.

[39] H.-J. Michael Shi, S. Tu, Y. Xu, and W. Yin, "A primer on coordinate descent algorithms," 2016, *arXiv:1610.00040*. [Online]. Available: http://arxiv.org/abs/1610.00040

[40] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2037–2060, Jan. 2013.

[41] N. Zhou, Y. Xu, H. Cheng, J. Fang, and W. Pedrycz, "Global and local structure preserving sparse subspace learning: An iterative approach to unsupervised feature selection," *Pattern Recognit.*, vol. 53, pp. 87–101, May 2016.

[42] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo, "Iteration complexity analysis of block coordinate descent methods," *Math. Program.*, vol. 163, nos. 1–2, pp. 85–114, May 2017.

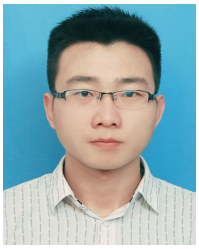
[43] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.

[44] Y. Xu, "Alternating proximal gradient method for sparse nonnegative tucker decomposition," *Math. Program. Comput.*, vol. 7, no. 1, pp. 39–70, Mar. 2015.

[45] E. Arias-Castro, E. J. Candès, and A. Durand, "Detection of an anomalous cluster in a network," *Ann. Statist.*, vol. 39, no. 1, pp. 278–304, Feb. 2011.

[46] Y. Wu, R. Jin, X. Zhu, and X. Zhang, "Finding dense and connected subgraphs in dual networks," in *Proc. IEEE 31st Int. Conf. Data Eng.*, Apr. 2015, pp. 915–926.

[47] O. Fercoq and P. Richtárik, "Accelerated, parallel, and proximal coordinate descent," *SIAM J. Optim.*, vol. 25, no. 4, pp. 1997–2023, Jan. 2015.
[48] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.
[49] A. Ostfeld *et al.*, "The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms," *J. Water Resour. Planning Manage.*, vol. 134, no. 6, pp. 556–568, Nov. 2008.
[50] F. Chen, C. Wang, and J.-H. Cho, "Collective subjective logic: Scalable uncertainty-based opinion inference," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 7–16.
[51] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, Jan. 1998.
[52] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature Commun.*, vol. 6, no. 1, p. 6864, Nov. 2015.
[53] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 990–998.



FEI JIE received the bachelor's degree in information security from the Hefei University of Technology, Hefei, China, in 2014, where he is currently pursuing the Ph.D. degree. His research interests include data mining, machine learning, and social media analytics.

CHUNPAI WANG received the bachelor's degree in computer science and statistics from the University of Rochester, Rochester, NY, USA, in 2014. He is currently pursuing the Ph.D. degree with the University at Albany - SUNY, Albany, NY. His research interests include machine learning, data mining, and recommender systems.

FENG CHEN (Member, IEEE) received the Ph.D. degree in computer science from Virginia Tech, Blacksburg, VA, USA, in 2012. He is currently an Associate Professor of computer science with The University of Texas at Dallas, Richardson, TX, USA. His research interests include anomalous pattern detection, event detection and forecasting, graph mining, and machine learning. His research has been supported by NSF, NIH, ARO, IARPA, and the U.S. Department of Transportation.

LEI LI (Senior Member, IEEE) received the bachelor's degree in information and computational science from Jilin University, Changchun, China, in 2004, the master's degree in applied mathematics from the Memorial University of Newfoundland, St. John's, Canada, in 2006, and the Ph.D. degree in computing from Macquarie University, Sydney, Australia, in 2012. He is currently an Associate Professor with the Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology) and the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. He has published over 70 peer-reviewed papers in prestigious journals and top international conferences including the IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON SERVICES COMPUTING, *Knowledge-Based Systems, World Wide Web Journal, Pattern Recognition Letters*, AAAI, IJCAI, ICDM, ICSOC and IEEE ICWS. His research interests include graph computing, social computing, data mining, and intelligent computing.

XINDONG WU (Fellow, IEEE) received the Ph.D. degree in artificial intelligence from the University of Edinburgh, U.K. He is currently the Chief Scientist with the Mininglamp Academy of Sciences, Mininglamp Technology, Beijing, China, and a Chang Jiang Scholar with the School of Computer Science and Information Engineering, Hefei University of Technology, China. His research interests include data mining, knowledge-based systems, and web information exploration. He is the Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM), the Editor-in-Chief of *Knowledge and Information Systems*, and the Editor-in-Chief of the Springer book series, *Advanced Information and Knowledge Processing* (AIKP). He is a Fellow of the AAAS.