2019 IEEE International Conference on Data Mining (ICDM)

Block-Structured Optimization for Anomalous Pattern Detection in Interdependent Networks

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

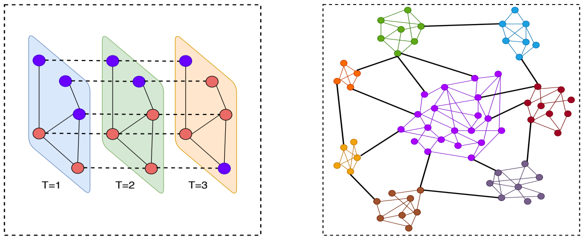
* Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, Hefei, China
†School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China
‡Department of Computer Science, University at Albany – SUNY, Albany, NY, USA
§Erik Jonsson School of Engineering & Computer Science, The University of Texas at Dallas, Dallas, TX, USA
¶Mininglamp Academy of Sciences, Mininglamp Technology, Beijing, China
‖Institute of Big Knowledge Science, Hefei University of Technology, Hefei, China
realfjie@gmail.com, cwang25@albany.edu, feng.chen@utdallas.edu, {lilei,xwu}@hfut.edu.cn

***Abstract*—We propose a generalized optimization framework for detecting anomalous patterns (subgraphs that are interesting or unexpected) in interdependent networks, such as multi-layer networks, temporal networks, networks of networks, and many others. We frame the problem as a non-convex optimization that has a general nonlinear score function and a set of block-structured and non-convex constraints. We develop an effective, efficient, and parallelizable projection-based algorithm, namely Graph Block-structured Gradient Projection (GBGP), to solve the problem. It is proved that our algorithm 1) runs in nearly-linear time on the network size, and 2) enjoys a theoretical approximation guarantee. Moreover, we demonstrate how our framework can be applied to two very practical applications, and we conduct comprehensive experiments to show the effectiveness and efficiency of our proposed algorithm.**

***Index Terms*—subgraph detection, sparse optimization, interdependent networks**

I. Introduction

Anomalous pattern 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 $\{\mathbb{G}^1, \mathbb{G}^2, \ldots, \mathbb{G}^k, \ldots\}$ and edges $\mathbb{E}^0$ interconnected among networks, where $\mathbb{G}^k = (\mathbb{V}^k, \mathbb{E}^k)$. $\mathbb{V}^k$ and $\mathbb{E}^k$ are vertex set and edge set of k^{th} network $\mathbb{G}^k$ 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 viewed as small networks or blocks that interconnect with each other (Figure 1b).



(a) Temporal Networks (b) Network of Networks

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.

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

$$\begin{aligned} &\min_{S_1 \subseteq \mathbb{V}^1, \ldots, S_K \subseteq \mathbb{V}^K} F(S_1, \cdots, S_K) \\ &\text{s.t.} \quad S_k \text{ satisfies a pre-defined topological constraint,} \end{aligned} \tag{1}$$

where F is a user-specified cost function regularized by block dependencies, for example, F could be $f(S_1, \cdots, S_K) + g(S_1, \cdots, S_K)$, 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^{th} network $\mathbb{G}^k$, $k = 1, ..., K$. 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 anomalous pattern 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:

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00137

- **Design of an efficient and scalable approximation algorithm.** We propose a novel generic framework, namely, Graph Block-structured Gradient Projection, for block structured non-convex optimization, which can be used to approximately solve a broad class of anomalous pattern detection problems in interdependent networks.
- **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 subgraphs.
- **Comprehensive empirical anlaysis.** We demonstrate how our framework can be applied to two practical applications: 1) anomalous evolving subgraph detection in temporal networks; 2) anomalous 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 non-convex optimization problem in continuous space. Interdependent networks can be viewed as one large network $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V} = \{1, \cdots, N\}$ could be cut into $\{\mathbb{V}^1, \cdots, \mathbb{V}^K\}$ and $\mathbb{E}$ could be split into $\{\mathbb{E}^0, \mathbb{E}^1, \cdots, \mathbb{E}^K\}$. Each pair of $(\mathbb{V}^k, \mathbb{E}^k)$ forms a small network $\mathbb{G}^k$ for $k = 1, \cdots, K$, and $\mathbb{E}^0$ are edges interconnected among different small networks. Edges in $\mathbb{E}^0$ should be treated differently with the edges in each $\mathbb{E}^k$, since they models the dependencies among different networks. $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_N] \in \mathbb{R}^{P \times N}$ is the feature matrix, and $\mathbf{w}_i \in \mathbb{R}^P$ is the feature vector of vertex i, $i \in \mathbb{V}$. $N_k = |\mathbb{V}^k|$ is the size of the subset of vertices $\mathbb{V}^k$.

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

$$\begin{aligned} &\min_{\mathbf{x}=(\mathbf{x}^1,\ldots,\mathbf{x}^K)} F(\mathbf{x}_1, \ldots, \mathbf{x}_K) \\ &\text{s.t. } \operatorname{supp}(\mathbf{x}^k) \in \mathbb{M}(\mathbb{G}^k, s), \quad k = 1, \cdots, K \end{aligned} \tag{2}$$

where the vector $\mathbf{x} \in \mathbb{R}^N$ is partitioned into multiple disjoint blocks $\mathbf{x}^1 \in \mathbb{R}^{N_1}, \cdots, \mathbf{x}^K \in \mathbb{R}^{N_K}$, and $\mathbf{x}^k$ are variables associated with nodes of network $\mathbb{G}^k$. 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. $\operatorname{supp}(\mathbf{x}^k)$ denotes the support set of vector $\mathbf{x}^k$, $\mathbb{M}(\mathbb{G}^k, s)$ denotes all possible subsets of vertices in $\mathbb{G}^k$ that satisfy a certain predefined topological constraint. One example of topological constraint for defining $\mathbb{M}(\mathbb{G}^k, s)$ is connected subgraph, and we can formally define it as follows:

$$\mathbb{M}(\mathbb{G}^k, s) := \{S | S \subseteq \mathbb{V}^k; |S| \leq s; \mathbb{G}^k_S \text{ is connected.}\} \tag{3}$$

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

Algorithm 1 Graph Block-structured Gradient Projection

```
Input: $\{\mathbb{G}^1, \ldots, \mathbb{G}^K\}$
Output: $\mathbf{x}^{1,t}, \cdots, \mathbf{x}^{K,t}$
Initialization, $i = 0$, $\mathbf{x}^{k,i}$ = initial vectors, k=1,..., K
repeat
    for $k = 1, \cdots, K$ do
        $\Gamma_{\mathbf{x}^k} = H(\nabla_{\mathbf{x}^k} F(\mathbf{x}^{1,i}, \ldots, \mathbf{x}^{K,i}))$
        $\Omega_{\mathbf{x}^k} = \Gamma_{\mathbf{x}^k} \cup \operatorname{supp}(\mathbf{x}^{k,i})$
    end for
    Get $(\mathbf{b}^i_{\mathbf{x}^1}, \ldots, \mathbf{b}^i_{\mathbf{x}^K})$ by solving problem (6)
    for $k = 1, \cdots, K$ do
        $\Psi^{i+1}_{\mathbf{x}^k} = T(\mathbf{b}^i_{\mathbf{x}^k})$
        $\mathbf{x}^{k,i+1} = [\mathbf{b}^i_{\mathbf{x}^k}]_{\Psi^{i+1}_{\mathbf{x}^k}}$
    end for
    $i = i + 1$
until $\sum_{k=1}^K \left\|\mathbf{x}^{k,i+1} - \mathbf{x}^{k,i}\right\| \leq \epsilon$
$C = (\Psi^i_{\mathbf{x^1}}, \ldots, \Psi^i_{\mathbf{x}^k})$
return $(\mathbf{x}^{1,i}, \cdots, \mathbf{x}^{K,i}), C$
```

B. Head and Tail Projections on $\mathbb{M}(\mathbb{G}, s)$

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

$$\|\mathbf{x} - \mathbf{x}_S\|_2 \leq c_T \cdot \min_{S' \in \mathbb{M}(\mathbb{G},s)} \|\mathbf{x} - \mathbf{x}_{S'}\|_2, \tag{4}$$

  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: $\min_{S' \in \mathbb{M}(\mathbb{G},s)} \|\mathbf{x} - \mathbf{x}_{S'}\|_2$. 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

$$\|\mathbf{x}_S\|_2 \geq c_H \cdot \max_{S' \in \mathbb{M}(\mathbb{G},s)} \|\mathbf{x}_{S'}\|_2, \tag{5}$$

  where $c_H \leq 1$. When $c_H = 1$, $H(\mathbf{x})$ returns an optimal solution to the problem: $\max_{S' \in \mathbb{M}(\mathbb{G},s)} \|\mathbf{x}_{S'}\|_2$. 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 the 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 $\mathbb{G}_k$ 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 $\Omega_{\mathbf{x}^k}$, in which pursuing the minimization will be most effective (**Line 2** $\sim$ **5**).
- **Step 2**: identify the intermediate solution $(\mathbf{b}^i_{\mathbf{x}^1}, \ldots, \mathbf{b}^i_{\mathbf{x}^K})$ that minimizes the objective function in intermediate space $\cup_{k=1}^{K} \Omega_{\mathbf{x}^k}$ (**Line 6**);

$$\begin{aligned} \left(\mathbf{b}^i_{\mathbf{x}^1}, \ldots, \mathbf{b}^i_{\mathbf{x}^K}\right) &= \underset{\mathbf{x}^1,\ldots,\mathbf{x}^K}{\operatorname{argmin}} F(\mathbf{x}^1, \ldots, \mathbf{x}^K) \\ &\text{s.t.} \quad \operatorname{supp}(\mathbf{x}^k) \subseteq \Omega_{\mathbf{x}^k} \end{aligned} \tag{6}$$

- **Step 3**: alternatively apply tail projections on the intermediate solution $(\mathbf{b}^i_{\mathbf{x}^1}, \ldots, \mathbf{b}^i_{\mathbf{x}^K})$ to the feasible space defined by constraints: "$\operatorname{supp}(\mathbf{x}^k) \in \mathbb{M}(\mathbb{G}^k, s)$" (**Line 7** $\sim$ **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 "$\operatorname{supp}(\mathbf{x}^k) \subseteq \Omega_{\mathbf{x}^k}$". The proximal linear update in our scenario is defined by:

$$\begin{aligned} \mathbf{x}^{k,t+1} = &\underset{\mathbf{x}^k}{\operatorname{argmin}} F(\hat{\mathbf{x}}^t) + \langle \nabla_{\mathbf{x}^k} F(\hat{\mathbf{x}}^{k,t}, \hat{\mathbf{x}}^{\neq k,t}), \mathbf{x}^k - \hat{\mathbf{x}}^{k,t} \rangle \\ &+ \frac{1}{2\alpha^{k,t}} \|\mathbf{x}^k - \hat{\mathbf{x}}^{k,t}\|_2^2 \qquad \text{s.t.} \ \operatorname{supp}(\mathbf{x}^k) \subseteq \Omega_{\mathbf{x}^k} \end{aligned} \tag{7}$$

where $\alpha^{k,t}$ serves as a step size and can be set as the reciprocal of the Lipschitz constant of $\nabla_{\mathbf{x^k}} F(\hat{\mathbf{x}}^{k,t}, \hat{\mathbf{x}}^{\neq k,t})$, and $\hat{\mathbf{x}}^{k,t}$ (**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: ${\mathbb{G}^1, \ldots, \mathbb{G}^K}$
    Output: $\mathbf{x}^{1,t}, \cdots, \mathbf{x}^{K,t}$
    Initialization: $t = 0, \epsilon = 10^{-3}, \rho_0 = 1.$
 1: repeat
 2:     Choose index  $k \in \{1, \cdots, K\}$
 3:     $\omega_t = (\rho_t - 1)/\rho_t,$
 4:     $\hat{\mathbf{x}}^{k,t} = \mathbf{x}^{k,t} + \omega_t(\mathbf{x}^{k,t} - \mathbf{x}^{k,t-1})$
 5:     Update $\mathbf{x}^{k,t+1} \leftarrow \hat{\mathbf{x}}^{k,t} - \frac{1}{\alpha^{k,t}} \nabla_{\mathbf{x^k}} F(\hat{\mathbf{x}}^{k,t}, \hat{\mathbf{x}}^{\neq k,t})$
 6:     Project $\mathbf{x}^{k,t+1}$ to feasible space by setting entries of
        $\mathbf{x}^{k,t+1}$ to zero if index of entry not in set $\Omega_{\mathbf{x}^k}$.
 7:     Keep $\mathbf{x}^{j,t+1} = \mathbf{x}^{j,t}$, for all $j \neq k$
 8:     $\rho_{t+1} = (1 + \sqrt{1 + 4\rho_t^2})/2,$
 9:     Let $t = t + 1$
10: until $\sum_{k=1}^{K} \|\mathbf{x}^{k,t} - \mathbf{x}^{k,t-1}\| \leq \epsilon$
11: return $\{\mathbf{x}^{1,t}, \cdots, \mathbf{x}^{K,t}\}$
```

III. Theoretical Analysis

In order to demonstrate the accuracy and efficiency of GBGP, we require that the objective function $F(\mathbf{x})$ 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 $F(\mathbf{x})$ has condition $(\xi, \delta, \mathbb{M})$-WRSC, if $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ and $\forall S \in \mathbb{M}$ with $\operatorname{supp}(\mathbf{x}) \cup \operatorname{supp}(\mathbf{y}) \subseteq S$, the following inequality holds for some $\xi > 0$ and $0 < \delta < 1$:*

$$\|\mathbf{x} - \mathbf{y} - \xi \nabla_S F(\mathbf{x}) + \xi \nabla_S F(\mathbf{y})\|_2 \leq \delta \|\mathbf{x} - \mathbf{y}\|_2 \tag{8}$$

where $\mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^K), \mathbf{y} = (\mathbf{y}^1, \ldots, \mathbf{y}^K), \mathbf{x}^k, \mathbf{y}^k \in \mathbb{R}^{N_k}, k = 1, \ldots, K$, topological constraint $\mathbb{M}$ can be expressed as $\mathbb{M}(\mathbb{G}, s) = \bigcup_{k=1}^{K} \mathbb{M}(\mathbb{G}^k, s_k), s = \sum_{k=1}^{K} s_k$, and the subgraph in 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 union sign "$\bigcup$" to denote combined model $\mathbb{M}$, in which $\mathbf{x} \in \mathbb{M} = \{\mathbf{x} | \mathbf{x}^k \in \mathbb{M}(\mathbb{G}^k, s_k), k = 1, \ldots, K\}$.

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

$$\|\mathbf{x}^{i+1} - \mathbf{x}^*\|_2 \leq \alpha \|\mathbf{x}^i - \mathbf{x}^*\|_2 + \beta \|\nabla_I F(\mathbf{x}^*)\|_2 \tag{9}$$

where $c_H = \min_{k=1,\ldots,K}\{c_{H_k}\}$, $c_T = \max_{k=1,\ldots,K}\{c_{T_k}\}$, $I = \operatorname{argmax}_{S \in \mathbb{M}} \|\nabla_S F(\mathbf{x})\|_2$, $\alpha = \frac{1+c_T}{1-\delta}\sqrt{1 - \eta^2}$, and $\beta = \frac{\xi(1+c_T)}{1-\delta}\left[\frac{1+c_H}{\eta} + \frac{\eta(1+c_H)}{\sqrt{1-\eta^2}} + 1\right]$. c_{H_k} and c_{T_k} denote head and tail projection approximation factors on k^th block.

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

$$O\left(\left(T + \sum_{k=1}^{K} |\mathbb{E}^k| \log^3 N_k\right) \log\left(\frac{\|\mathbf{x}^*\|_2}{\|\nabla_I F(\mathbf{x}^*)\|_2}\right)\right) \tag{10}$$

*where $|\mathbb{E}^k|$, 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 $|\mathbb{E}|$, then GBGP scales **nearly linearly** with N and $|\mathbb{E}|$.*

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: $\mathbf{c}^\top\mathbf{x}/\sqrt{\mathbf{x}^\top\mathbf{1}}$, where $\mathbf{x} \in \{0,1\}^N$, $\mathbf{c}$ 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:

$$-\frac{(\mathbf{c}^\top\mathbf{x})^2}{\mathbf{x}^\top\mathbf{1}} + \frac{1}{2}\|\mathbf{x}\|_2^2 \quad \text{where} \quad \mathbf{x} \in [0,1]^N \tag{11}$$

Most importantly, the relaxed negative EMS satisfies the RSC/RSS condition when $\mathbf{c}$ is normalized, which implies the 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 non-convex optimization with convex objective function and block-structured constraints:

$$\min_{\mathbf{x}^1,\cdots,\mathbf{x}^K} \sum_{k=1}^{K}\left(-\frac{(\mathbf{c}^{k\top}\mathbf{x}^k)^2}{\mathbf{x}^{k\top}\mathbf{1}} + \frac{1}{2}\|\mathbf{x}^k\|_2^2\right) + \lambda \cdot \sum_{k=2}^{K}\|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2^2 \tag{12}$$
$$\text{s.t. } \mathrm{supp}(\mathbf{x}^k) \in \mathbb{M}(\mathbb{G}^k, s)$$

where the first term is the summation of relaxed negative EMS, and the second term is soft constraints on $\mathbf{x}^k$ and $\mathbf{x}^{k-1}$ 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 $S_k = \mathrm{supp}(\mathbf{x}^k)$, i.e., the support set of the estimated $\mathbf{x}^k$ 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 network of networks, we can also leverage the relaxed negative EMS and formulate the detection problem in large-scale networks as follows:

$$\min_{\mathbf{x}^1,\cdots,\mathbf{x}^K} \sum_{k=1}^{K}\left(-\frac{(\mathbf{c}^{k\top}\mathbf{x}^k)^2}{\mathbf{x}^{k\top}\mathbf{1}} + \frac{1}{2}\|\mathbf{x}^k\|_2^2\right) + \lambda \cdot \sum_{i,j} e_{ij}\cdot(x_i - x_j)^2 \tag{13}$$
$$\text{s.t. } \mathrm{supp}(\mathbf{x}^k) \in \mathbb{M}(\mathbb{G}^k, s)$$

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 a graph cut), otherwise $e_{ij} = 0$, x_i and x_j are i^{th} and j^{th} entries of $\mathbf{x}$, 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 Datasets: 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 has the best overall performance for all of the settings, which verifies the robustness of our method.

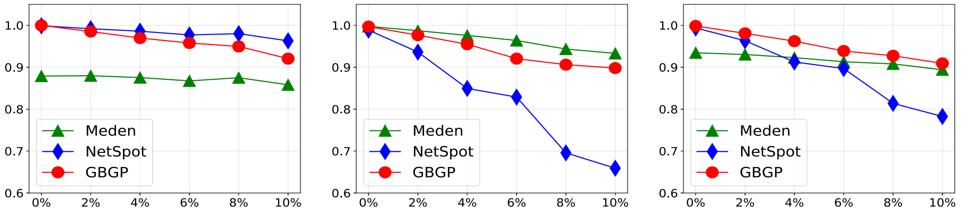
B. Subgraph Dectection in Network of Networks

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 μ. 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	0.7342	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
GBGP	0.6468	0.8899	**0.7489**	0.8487	0.9674	**0.9041**	0.9553	0.9914	**0.9730**



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

Fig. 2: Precision, Recall, and F-measure curves on Water Pollution dataset with respect to different noise ratios.

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

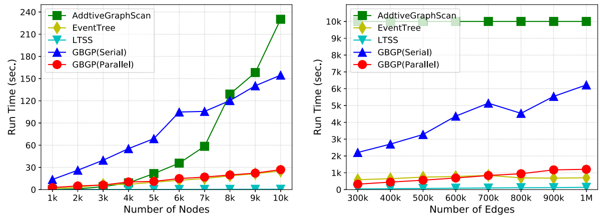
Datasets	Statistics			
	Node	Edge	Timestamp	Resolution
Synthetic	3,000	11,984	7	NA
Water Pollution	12,527	14,831	8	60 min.
Washington D.C.	1,188	1,323	17	60 min.
Beijing	59,000	70,317	12	10 min.

TABLE III: Results on Washingtong 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
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 $\mathbb{N}(5, 1)$, and the features of background nodes follow distribution $\mathbb{N}(0, 1)$. 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***[2]: the network contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. 3) ***CondMat Dataset***[2]: the collaboration network is from the e-print arXiv and covers scientific collaborations between authors of papers submitted to Condense Matter category. For Wikivote and CondMat datasets, we simulate the true subgraphs of size $1,000$ using random walk, and the node attribute in true subgraphs follows distribution $\mathbb{N}(5, 1)$, otherwise $\mathbb{N}(0, 1)$. 4) ***DBLP***[3]: the collaboration graph of authors of scientific papers from DBLP computer science bibliography. An edge between two authors represents a common publication, and node attribute is the number of publications. We extract a subset of the dataset ranging from year 1995 to 2005. We apply random walk to get subgraphs with size 20,000 and inject the anomalies as our true subgraph as suggested by [14]. Statistics of all datasets are provided in Table IV.



(a) Run Time vs Nodes (b) Run Time vs Edges

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

TABLE IV: Statistics of Datasets for the 2nd Application.

Datasets	Statistics			
	Node	Edge	Blocks	Processors
SynNode	1,000∼10,000	3,000∼30,000	10	10
SynEdge	100,000	300,000∼1,000,000	100	50
Beijing	59,000	70,317	100	50
Wikivote	7,115	103,689	10	10
CondMat	23,133	93,497	100	50
DBLP	329,404	1,082,106	100	50

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 is infeasible to tune the parameters. EventTree and LTSS are scalable, but their performances are not as good as our method.

[2] https://snap.stanford.edu/data/

[3] http://konect.uni-koblenz.de/networks/dblp_coauthor

TABLE V: Results on Beijing, Wikivote, CondMat and DBLP datasets. The run time is measured in seconds.

Method	Beijing				Wikivote				CondMat				DBLP			
	Precision	Recall	F-measure	Run Time	Precision	Recall	F-measure	Run Time	Precision	Recall	F-measure	Run Time	Precision	Recall	F-measure	Run Time
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**
GBGP(Serial)	0.9166	0.7286	**0.8057**	843.37	0.8287	0.9908	0.90254	610.54	0.9132	0.9859	0.9479	1243.71	0.4701	0.9672	**0.6354**	13497.50
GBGP(Parallel)	0.9105	0.7283	0.8028	154.12	0.9637	0.9888	**0.9761**	171.98	0.9423	0.9835	0.9624	113.08	0.4683	0.9672	0.6311	567.20

e) Scalability Analysis: We evaluate the scalability of different methods in terms of the sizes of nodes and edges. Figure 3 reports the run time of our methods compared with the baseline methods. In order to run our algorithm, we partition the static network into multiple blocks with METIS [19], and run the parallel algorithm with multiple processors. Our method is able to get comparable performance as those customized algorithms of this specific problem, and it is more scalable if we properly utilize the computing resource based on network properties.

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 EventTree [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 non-convex 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 2016YFB1000901, the National Natural Science Foundation of China under grant 91746209, the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education of China under grant IRT17R32, and the US National Science Foundation under grants IIS-1815696 and IIS-1750911.

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. Rozenshtein, 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, multiscale, and multiplex networks," *Science*, vol. 328, no. 5980, pp. 876–878, 2010.

[4] C. Hegde, P. Indyk, and L. Schmidt, "A nearly-linear time framework for graph-structured sparsity," in *International Conference on Machine Learning*, 2015, pp. 928–937.

[5] O. Fercoq and P. Richtárik, "Accelerated, parallel, and proximal coordinate descent," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 1997–2023, 2015.

[6] F. Chen and B. Zhou, "A generalized matching pursuit approach for graph-structured sparsity," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, pp. 1389–1395.

[7] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Mathematical Programming*, vol. 117, no. 1-2, pp. 387–423, 2009.

[8] H.-J. M. Shi, S. Tu, Y. Xu, and W. Yin, "A primer on coordinate descent algorithms," *arXiv preprint arXiv:1610.00040*, 2016.

[9] X. Yuan, P. Li, and T. Zhang, "Gradient hard thresholding pursuit for sparsity-constrained optimization," in *International Conference on Machine Learning*, 2014, pp. 127–135.

[10] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[11] A. Ostfeld, J. G. Uber, E. Salomons, J. W. Berry, W. E. Hart, C. A. Phillips, J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan *et al.*, "The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 556–568, 2008.

[12] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu, "Inferring gas consumption and pollution emission of vehicles throughout a city," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 1027–1036.

[13] P. Bogdanov, M. Mongiovì, and A. K. Singh, "Mining heavy subgraphs in time-evolving networks," in *2011 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2011, pp. 81–90.

[14] M. Mongiovi, P. Bogdanov, R. Ranca, E. E. Papalexakis, C. Faloutsos, and A. K. Singh, "Netspot: Spotting significant anomalous regions on dynamic networks," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 28–36.

[15] 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 *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17, 2017, pp. 1631–1639.

[16] B. Zhou and F. Chen, "Graph-structured sparse optimization for connected subgraph detection," in *2016 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 709–718.

[17] S. Speakman, Y. Zhang, and D. B. Neill, "Dynamic pattern detection with temporal consistency and connectivity constraints," in *2013 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2013, pp. 697–706.

[18] D. B. Neill, "Fast subset scan for spatial pattern detection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 2, pp. 337–360, 2012.

[19] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

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