Block-Structured Optimization for Subgraph Detection in Interdependent Networks

Fei Jie*,†, Chunpai Wang‡, Feng Chen§, Lei Li*, Xindong Wu*∥

* Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, Hefei, China.
†School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China.
‡Department of Computer Science, University at Albany – SUNY, Albany, NY, USA
§Erik Jonsson School of Engineering & Computer Science, The University of Texas at Dallas, Dallas, TX, USA
∥Mininglamp Academy of Sciences, Mininglamp Technologies, Beijing, China

Abstract—We propose a generalized framework for block-structured nonconvex optimization, which can be applied to structured subgraph detection in interdependent networks, such as multi-layer networks, temporal networks, networks of networks, and many others. Specifically, we design an effective, efficient, and parallelizable projection algorithm, namely Graph Block-structured Gradient Projection (GBGP), to optimize a general non-linear function subject to graph-structured constraints. We prove that our algorithm: 1) runs in nearly-linear time on the network size; 2) enjoys a theoretical approximation guarantee. Moreover, we demonstrate how our framework can be applied to two very practical applications and conduct comprehensive experiments to show the effectiveness and efficiency of our proposed algorithm.

Index Terms—subgraph detection, sparse optimization, interdependent networks

I. INTRODUCTION

Subgraph detection in network data has aroused many interests in recent years because of many real-world applications, such as disease outbreak detection [1], intrusion detection in computer networks, event detection in social networks [2], congestion detection in traffic networks, etc. However, most of existing works investigate the subgraph mining on static, isolated networks, and such a problem involving interdependent networks has not been well studied. Interdependent networks are comprised of multiple networks \{G_1, G_2, \ldots, G_k, \ldots\} and edges E_0 interconnected among networks, where G_k = (V_k, E_k). V_k and E_k are vertex set and edge set of k\textsuperscript{th} network G_k respectively. Some nodes in different networks exhibit node-node dependencies that could be captured by explicit edges or implicit correlation on node attributes (implicit edges). For instance, a temporal network can be viewed as multiple temporal-dependent networks, in which each network represents a snapshot of the temporal network at a specific time stamp, where current node’s attributes depend on attributes in the previous time-stamp implicitly [3] (Figure 1a). A web-scale social network comprised of many communities is a network of networks (a trivial interdependent networks) with explicit connections, where communities can be captured by explicit edges or implicit correlation on node attributes.

The first two authors contributed equally to this research.

Fig. 1: Examples of Interdependent Networks. (a) Temporal Networks: black dashed lines capture implicit temporal dependencies or consistencies. (b) Network of Networks: black solid lines are bridges across networks.

be viewed as small networks or blocks that interconnect with each other (Figure 1b).

Subgraph detection in multiple interdependent networks can be formulated as a block-structured optimization problem with multiple topological constraints on blocks,

\[
\begin{align*}
\min_{S_1 \subseteq V_1, \ldots, S_K \subseteq V_K} & \quad F(S_1, \ldots, S_K) \\
\text{s.t.} & \quad S_k \text{ satisfies a pre-defined topological constraint,}
\end{align*}
\]

where F is a user-specified cost function regularized by block dependencies, for example, F could be \( f(S_1, \ldots, S_K) + g(S_1, \ldots, S_K) \), where \( f \) is used to capture signals in interdependent networks and \( g \) models the dependencies between networks. \( S_k \) is a subset of nodes in k\textsuperscript{th} network G_k, \( k = 1, \ldots, K \). Vanilla subgraph detection problem is a special case of problem (1) when number of networks (blocks) is 1.

To the best of our knowledge, most of related studies on subgraph detection in interdependent networks only focus on specific applications and are lack of generality. Furthermore, they are heuristic-driven with no theoretical guarantee. Therefore, we propose a general framework that leverages graph structured sparsity model [4] and block coordinate descent method [5] to solve this problem which can be modeled as a block-structured optimization problem.

The contributions of our work are summarized as follows:
• Design of an efficient and scalable approximation algorithm. We propose a novel generic framework, namely, Graph Block-structured Gradient Projection, for block structured nonconvex optimization, which can be used to approximately solve a broad class of subgraph detection problems in interdependent networks in nearly-linear time.

• Theoretical guarantees. We present a theoretical analysis of the proposed GBGP algorithm and show that it enjoys a good convergence rate and a tight error bound on the quality of the detected subgraph.

• Two practical applications with comprehensive experiments. We demonstrate how our framework can be applied to two practical applications: 1) anomalous evolving subgraph detection; 2) subgraph detection in network of networks. We conduct comprehensive experiments on both synthetic and real networks to validate the effectiveness and efficiency of our proposed algorithm.

II. METHODOLOGY

A. Problem Formulation

First, we reformulate the combinatorial problem (1) in discrete space as a nonconvex optimization problem in continuous space. Interdependent networks can be viewed as one large network \( G = (V, E) \), where \( V = \{1, \ldots, N\} \) could be cut into \( \{V^1, \ldots, V^K\} \) and \( E \) could be split into \( \{E^0, E^1, \ldots, E^K\} \). Each pair of \( (V^k, E^k) \) forms a small network \( G^k \) for \( k = 1, \ldots, K \), and \( E^0 \) are edges interconnected among different small networks. Edges in \( E^0 \) should be treated differently with the edges in each \( E^k \), since they models the dependencies among different networks. \( \mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_N] \in \mathbb{R}^{P \times N} \) is the feature matrix, and \( \mathbf{w}_i \in \mathbb{R}^P \) is the feature vector of vertex \( i \), \( i \in V \). \( N_k = |V^k| \) is the size of the subset of vertices \( V^k \).

The general subgraph detection problem in interdependent networks can be formulated as following general block-structured optimization problem with topological constraints:

\[
\min_{\mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^K)} F(\mathbf{x}^1, \ldots, \mathbf{x}^K)
\]

\[
s.t. \quad \text{supp}(\mathbf{x}^k) \in M(G^k, s), \quad k = 1, \ldots, K
\]

where the vector \( \mathbf{x} \in \mathbb{R}^N \) is partitioned into multiple disjoint blocks \( \mathbf{x}^1 \in \mathbb{R}^{N_1}, \ldots, \mathbf{x}^K \in \mathbb{R}^{N_K} \), and \( \mathbf{x}^k \) are variables associated with nodes of network \( G^k \). The objective function \( F(\cdot) \) is a continuous, differentiable and convex function, which will be defined based on the feature matrix \( \mathbf{W} \). In addition, \( F(\cdot) \) could be decomposed as \( f(\mathbf{x}) + g(\mathbf{x}) \), where \( f \) is used to capture signals on nodes in interdependent networks and \( g \) models the dependencies between networks. \( \text{supp}(\mathbf{x}^{k}) \) denotes the support set of vector \( \mathbf{x}^{k} \), \( M(G^k, s) \) denotes all possible subsets of vertices in \( G^k \) that satisfy a certain predefined topological constraint. One example of topological constraint for defining \( M(G^k, s) \) is connected subgraph, and we can formally define it as follows:

\[
M(G^k, s) := \{ S | S \subseteq V^k; |S| \leq s; G^k_S \text{ is connected} \}
\]

where \( s \) is a predefined upperbound size of \( S \), \( S \subseteq V^k \), and \( G^k_S \) refers to the induced subgraph by a set of vertices \( S \). The topological constraints can be any graph structured sparsity constraints on \( G^k_S \), such as connected subgraphs, dense subgraphs, compact subgraphs [6]. Moreover, we do not restrict all \( \text{supp}(\mathbf{x}^1), \ldots, \text{supp}(\mathbf{x}^K) \) satisfying an identical topological constraint.

Algorithm 1 Graph Block-structured Gradient Projection

Input: \( \{G^1, \ldots, G^K\} \)

Output: \( \mathbf{x}^{1, i}, \ldots, \mathbf{x}^{K, i} \)

Initialization. \( i = 0, \mathbf{x}^{k, i} = \) initial vectors, \( k = 1, \ldots, K \)

repeat

1: for \( k = 1, \ldots, K \) do

2: \( \Gamma^{k} = H(\nabla_{\mathbf{x}^k} F(\mathbf{x}^{1, i}, \ldots, \mathbf{x}^{K, i})) \)

3: \( \Omega^{k} = \Gamma^{k} \cup \text{supp}(\mathbf{x}^{k}) \)

5: end for

6: Get \( (b^{1, i}, \ldots, b^{K, i}) \) by solving problem (6)

7: for \( k = 1, \ldots, K \) do

8: \( \Psi^{i+1} = T(b^{k, i}) \)

9: \( \mathbf{x}^{k, i+1} = [\mathbf{x}^{k, i}] \cdot \Psi^{i+1} \)

10: end for

11: \( i = i + 1 \)

12: until \( \sum_{k=1}^{K} \|\mathbf{x}^{k, i+1} - \mathbf{x}^{k, i}\| \leq \epsilon \)

13: \( C = (\Psi^{1}, \ldots, \Psi^{K}) \)

14: return \( (\mathbf{x}^{1, i}, \ldots, \mathbf{x}^{K, i}), C \)

B. Head and Tail Projections on \( M(G, s) \)

• Tail Projection \( (T(\mathbf{x})) \): is to find a subset of nodes \( S \subseteq V \) such that

\[
\|\mathbf{x} - \mathbf{x}_S\|_2 \leq c_T \cdot \min_{\mathbf{x}_S^r \in M(G, s)} \|\mathbf{x} - \mathbf{x}_S^r\|_2,
\]

where \( c_T \geq 1 \), and \( \mathbf{x}_S \) is a restriction of \( \mathbf{x} \) on \( S \) such that \( (\mathbf{x}_S)_i = (\mathbf{x})_i \) if \( i \in S \), and \( (\mathbf{x}_S)_i = 0 \) otherwise. When \( c_T = 1 \), \( T(\mathbf{x}) \) returns an optimal solution to the problem: \( \min_{\mathbf{x}_S^r \in M(G, s)} \|\mathbf{x} - \mathbf{x}_S^r\|_2 \). When \( c_T > 1 \), \( T(\mathbf{x}) \) returns an approximate solution to this problem with the approximation factor \( c_T \).

• Head Projection \( (H(\mathbf{x})) \): is to find a subset of nodes \( S \) such that

\[
\|\mathbf{x}_S\|_2 \geq c_H \cdot \max_{\mathbf{x}_S^r \in M(G, \mathbf{s})} \|\mathbf{x}_S^r\|_2,
\]

where \( c_H \leq 1 \). When \( c_H = 1 \), \( H(\mathbf{x}) \) returns an optimal solution to the problem: \( \max_{\mathbf{x}_S^r \in M(G, \mathbf{s})} \|\mathbf{x}_S^r\|_2 \). When \( c_H < 1 \), \( H(\mathbf{x}) \) returns an approximate solution to this problem with the approximation factor \( c_H \).

Although the head and tail projections are NP-hard when we restrict \( c_T = 1 \) and \( c_H = 1 \), these two projections can still be implemented in nearly-linear time when approximated solutions with \( c_T > 1 \) and \( c_H < 1 \) are allowed.

C. Algorithm Details

We propose a novel Graph Block-structured Gradient Projection, namely GBGP, to approximately solve problem (2) in nearly-linear time on the network size. The key idea is to
alternatively search for a close-to-optimal solution by solving easier sub-problems for graph $G_k$ in each iteration $i$ until converged. The pseudo-code of our proposed algorithm is described in Algorithm 1. Our algorithm can be decomposed into three main steps, including:

- **Step 1**: alternatively identify a subset of nodes in each block $\Omega_{x_k}$, in which pursuing the minimization will be most effective (Line 2 5).
- **Step 2**: identify the intermediate solution $\{b^i_{x_1}, \ldots, b^i_{x_K}\}$ that minimizes the objective function in intermediate space $\bigcup_{k=1}^{K} \Omega_{x_k}$ (Line 6):

$$\{b^i_{x_1}, \ldots, b^i_{x_K}\} = \text{argmin}_{x^1, \ldots, x^K} F(x^1, \ldots, x^K)$$

subject to $\text{supp}(x^k) \subseteq \Omega_{x_k}$

- **Step 3**: alternatively apply tail projections on the intermediate solution $\{b^i_{x_1}, \ldots, b^i_{x_K}\}$ to the feasible space defined by constraints: “$\text{supp}(x^k) \subseteq \Omega_{x_k}$” (Line 7 10).

We utilize the block-coordinate descent method with proximal linear update [7], [8] to solve the problem (6) (Algorithm 2). In addition, proximal linear update is used to ensure the convergence of the algorithm on convex problems with convex constraints “$\text{supp}(x^k) \subseteq \Omega_{x_k}$”. The proximal linear update in our scenario is defined by:

$$x^{k,t+1} = \text{argmin}_x \left\{ F(x^t) + \langle \nabla_k F(x^t), x^t-x^t \rangle \right\}$$

$$+ \frac{1}{2\alpha_k} \| x^t - x^t \|_2^2$$

subject to $\text{supp}(x^k) \subseteq \Omega_{x_k}$

(7)

where $\alpha_k^t$ serves as a step size and can be set as the reciprocal of the Lipschitz constant of $\nabla_k F(x^{k,t}, \hat{x}^{k,t})$, and $\hat{x}^{k,t}$ (Line 4) is an extrapolated point that helps accelerate the convergence of the proximal point update scheme. The overall block coordinated gradient projection method on convex function with convex constraint (i.e. Algorithm 2) has a sublinear rate of convergence [8].

**Algorithm 2 Block-Coordinate Descent Method with Proximal Linear Update to Solve Problem (6)**

**Input**: $\{G^1, \ldots, G^K\}$  
**Output**: $x^1, \ldots, x^K$  
**Initialization**: $t = 0$, $\epsilon = 10^{-3}$, $\rho_0 = 1$.

1: repeat

2: Choose index $k \in \{1, \ldots, K\}$

3: $\rho_t = (\rho_t - 1)/\rho_t$

4: $\hat{x}^{k,t} = x^{k,t} + \rho_t(x^{k,t} - x^{k,t-1})$

5: Update $x^{k,t+1} \leftarrow \hat{x}^{k,t} - \frac{1}{\rho_t+\epsilon} \nabla_k F(\hat{x}^{k,t}, \hat{x}^{k,t})$

6: Project $x^{k,t+1}$ to feasible space by setting entries of $x^{k,t+1}$ to zero if index of entry not in set $\Omega_{x_k}$.

7: Keep $x^{i,t+1} = x^{i,t}$, for all $i \neq k$.

8: $\rho_t+1 = (1 + \sqrt{1+4\rho_t^2})/2$.

9: Let $t = t + 1$

10: until $\sum_{k=1}^{K} \| x^{k,t} - x^{k,t-1} \|_2 \leq \epsilon$

11: return $\{x^1, \ldots, x^K\}$

**III. THEORETICAL ANALYSIS**

In order to demonstrate the accuracy and efficiency of GBGP, we require that the objective function $F(x)$ satisfies the Weak Restricted Strong Convexity (WRSC) condition, which is a variant of the Restricted Strong Convexity/Smoothness (RSC/RSS) [9]:

**Definition 1** (Weak Restricted Strong Convexity (WRSC)). A function $F(x)$ has condition ($\xi, \delta, M$)-WRSC, if $\forall x, y \in \mathbb{R}^N$ and $\forall S \in \mathbb{M}$ with $\text{supp}(x) \cup \text{supp}(y) \subseteq S$, the following inequality holds for some $\xi > 0$ and $0 < \delta < 1$:

$$\| x - y - \nabla_x F(x) + \nabla_y F(y) \|_2 \leq \delta \| x - y \|_2$$

(8)

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

**Theorem 1.** Consider the graph block-structured constraint with $K$ blocks $M(G, s) = \bigcup_{k=1}^{K} M(G^k, s_k)$ and a cost function $F : \mathbb{R}^N \rightarrow \mathbb{R}$ that satisfies condition ($\xi, \delta, M(G, 8s)$)-WRSC. If $\eta = c_H (1 - \delta) - \delta > 0$, then for any true $x^* \in \mathbb{R}^N$ with $\text{supp}(x^* \in M(G, s)$, the iteration of algorithm obeys

$$\| x^{i+1} - x^* \|_2 \leq \alpha \| x^i - x^* \|_2 + \beta \| \nabla_i F(x^i) \|_2$$

(9)

where $c_H = \min_{k=1,\ldots,K} c_{H_k}$, $c_T = \max_{k=1,\ldots,K} c{T_k}$, $I = \arg\max_{S \in M} \| \nabla S F(x) \|_2$, $\alpha = \frac{1 + c_T}{1 - \frac{\beta}{\alpha}} \sqrt{1 - \eta^2}$, and $\beta = \frac{\gamma + \eta}{\sqrt{1 - \eta^2}}$.

**Theorem 2.** Let $x^* \in \mathbb{R}^N$ be a true optimum such that $\text{supp}(x^*) \subseteq M(G, s)$, and $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a cost function that satisfies condition ($\xi, \delta, M(G, 8s)$)-WRSC. Assuming that $\alpha < 1$, GBGP returns an $x$ such that, $\text{supp}(x) \subseteq M(G, 5s)$ and $\| x^* - x \|_2 \leq c \| \nabla_i F(x^*) \|_2$, where $c = (1 + \frac{\beta}{1 - \alpha})$ is a fixed constant. Moreover, GBGP runs in time

$$O \left( \left( T + \sum_{k=1}^{K} |E_k| \log^3 N_k \right) \log \left( \frac{\| x^i \|_2}{\| \nabla_i F(x^i) \|_2} \right) \right)$$

(10)

where $|E_k|$, $N_k$ denote edge and node size of $k$th block and $T$ is the time complexity of one execution of the subproblem in line 6 of Algorithm 1. In particularly, if $T$ scales linearly with $N$ and $|E|$, then GBGP scales nearly linearly with $N$ and $|E|$.

Note that the proofs of Theorem 1 and Theorem 2 are omitted due to space limitation.

**IV. EXAMPLE APPLICATIONS**

In this section, we show how to formulate two subgraph detection applications: 1) anomalous evolving subgraph detection and 2) subgraph detection in network of networks as
problem (2) with specific objective function $F$ and topological constraints. For these two applications, we leverage the Elevated Mean Scan (EMS) statistics, which is defined as: $c^T x / \sqrt{x^T 1}$, where $x \in \{0, 1\}^N$. $c$ denotes the feature vector of all nodes, and $c_i \in \mathbb{R}$ denotes the uni-variate feature for node $i$. Assuming $S$ is some unknown anomalous cluster which forms a connected component, $S \subseteq \mathbb{V}$. Empirically, maximizing the score of EMS leads to discovering significant nodes in the network precisely. Instead of maximizing the EMS in the domain $\{0, 1\}^N$, we relax EMS to continuous space and minimize the relaxed negative EMS in our applications, which can be defined as:

$$-\left(\frac{c^T x}{x^T 1}\right)^2 + \frac{1}{2} \|x\|^2 \quad \text{where} \quad x \in [0, 1]^N$$

(11)  

Most importantly, the relaxed negative EMS satisfies the RSC/RSS condition when $c$ is normalized, which implies WRSC condition [6], [9].

A. Anomalous Evolving Subgraphs Detection

We can leverage the relaxed EMS and mathematically formulate the anomalous evolving subgraphs detection problem as nonconvex optimization with convex objective function and block-structured constraints:

$$\min_{x^k, \ldots, x^K} \sum_{k=1}^{K} \left( -\frac{(c^T x^k)^2}{x^k 1} + \frac{1}{2} \|x^k\|^2 \right) + \lambda \cdot \sum_{k=2}^{K} \|x^k - x^{k-1}\|^2$$

s.t. $\text{supp}(x^k) \in \mathbb{M}(G^k, s)

(12)

where the first term is the summation of relaxed negative EMS, and the second term is soft constraints on $x^k$ and $x^{k-1}$ to ensure temporal consistency on detected subgraphs, and $\lambda > 0$ is a trade-off parameter. The connected subset of nodes at time stamp $k$ can be found as $S_k = \text{supp}(x^k)$, i.e., the support set of the estimated $x^k$ that minimizes the objective function.

B. Subgraph Detection in Network of Networks

Our proposed framework is also applicable to subgraph detection in network of networks. For subgraph detection in a network of networks, we can also leverage the relaxed negative EMS and formulate the detection problem in large-scale networks as follows:

$$\min_{x^1, \ldots, x^K} \sum_{k=1}^{K} \left( -\frac{(c^T x^k)^2}{x^k 1} + \frac{1}{2} \|x^k\|^2 \right) + \lambda \cdot \sum_{i,j} e_{ij} \cdot (x_i - x_j)^2$$

s.t. $\text{supp}(x^k) \in \mathbb{M}(G^k, s)

(13)

where the first term is the summation of relaxed negative EMS, the second term is soft constraints on bridge nodes of two partitions to ensure dependencies; $e_{ij} = 1$ if node $i$ and node $j$ are connected but in two different partitions (in other words, edge $(i, j)$ is an graph cut), otherwise $e_{ij} = 0$, $x_i$ and $x_j$ are $i^{th}$ and $j^{th}$ entries of $x$, and $\lambda > 0$ is a trade-off parameter. In addition, we propose a parallel version of our algorithm to speed up the computation by integrating the APPROX algorithm, a randomized coordinate descent method proposed in [5].

V. EXPERIMENTS

A. Anomalous Evolving Subgraph Detection

a) Synthetic Dataset: We generate networks using Barabási-Albert preferential attachment model [10]. The evolving true subgraphs spanning within 7 time stamps are simulated from node size 100 to 300, and the true subgraphs in two consecutive time stamps have 50% of node overlap. The univariate feature values of background nodes and true nodes are randomly generated in $\mathbb{N}(0, 1)$ and $\mathbb{N}(\mu, 1)$ distributions, respectively. We generate 50 temporal networks for each setting of $\mu = [3, 4, 5]$.  

b) Real-world Dataset: 1) Water Pollution Dataset: a real world sensor network [11]. For each hour, each vertex has a sensor that reports 1 if it is polluted; otherwise, reports 0.  

2) Washington D.C. Road Traffic Dataset: a traffic dataset of Washington D.C from INRIX \(^1\).  

3) Beijing Road Traffic Dataset: the dataset contains the real-time traffic conditions of Beijing city. [12]. For both traffic datasets, the node attribute is the difference between reference speed and current speed, and the true congested roads are provided. Statistics of all datasets are provided in Table II.

c) Performance Metrics: Precision, Recall, and F-measure are deployed to evaluate the quality of detected subgraphs by different methods. Higher F-measure reveals better overall performance. For synthetic data, we use the averaged precision, recall, and f-measure over 50 simulated examples.

d) Comparison Methods and Results: We compare our algorithm with two state of the art baseline methods: Meden [13] and NetSpot [14], which were designed specifically for detecting significant anomalous region in dynamic networks and provide implementations. The comparison of results are reported in Table I and Table III. As you can see, our method outperforms these two baseline methods on both synthetic data and real-world data. Both of baselines are heuristic, which can not guarantee the quality of results and cause worse performance than ours.

e) Robustness Validation: Except for measuring the accuracy of subgraph detection, we also test the robustness of subgraph detection method on water pollution dataset as [15], [16]. $P$ percent of nodes are selected randomly, and their sensor binary values are flipped in order to test the robustness of methods to noises, where $P \in \{2, 4, 6, 8, 10\}$. Figure 2 shows the precision, recall, and f-measure of all the comparison methods on the detection of polluted nodes in the water pollution dataset with respect to different noise ratios. The results indicate that our proposed method GBGP is the best overall performance for all of the settings, which verifies the robustness of our method.

B. Subgraph Detection in Network of Networks

a) Synthetic Datasets: We generate several networks with different network sizes using Barabási-Albert model, and then apply random walk algorithm to simulate the ground-truth

\(^1\)http://inrix.com/publicsector.asp.
TABLE I: Results on synthetic datasets with different $\mu$. It shows that GBGP is more robust than Meden and Netspot.

| Methods   | $\mu = 3$ |        |        |        | $\mu = 4$ |        |        | $\mu = 5$ |        |        |
|-----------|-----------|--------|--------|--------|-----------|--------|--------|-----------|--------|--------|
|           | Precision | Recall | $F$-measure | Precision | Recall | $F$-measure | Precision | Recall | $F$-measure |
| Meden     | 0.7588    | 0.7453 | 0.7455 | 0.8586 | 0.8591 | 0.8107 | 0.7964 | 0.7956 | 0.8068 |
| NetSpot   | 0.6658    | 0.7267 | 0.6947 | 0.7615 | 0.7922 | 0.7763 | 0.7956 | 0.8185 | 0.8068 |
| GBGP      | 0.6468    | 0.8899 | 0.7489 | 0.8487 | 0.9674 | 0.9041 | 0.9553 | 0.9914 | 0.9730 |

Fig. 2: Precision vs Noise Level, Recall vs Noise Level, F-measure vs Noise Level

TABLE II: Statistics of Datasets for the 1st Application.

| Datasets          | Statistics                  | Node | Edge | Timestamp | Resolution |
|-------------------|-----------------------------|------|------|-----------|------------|
| Synthetic         | dot dataset                  | 3,000| 11,984| 7         | NA         |
| Water Pollution   | synthetic network             | 12,527| 14,831| 8         | 60 min.    |
| Beijing           | road traffic                | 59,000| 70,317| 12        | 10 min.    |

TABLE III: Results on Washington D.C. and Beijing datasets.

| Methods   | Washington D.C. | Beijing |
|-----------|-----------------|---------|
|           | Precision | Recall | $F$-measure | 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 |
| GBGP      | 0.7049   | 0.9192 | 0.7853 | 0.6627 | 0.9634 | 0.7788 |

subgraph with size as 10% of network size. The nodes in true subgraph have features following normal distribution $N(5, 1)$, and the features of background nodes follows distribution $N(0, 1)$. The synthetic datasets are used for scalability analysis in terms of size of nodes and size of edges, which we denote them as SynNode and SynEdge respectively.

b) Real-world Datasets: 1) Beijing Road Traffic Dataset: we use static network data per time stamp from 5PM to 7PM, in previous application. 2) Wikivote Dataset: the network contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. 3) CondMat Dataset: the collaboration network is from the e-print arXiv and covers scientific collaborations between authors papers submitted to Condense Matter category. For Wikivote and CondMat datasets, we simulate the true subgraphs of size $\sim 30,000$ nodes and $\sim 10,000$ edges respectively. Their performances are not as good as our method.

c) Performance Metrics: Except for metrics (precision, recall and f-measure) used for evaluating the detection performance, we also compare and report the run time among different methods in this application to evaluate the scalability.

d) Comparison Methods and Results: We compare our method with three baselines: 1) EventTree [2], 2) AdditiveGraphScan [17], and 3) LTSS [18], which were designed specifically for event detection on static networks. The average precision, recall, f-measure, as well as run time on all methods are reported in Table V. Our method outperforms the baselines in terms of f-measure by the compromise on a small amount of run time. All of baselines have their own shortcomings. Despite AdditiveGraphScan can get comparable performance as our method on some datasets, it is a heuristic algorithm without theoretical guarantees and not scalable for large scale networks. We do not report the result of AdditiveGraphScan on DBLP dataset, since it takes over one day to run and infeasible to tune the parameters. EventTree and LTSS are scalable, but their performances are not as good as our method.
TABLE V: Results on Beijing, Wikivote, CondMat and DBLP datasets. The run time is measured in seconds.

| Method         | Precision | Recall | F-measure | Run Time |
|----------------|-----------|--------|-----------|----------|
| AdiveGraphScan | 0.4295    | 0.6884 | 0.9474    | 0.9747   |
| EvenTree       | 0.5547    | 0.5777 | 0.9637    | 0.9747   |
| LTSS           | 0.5144    | 0.8333 | 0.9637    | 0.9747   |
| GBGP(Serial)   | 0.9105    | 0.7283 | 0.9637    | 0.9747   |
| GBGP(Parallel) | 0.9105    | 0.7283 | 0.9637    | 0.9747   |

VI. RELATED WORK

a) Subgraph Detection. Subgraph detection methods mainly find subgraphs that satisfy some topological constraints, such as connected subgraphs, dense subgraphs and compact subgraphs, including EvenTree [2], NPHGS[1] for static graphs, Meden [13], NetSpot[14], and AdditiveGraphScan [17] for dynamic graphs, which are all heuristic. b) Structured Sparse Optimization. The seminal work on general approximate graph-structured sparsity model is [4]. General structured optimization methods on single graph was proposed to do subgraph [6], [16] or subspace [20] detection.

VII. CONCLUSION AND FUTURE WORK

This paper presents a general framework, GBGP, to solve a nonconvex optimization problem subject to graph block-structured constraints in nearly linear time with a theoretical approximation guarantee. We evaluate our model on two applications, and results of both experiments show that the algorithm enjoys better effectiveness and efficiency than state of the art methods while our work is a general framework and can be used in more scenarios. For future work, we will extend the work on network data with high-dimensional node attributes and different graph topological constraints.

ACKNOWLEDGMENTS

The first author has been supported by the CSC scholarship. This work has been supported by the National Key Research and Development Program of China under grant 2016YFB100009, the National Natural Science Foundation of China under grant 91746209, the Program for Changjiang Scholars and Innovative Research Team in University (PC-SIRT) of the Ministry of Education of China under grant IRT17R32, and the US National Science Foundation under grant IIS-1815696 and IIS-1759011.

REFERENCES

[1] F. Chen and D. B. Neill, “Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs,” in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2014, pp. 1166–1175.
[2] P. Rozenstein, A. Anagnostopoulos, A. Gionis, and N. Tatti, “Event detection in activity networks,” in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2014, pp. 1176–1185.
[3] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, “Community structure in time-dependent, multisu...