Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)

Query-Driven Discovery of Anomalous Subgraphs in Attributed Graphs

Nannan Wu*, **Feng Chen**†, **Jianxin Li***, **Jinpeng Huai***, **Bo Li***
*Dept. of Computer Science & Engineering, Beihang University, Beijing 100191, China
†Dept. of Computer Science, University at Albany, SUNY, Albany, NY 12203
{wunannan, lijx, libo}@act.buaa.edu.cn, fchen5@albany.edu, huaijp@buaa.edu.cn

Abstract

For a detection problem, a user often has some prior knowledge about the *structure-specific* subgraphs of interest, but few traditional approaches are capable of employing this knowledge. The main technical challenge is that few approaches can efficiently model the space of *connected subgraphs* that are isomorphic to a query graph. We present a novel, efficient approach for optimizing a generic nonlinear cost function subject to a query-specific structural constraint. Our approach enjoys strong theoretical guarantees on the convergence of a nearly optimal solution and a low time complexity. For the case study, we specialize the nonlinear function to several well-known graph scan statistics for anomalous subgraph discovery. Empirical evidence demonstrates that our method is superior to state-of-the-art methods in several real-world anomaly detection tasks.

1 Introduction

In recent years, graph-structure optimization for anomalous subgraph discovery in high-dimensional data and graphs as an open problem has attracted much attention [Qian *et al.*, 2014; Wu *et al.*, 2016; Chen and Zhou, 2016; Chen and Neill, 2014; Tong *et al.*, 2007; McFowland *et al.*, 2013; Gionis *et al.*, 2015]. In many settings, anomaly in high-dimensional data presents a *structure-specific* shape, such as the *cholera* epidemic infection along a river [Patil *et al.*, 2003]. To motivate this scenario, consider the *Botnet Infection* problem [Choi *et al.*, 2009] as shown in Figure 1. *Group activity* presented in a "*specific*" shape is an inherent property of bots attacking on networks [Choi *et al.*, 2009]. Therefore, we employ the prior knowledge about the structure of cyber attacks as the side information regularizing the generic nonlinear function optimization in attributed graphs as follows.

$$\min_{\mathbf{x}\in\mathbb{R}^n} \varphi(\mathbf{x}) \qquad s.t.\ supp(\mathbf{x}) \subseteq \mathcal{M} \tag{1}$$

Given the graph $\mathbb{G} = (V, E)$ where $V = [n] = \{1, \cdots, n\}$ and $E \subseteq V \times V$, based on attributes (e.g., "*transfer rate*") for each vertex $v \in V$, the differentiable function φ is formulated as $\mathbb{R}^n \to \mathbb{R}$. The support set of $\mathbf{x}$ is $supp(\mathbf{x}) = \{i|\mathbf{x}_i \neq 0\}$.

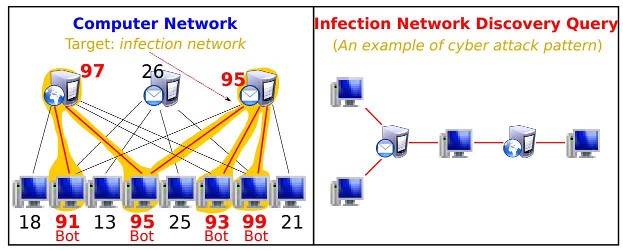


Figure 1: Graph-structured optimization for the botnet infection network discovery within the "*transfer rates*" attributed network under the structure-specific query constraint.

The domain for Problem (1) is the *structure-specific* model $\mathcal{M} = \{Q_1, \cdots, Q_K\}$ where $Q_i \subseteq [n]$ is derived from $\mathbb{G}$ based on the "***query graph***". The model $\mathcal{M}$ is a family of structured support sets (e.g., its graphs are *stars* or *trees*).

Related work. Recent works solving Problem (1) fall into two main categories: 1) **Shape induced.** By encoding *specific-shaped* models (e.g., connectivity) [Bach *et al.*, 2011] as structured sparsity-inducing norms, the methods can reformulate Problem (1) as a convex (or non-convex) optimization problem: $\min_{\mathbf{x}\in\mathbb{R}^n} \varphi(\mathbf{x}) + \lambda \cdot \Omega(\mathbf{x})$ where λ is a trade-off parameter and $\Omega(\mathbf{x})$ is a structured sparsity-inducing norm of $\mathcal{M}$ that is typically non-smooth and non-Euclidean. These methods just focus on a connecting property without the *specific-shape*. 2) **Model-projection based.** The methods depend on a projection oracle of $\mathcal{M}$: $\mathcal{P}(\mathbf{b}) = \arg\min_{\mathbf{x}\in\mathbb{R}^n} ||\mathbf{x} - \mathbf{b}||_2^2$ subject to $supp(\mathbf{x}) \in \mathcal{M}$, and decompose Problem (1) into two subproblems: minimizing the unconstrained $\varphi(\mathbf{x})$ and the projection oracle $\mathcal{P}(\mathbf{b})$. There are many methods to solve Problem (1) under the assumption that $\mathcal{P}(\mathbf{b})$ can be addressed exactly [Bahmani *et al.*, 2013; Jain *et al.*, 2014; Yuan *et al.*, 2014; Bahmani *et al.*, 2016; 2013; Yuan and Liu, 2014; Hegde *et al.*, 2015]. These methods are only applicable to quadratic functions subject to linear constraints. However, few works have investigated combinations of *generic nonlinear functions* and *specific-shaped constraints*.

We present an approach for optimizing generic nonlinear cost functions subject to *structure-specific* constraints, where the constraints are formalized to *query graphs*. Then the *output subgraph* is ***isomorphic*** to the *query graph*. By a proper choice of the cost functions $\varphi(\cdot)$, our approach can be im-

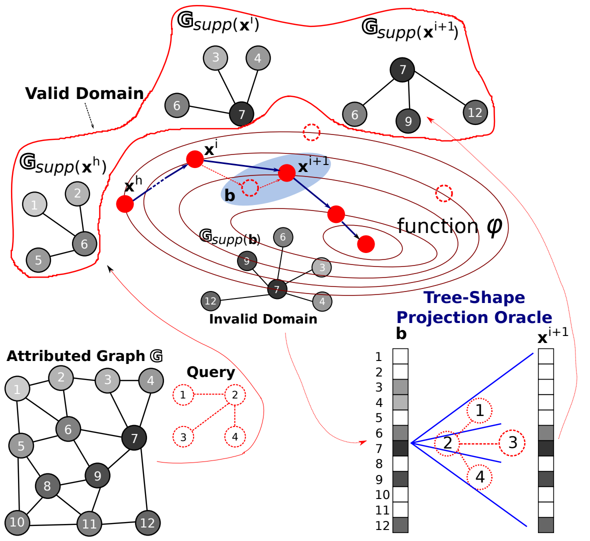


Figure 2: An illustration of our work. The attributed graph with n vertices labeled $1, \cdots, n$, is described by the underlying graph $\mathbb{G} = (V, E)$ and the attribute matrix $\mathbf{W}$ (i.e., the darker color denotes the larger value for each vertex). The variables $\mathbf{x}, \mathbf{b}, \mathbf{g} \in \mathbb{R}^n$ denote the coefficients of vectors defined by $\mathbb{G}$ (i.e., $V = [n]$); for example, $\mathbf{g}$ is called the gradient vector.

plemented easily in different types of attributed graphs, such as *water sensor* and *emergency department* networks. Main contributions of this study are summarized as follows:

- **Design of an efficient structure-specific subgraph discovery algorithm.** A new algorithm is proposed to approximately solve Problem (1) where φ is a differentiable nonlinear cost function. Over the *structure-specific* model $\mathcal{M}$, our proposed algorithm is required to minimize $\varphi(\mathbf{x})$ over a projected subspace supported by the tree-shape projection oracle.
- **Theoretical guarantees.** The convergence rate and accuracy of our proposed algorithm are analyzed under the *Weak Restricted Strong Convexity* condition, which is more general than the popular Restricted Strong Convexity (RSC) condition. We prove that our proposed algorithm enjoys rigorous theoretical guarantees.
- **Compressive experiments to validate our approach from the effectiveness and efficiency perspectives.** Our proposed algorithm is suitable for optimization of a variety of graph scan statistics as the target for anomaly *specific-shaped* subgraph detection. Extensive experiments on a number of benchmark datasets demonstrate that the algorithm performs better than the representative methods for this task on both accuracy and run time.

2 Problem Formulation

Before the problem formulation for the anomalous subgraph discovery in attributed graphs, we illustrate the three key concepts by a case of *anomalous subgraph discovery* in Figure 2.

Attributed Graph. A network (e.g., *computer network*) can be modeled as an *attributed graph* composed of the underlying graph $\mathbb{G} = (V, E)$ and attribute matrix $\mathbf{W} \in \mathbb{R}^{n \times p}$,

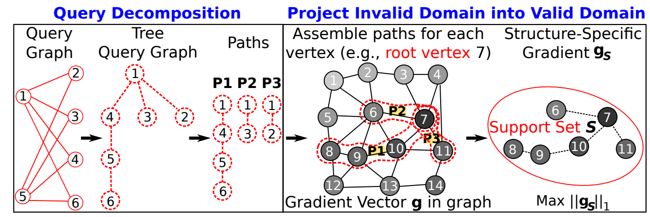


Figure 3: Tree-Shape Projection Oracle. First, we obtain the tree approximation in the query graph, and then search maximum weight paths in gradient or solution graphs, assemble paths to a tree that is isomorphic to the tree query graph.

where n, p denote the number of vertices and attributes, $V = [n]$ and $E \subseteq V \times V$, (e.g., $\mathbf{W}_{v,0}$ denotes the "*transfer rate*" of the vertex v at the attribute "0").

Query Graph. Given a *query* $\mathbb{Q} = (V, E)$, we focus on the query having a structure-specific shape (e.g., *tree* shape).

Match Model. A graph $\mathbb{C}$ is a subgraph of $\mathbb{G}$, denoted as $\mathbb{C} \subseteq \mathbb{G}$, if $V_\mathbb{C} \subseteq V_\mathbb{G}$, $E_\mathbb{C} \subseteq E_\mathbb{G}$ and $\forall (u, v) \in E_\mathbb{C}$, $u, v \in V_\mathbb{C}$. A graph $\mathbb{C}$ is isomorphic to a query graph $\mathbb{Q}$, denoted as $\mathbb{C} \cong \mathbb{Q}$, if there is a bijection $\psi : V_\mathbb{C} \to V_\mathbb{Q}$ such that, for every pair of vertices $u, v \in V_\mathbb{C}$, $(u, v) \in E_\mathbb{C}$ if and only if $(\psi(u), \psi(v)) \in E_\mathbb{Q}$. Given the underlying graph $\mathbb{G}$ and the query graph $\mathbb{Q}$, we present the *structure-specific* domain

$$\mathcal{M}(\mathbb{Q}) := \{V_\mathbb{C} \mid \mathbb{C} \subseteq \mathbb{G}, \mathbb{C} \cong \mathbb{Q}\} \tag{2}$$

The valid domain $\mathcal{M}(\mathbb{Q})$ illustrates all vertex subsets of $\mathbb{G}$ whose shapes are same to the query graph $\mathbb{Q}$.

Anomalous Structure-Specific Subgraph Discovery. We aim to obtain $S \in \mathcal{M}(\mathbb{Q})$ for minimizing the cost function over the domain. Given $\mathbb{G}$, we assume that $V_\mathbb{G} = \{1, \cdots, n\}$. The vector form of $V_\mathbb{G}$ is denoted as $\mathbf{x} \in \mathbb{R}^n$, and the support set of $\mathbf{x}$ in $V_\mathbb{G}$ is denoted as $supp(\mathbf{x}) = \{i \mid \mathbf{x}_i \neq 0\}$. Thus this problem is formulated as

$$\min_{\mathbf{x} \in \mathbb{R}^n} \varphi(\mathbf{x}) \quad s.t.\ supp(\mathbf{x}) \in \mathcal{M}(\mathbb{Q}) \tag{3}$$

where $\varphi(\cdot)$ is a differentiable nonlinear cost function modeled from the attribute matrix $\mathbf{W}$. Figure 2 illustrates our problem through an attributed computed network and a query graph. There exist multiple matchings of the *query* in the network.

3 Preliminary: Tree-Shape Projection Oracle

For Problem (3), the gradient or solution vector of the function may fall into the invalid domain. We present the work on projecting a vector variable into the valid domain.

Given the *query graph* $\mathbb{Q}$, the underlying graph $\mathbb{G}$, the match model $\mathcal{M}(\mathbb{Q})$, and the vector $\mathbf{g}$, the projection oracle, $\mathcal{P} : \mathbb{R}^n \to V_\mathbb{G}$, is defined as

$$\mathcal{P}(\mathbf{g}) = \arg\max ||\mathbf{g}_S||_1 \qquad s.t.\ S \in \mathcal{M}(\mathbb{Q}) \tag{4}$$

where $\mathbf{g}_S$ denotes the restriction of $\mathbf{g}$ to indices S so that $(\mathbf{g}_S)_i = \mathbf{g}_i$ for $i \in S$ and $(\mathbf{g}_S)_i = 0$ otherwise. There are many works about solving Problem (4) with high time complexity. For a general query graph, we present a simple oracle to Problem (4) in Figure 3. This oracle consists of two main steps, ***query decomposition*** and ***projection***.

Query decomposition. Wu et al., looked at approximating arbitrary query graph metrics by a tree shaped prior (e.g., *spanning tree*) [Wu *et al.*, 2016]. By a prior, we generate a spanning tree of the query graph. From the root of the tree, we collect a set of query paths terminated with leaf nodes.

Projection. Each vertex $v \in V_{\mathbb{G}}$ is assigned with the value $\mathbf{g}_v$. We search the maximum-weight paths from a root vertex v in $\mathbb{G}$ for all of query paths, on condition that the assembled paths are isomorphic to the tree of the query graph. The condition is easily satisfied by caring for "*share nodes*" (i.e., *a share node exists in multiple query paths*). Within all paths for $v \in V_{\mathbb{G}}$, we select the maximum weight assembled paths as the target structure-specific graph, whose set of nodes is S.

4 Methodology

We propose a new graph-structured optimization method for nonlinear cost functions and discuss its theoretical properties.

4.1 Graph Tree-shape Projection Pursuit Algorithm (Graph-TPP)

The algorithm described here iterates on the solution vector for Problem (3). For each iteration, we first obtain the most significant support set within the valid domain for the gradient of the cost function at the current solution. Then we refine the solution over the significant support set of gradient and the previous solution. Graph-TPP is illustrated in Algorithm 1 in detail, where each iteration consists of six steps.

Algorithm 1: Graph-TPP

Input: Attribute $\mathbf{W}$, underlying graph $\mathbb{G}$, query graph $\mathbb{Q}$ and step size η (1 by default)
Result: Vertex set of *structure-specific* subgraph S

1 Set $i = 0, \mathbf{x}^i = \mathbf{0}$;
2 **repeat**
3 $\quad \Gamma = \mathcal{P}(\nabla\varphi(\mathbf{x}^i))$; $\triangleright$ *Projection*
4 $\quad \Omega = supp(\mathbf{x}^i - \eta\nabla_{\Gamma}\varphi(\mathbf{x}^i))$; $\triangleright$ *Support*
5 $\quad \mathbf{b} = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \varphi(\mathbf{x}) \;\; s.t. \; \mathbf{x}_{\Omega^c} = 0$; $\triangleright$ *Estimate*
6 $\quad S = \mathcal{P}(\mathbf{b})$; $\triangleright$ *Projection*
7 $\quad \mathbf{x}^{i+1} = \mathbf{b}_S$; $\triangleright$ *Prune*
8 $\quad i \leftarrow i + 1$;
9 **until** *halting condition holds*;
10 **return** S;

Projection. For a gradient $\nabla\varphi(\mathbf{x}^i)$ or a solution vector $\mathbf{b}$, we identify the most significant support set (i.e., the set maximizing Problem (4)) within the valid domain.

Support. First we calculate the projected gradient descent at the current $\mathbf{x}^i$ with step-size η (1 by default). Then the support set Ω is obtained and can be interpreted as the subspace where the nonconvex set $\{\mathbf{x} \mid supp(\mathbf{x}) \subseteq \Omega\}$ is located.

Estimate. Over the support set Ω, let $\Omega^c = V_{\mathbb{G}} \setminus \Omega$, the function φ is minimized to make an intermediate estimate $\mathbf{b}$.

Prune. This step (Line 7) calculates the next solution vector $\mathbf{x}^{i+1}$: $\mathbf{x}^{i+1} = \mathbf{b}_S$, which retains the most significant structured-entries in $\mathbf{b}$.

We have two popular options for defining the halting criterion: (1) the difference between the score functions is less than a threshold $|\varphi(\mathbf{x}^i) - \varphi(\mathbf{x}^{i+1})| < \epsilon$; and (2) the difference between the vectors is less than a threshold $||\mathbf{x}^i - \mathbf{x}^{i+1}||_2 < \epsilon$ (e.g., $\epsilon = 0.01$).

Discussion: The projection problem is hard to solve due to the NP-hard subgraph isomorphism problem. For connected subgraph detection, work [Wu *et al.*, 2016] yielded compelling results by tree shape priors (e.g., spanning tree). For a general query graph, we employ the tree-shape projection oracle to perform graph matching, which is different from the previous work that has *head* and *tail* projections [Chen and Zhou, 2016]. We focus on the *isomorphic* subgraph discovery; however, the work [Chen and Zhou, 2016] focuses on the *connected* subgraph discovery.

4.2 Theoretical Analysis

The theoretical properties of Graph-TPP are examined in two aspects: Studying convergence rate, and time complexity. Before obtaining the theoretical results, we require the following key technical condition, under which the results are guaranteed. Let $s = |V_{\mathbb{Q}}|$.

Definition 1 (Weak Restricted Strong Convexity Property (WRSC)). Let $\mathcal{M} = \{S \mid S \subset [n], |S| \leq 2s\}$, and there must be $S \in \mathcal{M}$ if $S \in \mathcal{M}(\mathbb{Q})$. The differentiable function φ has the condition $(\xi, \delta, \mathcal{M})$-*WRSC* if $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\forall S \in \mathcal{M}$ with $supp(\mathbf{x}) \cup supp(\mathbf{y}) \subseteq S$, for some $\xi > 0$ and $0 < \delta < 1$.

$$||\mathbf{x} - \mathbf{y} - \xi\nabla_S\varphi(\mathbf{x}) + \xi\nabla_S\varphi(\mathbf{y})||_2 \leq \delta||\mathbf{x} - \mathbf{y}||_2 \quad (5)$$

Theorem 1 (**Graph-TPP Convergence**). *Given an attributed graph* $\mathbf{W}$, $\mathbb{G}$ *and a query graph* $\mathbb{Q}$*, we obtain the domain* $\mathcal{M}$*. Consider the differentiable cost function* $\varphi : \mathbb{R}^n \to \mathbb{R}$ *that satisfies the condition* $(\xi, \delta, \mathcal{M})$*-WRSC. Let* $\mathbf{x}^i \in \mathbb{R}^n$*. Then for any true* $\mathbf{x}^* \in \mathbb{R}^n$ *and* $supp(\mathbf{x}^*) \in \mathcal{M}(\mathbb{Q})$*, the iteration of Algorithm 1 holds*

$$||\mathbf{x}^{i+1} - \mathbf{x}^*||_2 \leq \alpha||\mathbf{x}^i - \mathbf{x}^*||_2 + \beta||\nabla_I\varphi(\mathbf{x}^*)||_2 \quad (6)$$

where $\alpha = \frac{2\sqrt{2}}{1-\delta}\Big(2\sqrt{\delta - \delta^2} + (2 - \frac{\eta}{\xi})\delta + 1 - \frac{\eta}{\xi}\Big)$, $\beta = \frac{2}{1-\delta}\Big(\xi + \frac{4\sqrt{2}(1-\delta)\xi}{1-2\delta} + \frac{\sqrt{2}(1-2\delta)\xi}{\sqrt{\delta-\delta^2}} + 2(1 - \sqrt{2}\eta)\Big)$, $\mathbf{x}^*$ *is the optimal solution, and* $I = \arg\max_{S\in\mathcal{M}} ||\nabla_S\varphi(\mathbf{x}^*)||_2$*. The shrinkage rate* $\alpha < 1$ *controls the convergence of Graph-TPP. (Proof: see Appendix)*

The estimation error of Graph-TPP depends on the multipliers of $||\nabla_S\varphi(\mathbf{x}^*)||_2$. Before reaching the estimation error level, Graph-TPP geometrically converges to the near-optimal $\mathbf{x}^*$. Especially, when $||\nabla_I\varphi(\mathbf{x}^*)||_2 = 0$, Graph-TPP guarantees that the true $\mathbf{x}^*$ is obtained within finite iterations.

Theorem 2 (**Graph-TPP Time Cost**). *Given* $\alpha < 1$, $supp(\mathbf{x}^*) \in \mathcal{M}(\mathbb{Q})$ *and the function* φ *satisfies the WRSC condition, Graph-TPP in Algorithm 1 returns a* $\hat{\mathbf{x}}$ *such that* $||\mathbf{x}^* - \hat{\mathbf{x}}||_2 \leq (1 + \frac{\beta}{1-\alpha})||\nabla_I\varphi(\mathbf{x}^*)||_2$*. The estimation error is less than a constant. Graph-TPP runs in time*

$$O\big(T \log(||\mathbf{x}^*||_2/||\nabla_I\varphi(\mathbf{x}^*)||_2)\big) \quad (7)$$

where T *is the time cost of execution for one iteration in Algorithm 1. If* T *can scale linearly with* n*, then Graph-TPP scales nearly linearly with* n*. (Proof: see Appendix)*

4.3 Application to Graph Scan Statistics

We specialize φ to be a number of graph scan statistics that are widely employed in pattern detection in graphs.

Graph Scan Statistic. First select a graph $\mathbb{C} \subseteq \mathbb{G}$ and $\mathbb{C} \cong \mathbb{Q}$. Then a graph scan statistic is defined as $F(V_{\mathbb{C}}) = \log \frac{Prob(data|H_1(V_{\mathbb{C}}))}{Prob(data|H_0)}$ that is the log-likelihood ratio statistic over all *matching subgraphs* of $\mathbb{G}$. For the null hypothesis H_0, we assume no *anomaly structure-specific subgraphs* (i.e., all the *observed data* are generated from the expected distribution). However, for the alternative hypothesis $H_1(V_{\mathbb{C}})$, we assume the *observed data* in the subgraph $\mathbb{C}$ show a significant increase in, such as the *number of visiting patients*, for some multiplicative factors. At last the detection problem is formulated as: $\min_{\mathbb{C}\subseteq\mathbb{G}} -F(V_{\mathbb{C}})\ s.t.\ V_{\mathbb{C}} \in \mathcal{M}(\mathbb{Q})$.

Let $\varphi(\mathbf{x}) = -F(V_{\mathbb{C}})$. Denote the vector form of $V_{\mathbb{C}}$ as $\mathbf{x} \in \{0,1\}^n$, such that $supp(\mathbf{x}) = V_{\mathbb{C}}$ where $\mathbf{x}$ can be relaxed to $\mathbf{x} \in [0,1]^n$ [Chen and Zhou, 2016]. We mainly examine φ in Equation (8) to log forms of **expectation-based Poisson** (EBP) and **Kulldorff's** (KULL) graph scan statistics. Let $\mathbf{c} = \mathbf{W}_{:,0}$ denote the "observed count" attribute values. Similarly, let $\mathbf{d} = \mathbf{W}_{:,1}$ denote the "expected count" attribute values.

$$\begin{aligned} \varphi_{EBP}(\mathbf{x}) &= -\mathbf{x}^T\mathbf{c}\log\left(\mathbf{x}^T\mathbf{c}\ /\ \mathbf{x}^T\mathbf{d}\right) - \mathbf{x}^T\mathbf{d} + \mathbf{x}^T\mathbf{c} \\ \varphi_{KULL}(\mathbf{x}) &= -\mathbf{x}^T\mathbf{c}\log\left(\mathbf{x}^T\mathbf{c}\ /\ \mathbf{x}^T\mathbf{d}\right) - (1-\mathbf{x})^T\mathbf{c} \\ &\quad \cdot \log\left((1-\mathbf{x})^T\mathbf{c}\ /\ (1-\mathbf{x})^T\mathbf{d}\right) \end{aligned} \tag{8}$$

where $\mathbf{x} \in [0,1]^n$. Let $\hat{r} = \min\{\mathbf{d}_i/\mathbf{c}_i\}$ for $i \in V_{\mathbf{G}}$. By [Yuan and Liu, 2014], φ_{EBP} satisfies the WRSC condition that $\delta = \sqrt{1 - 2\xi(1-\hat{r}^2) + \xi^2}$ for $\xi < 2(1-\hat{r}^2)$.

5 Experiments

We compare the effectiveness and efficiency of our method Graph-TPP with respect to competitive methods.

5.1 Experimental Design

1) ***Respiratory Emergency Department* (ED) Dataset.** Given a grid network which consists of 10,000 nodes and 14,850 edges, we consider each node as a zip code. For each node $v \in V$, we collected the T day period of respiratory ED visit data $D_v \in \mathbb{R}^T$ (e.g., $T = 28$), where especially the time $t = 0$ denotes the current day. During non-outbreak period, the number of patients visiting ED in v is simulated with the Poisson distribution [Neill, 2009], i.e., $D_v^t \leftarrow \text{Poisson}(\mu)$ for $t = 0, \cdots, T$ where μ denotes the expected number of patients visiting ED in v on those days and μ is randomly selected from $\{1, \cdots, 34\}$. We randomly select the outbreak duration of U days from $\{1, \cdots, 7\}$. On each day $t \in \{0, \cdots, U\}$ of the outbreak, we inject $\text{Poisson}((T-t)w_v\Delta)$ cases into each infected node v where $w_v \propto \sum_t D_v^t$ is normalized so that the total weight equals to 1 in infected zips and Δ denotes the outbreak severity (e.g., Δ=800), i.e., $D_v^t \leftarrow D_v^t + \text{Poisson}((T-t)w_v\Delta)$ for $t = 0, \cdots, U$. We performed a simulation for this dataset with *medium-size* injects affecting 10 percent nodes, which consist of the ***ground-truth connected target*** subgraph [Neill, 2009]. We simulated 500 graph snapshots. For testing the robustness of methods to noise, we randomly selected $K \in \{2,4,6,8,10\}$ percent nodes in this network, and flipped their values (i.e., no inject outbreak cases if the nodes are infected, inject outbreak cases otherwise). Let $\mathbf{c}_v = D_v^0$ and $\mathbf{d}_v = \frac{1}{T}\sum_{t=1}^T D_v^t$. 2) **Water Pollution Data.** These data were collected on the real-world network of 12,527 nodes and 14,831 edges, and there were four nodes with chemical contaminant plumes, distributed in four different areas [Chen and Zhou, 2016]. "The spreads of contaminant plumes were simulated using the water network simulator EPANET for 8 hours" [Chen and Zhou, 2016]. For each hour, the value at each node v was reported with the corresponding sensor, $D_v \leftarrow 1$ if it is polluted and $D_v \leftarrow 0$ otherwise. We randomly selected K percent nodes, and flipped their sensor binary values for testing the robustness of methods to noises, where $K \in \{2,4,6,8,10\}$. We considered the assigned attributed graphs as $\mathbf{c}_v \leftarrow D_v$ (i.e., the report of the sensor at the node v) and $\mathbf{d}_v \leftarrow K\%$ (i.e., the noise ratio) [Chen and Zhou, 2016]. 3) **Real-World Network Dataset.** An Internet company[1] provided a traffic network of 131,107 nodes and 358,386 edges derived from $*$***edu.cn*** web sites browsing logs. The total 3,978,073 logs were collected from May 31, 2014 to May 13, 2015. For a day t during this period and a node v in this network, we denoted the number of logs within v on that day t as the *observed value*, and the average number of logs within v before t as the *expected value*. We have a graph snapshot for each day.

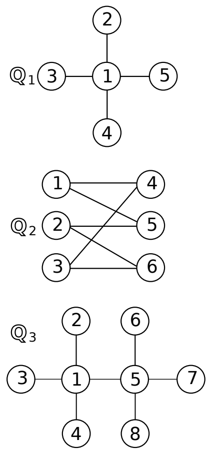
Comparison Methods. We considered two state-of-the-art methods: *Top-K* [Gupta *et al.*, 2014] and *Fast-K* [Yang *et al.*, 2016]. The main idea behind the methods is as follows: Given a query graph $\mathbb{Q}$ and an attributed graph $\mathbb{G}$, the goal is to find top k subgraphs that are matched to $\mathbb{Q}$ and have the highest ranking scores by a ranking function (e.g., summing vertex weights for *Fast-K* and edge weights for *Top-K*). From the k subgraphs, we selected the top one subgraph as the anomalous subgraph. By author recommendations, set $k = 10$ for *Top-K*, and $k = 20$ for *Fast-K*. Let the vertex weight $w(v) \leftarrow \mathbf{W}_{v,0}$. As the *Top-k* method is applied to edge weights, we replicated each node v to v' and added edge (v, v') to $\mathbb{G}$. We set the weight of edge (v, v') to $w(v)$ for a pair of original node and replicate node, and set the weight of edge (v, v') to 0 otherwise. For *Top-k* method, return the subgraph by removing replicate nodes and corresponding edges.

Performance Metrics. 1) Effectiveness: *Precision*. We compute the precision of the target subgraph (i.e., the ratio of the number of correct anomalous nodes and the number of nodes). As the size of returned target subgraph is fixed, we ignore the recall metric. **2) Efficiency:** *Graph Scan Statistic Score* and *Running Time*. The optimization power of our method is analyzed through the scores and running times. **3) Case Study.** We present some cases to illustrate our method.

5.2 Effectiveness

As the size of detected subgraph is fixed for the graph isomorphism constraint, we focus on the precision of methods in Figure 4. By evaluating methods to three *typical query* graphs in the left sub-figure of Figure 4 [Du and Yang, 2011; Yang *et al.*, 2014; Kim *et al.*, 2015; Pan *et al.*, 2013], we

[1] An Internet security company in China with more than 0.6 billion users.



Methods	Emergency Department Data				Water Pollution Data			
	$\mathbb{Q}_1$	$\mathbb{Q}_2$	$\mathbb{Q}_3$	Run Time	$\mathbb{Q}_1$	$\mathbb{Q}_2$	$\mathbb{Q}_3$	Run Time
φ_{EBP} (2%)	0.96	**1.00**	0.97	107.09s	**1.00**	0.97	**0.96**	30.48s
φ_{EBP} (4%)	**0.90**	**0.98**	0.88	102.76s	1.00	**1.00**	**0.98**	34.75s
φ_{EBP} (6%)	0.79	0.91	**0.98**	95.00s	0.97	**0.94**	**0.88**	34.03s
φ_{EBP} (8%)	**0.80**	**0.94**	0.86	100.32s	**1.00**	1.00	**0.94**	28.14s
φ_{EBP} (10%)	**0.68**	**0.84**	0.75	94.79s	1.00	0.86	**0.91**	30.84s
φ_{KULL} (2%)	**1.00**	0.98	**0.98**	92.62s	1.00	0.97	0.89	28.11s
φ_{KULL} (4%)	0.88	0.94	**0.95**	102.52s	1.00	1.00	0.98	29.65s
φ_{KULL} (6%)	**0.80**	**0.95**	0.88	96.27s	0.97	0.94	0.88	31.60s
φ_{KULL} (8%)	0.76	0.77	**0.89**	99.62s	1.00	1.00	0.77	31.96s
φ_{KULL} (10%)	0.58	0.63	**0.76**	80.45s	1.00	0.86	0.91	35.93s
Fast-K (2%)	0.83	0.53	0.90	**30.97**s	0.96	**1.00**	0.83	25.60s
Fast-K (4%)	0.72	0.53	0.73	32.15s	1.00	0.97	0.83	25.86s
Fast-K (6%)	0.53	0.56	0.66	33.85s	**1.00**	0.80	0.81	**20.66**s
Fast-K (8%)	0.53	0.42	0.55	34.22s	0.97	1.00	0.75	25.61s
Fast-K (10%)	0.43	0.24	0.41	32.51s	1.00	**1.00**	0.75	25.08s
Top-K (2%)	0.33	0.60	0.00	680.61s	1.00	0.81	0.85	139.87s
Top-K (4%)	0.42	0.50	0.00	661.55s	1.00	0.79	0.87	183.92s
Top-K (6%)	0.35	0.25	0.00	701.34s	1.00	0.83	0.87	214.27s
Top-K (8%)	0.21	0.28	0.00	640.08s	1.00	0.81	0.79	215.16s
Top-K (10%)	0.30	0.50	0.00	696.94s	1.00	0.66	0.87	282.82s

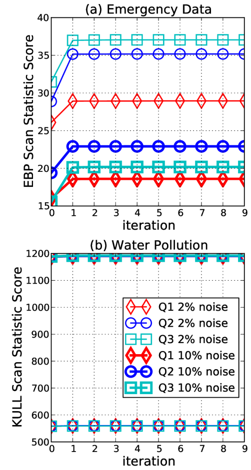


Figure 4: Query graphs $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$. Comparison on *precision* of structure-specific anomalous subgraphs discovered by methods, run times and graph scan statistic scores, (e.g., 2% refers to the noise level).

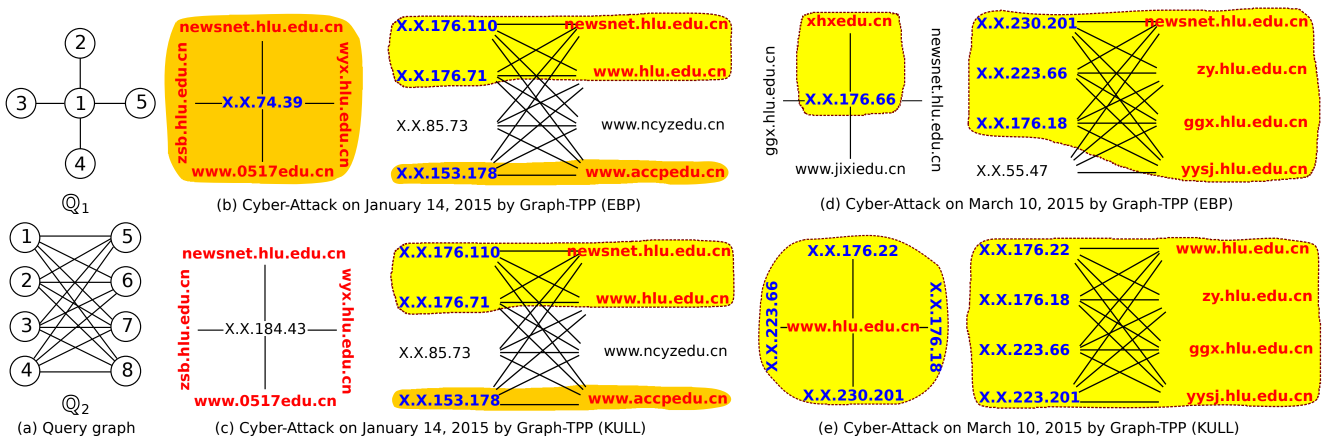


Figure 5: Given the query graphs $\mathbb{Q}_1$ and $\mathbb{Q}_2$, we present the *structure-specific cyber-attack* detection networks, on January 14, 2015 and March 10, 2015, in the $*$*.edu.cn* networks. The darker yellow area represents the *Dedecms Attack*, and the lighter yellow area represents the *FckEditor Attack*. The bold text represents an attack source or an attack object.

can observe precision in the middle sub-figure of Figure 4. The query graph $\mathbb{Q}_1$ shows the *star-shape*, which means that a node infects its neighbors, and $\mathbb{Q}_2$ denotes a *many-infect-many* case. The query graph $\mathbb{Q}_3$ denotes the *infection* from one *star-shape* area to another neighbor *star-shape* area.

In Figure 4, we present a comparison of precision for methods in these two data sets in detail. For the *emergency data*, at the 2 percent noise level, our proposed Graph-TPP (i.e., φ_{EBP} and φ_{KULL}) achieved higher precision (*close to 1*) than competitive baselines (*close to 0.9*). However, even at 10 percent noise level, our methods achieved at least 0.58 precision, and baselines achieved the best precision to 0.5. Especially for $\mathbb{Q}_1$ and $\mathbb{Q}_3$, the precision of baselines decreases sharply to about 0.43 with the noise increasing, but the precision of φ_{EBP} is greater than 0.68. For the *water pollution data*, Graph-TPP and Fast-K perform similarly in $\mathbb{Q}_1$ and $\mathbb{Q}_2$, but perform better than Top-K. However, in $\mathbb{Q}_3$, the *worst* precision 0.88 for Graph-TPP (φ_{EBP}) is greater than the *best* precision 0.87 for baselines. Our methods outperformed all the baselines on precision in both of datasets.

5.3 Efficiency

Right sub-figures of Figure 4 depicts the scores of graph scan statistics (EBP and KULL) changing with the iteration. The score corresponds to the graph scan statistic $F(V_{\mathbb{C}})$. We can observe that our proposed Graph-TPP has converged at most five iterations. According to Theorem 1, our methods geometrically converge to the near-optimal stationary point.

The table of Figure 4 shows the time cost of all competitive methods on the two datasets. We can observe that the running times of Graph-TPP (EBP and KULL) are approximately equal in the two datasets. Although Fast-K performed well on the run time, it is a heuristic algorithm. Our methods enjoy a rigorous theoretical guarantee. In particular, for the *water pollution* data, the run time of Graph-TPP is close to the Fast-K method.

5.4 Case Study

We tested our methods Graph-TPP on the *real-world network dataset* for the star and bipartite shaped attack patterns.

Star-shaped attack case. From the left *star-shaped* subfigures of Figure 5(b-e), the left subfigure of Figure 5(b) is a ***Dedecms Attack***[2] network, and the left subfigure of Figure 5(e) is a ***FckEditor Attack***[3] network. Our methods successfully discovered the *cyber-attack* networks without innocent nodes. The difference between the two networks is that the client *X.X.74.39* attacked the other server sites $*$*.hlu.edu.cn* and *0517edu.cn*; however, the server site *www.hlu.edu.cn* was attacked by four clients. These *cyber-attack* patterns are the most common forms in networks. For the left subfigures of Figure 5(c, d), our method detected the attack client *X.X.176.66* and most of the attacked server sites.

Bipartite-shaped attack case. In Figure 5(b-e), the right subfigures are *bipartite-shaped cyber-attack* networks. In the right subfigures of Figure 5(b-c), Graph-TPP (EBP and KULL) detected the same *cyber-attack* network. In the right subfigures of Figure 5(d-e), Graph-TPP (KULL) detected the *cyber-attack* network without innocent nodes, and Graph-TPP (EBP) detected the *cyber-attack* network with one innocent node *X.X.55.47*. As the client *X.X.55.47* is replaced by *X.X.176.22*, a new server site *www.hlu.edu.cn* attacked by the four clients is discovered. For this case, Graph-TPP (KULL) performs better than Graph-TPP (EBP).

In Figure 5(b-e), the ***variations*** of *X.X.176.X* contributed to many *cyber-attacks*. The server site *www.ncyzedu.cn* is the innocent node in the right subfigures of Figure 5(b-c), and this server site is not successfully attacked for removing the common vulnerability. From Figure 5, we can observe that Graph-TPP (EBP and KULL) works well on different scenarios. In this paper, we present a feasible framework Graph-TPP to optimize a generic nonlinear function on the *structure-specific* query graph constraints.

6 Conclusions

We present an efficient algorithm to optimize a general nonlinear function on the *structure-specific* constraints. For future work, we will extend our work on large graphs, other similarity measures (e.g., exploiting domain semantics), and other types of attributes (e.g., categorical attributes).

Acknowledgments

The work is supported by China 973 Fundamental R&D Program (No.2014CB340300), NSFC program (No.61472022, 61421003), SKLSDE-2016ZX-11, and partly by the Beijing Advanced Innovation Center for Big Data and Brain Computing. The corresponding author is Jianxin Li.

A Proof of Theorem 1

Proof. Let $r^{i+1} = \mathbf{x}^{i+1} - \mathbf{x}^*$. Compute upper bound of the residual $||r^{i+1}||_2$.

$$\begin{aligned} ||r^{i+1}||_2 = ||\mathbf{x}^{i+1} - \mathbf{x}^*||_2 &\le ||\mathbf{x}^{i+1} - \mathbf{b}||_2 + ||\mathbf{b} - \mathbf{x}^*||_2 \\ &= ||\mathbf{b}_S - \mathbf{b}||_2 + ||\mathbf{b} - \mathbf{x}^*||_2 \\ &\le 2||\mathbf{x}^* - \mathbf{b}||_2 \end{aligned} \tag{9}$$

where as the support set of optimal $\mathbf{x}^*$ can be built from the space $\mathcal{M}(\mathbb{Q})$, $\mathbf{b}_S$ is restricted to the elements in $\mathbf{b}$ with the *maximal score*. Thus we have $||\mathbf{b}_S - \mathbf{b}||_2 \le ||\mathbf{b} - \mathbf{x}^*||_2$. Next we compute the upper bound of the component $||\mathbf{x}^*_\Omega - \mathbf{b}_\Omega||_2$, $||\mathbf{x}^*_\Omega - \mathbf{b}_\Omega||_2^2 =< \mathbf{x}^* - \mathbf{b}, \mathbf{x}^*_\Omega - \mathbf{b}_\Omega >$

$$=< \mathbf{x}^* - \mathbf{b} - \xi\nabla_\Omega\varphi(\mathbf{x}^*) + \xi\nabla_\Omega\varphi(\mathbf{b}), \mathbf{x}^*_\Omega - \mathbf{b}_\Omega > + < \xi\nabla_\Omega\varphi(\mathbf{x}^*), \mathbf{x}^*_\Omega - \mathbf{b}_\Omega >$$

$$\le \delta||\mathbf{x}^* - \mathbf{b}||_2||\mathbf{x}^*_\Omega - \mathbf{b}_\Omega||_2 + \xi||\nabla_\Omega\varphi(\mathbf{x}^*)||_2||\mathbf{x}^*_\Omega - \mathbf{b}_\Omega||_2$$

where $\nabla_\Omega\varphi(\mathbf{b}) = 0$ since $\mathbf{b}$ is minimized at Line 5 of Graph-TPP, and Property (5) is applied. Then we have $||\mathbf{x}^*_\Omega - \mathbf{b}_\Omega||_2 \le \delta||\mathbf{x}^* - \mathbf{b}||_2 + \xi||\nabla_\Omega\varphi(\mathbf{x}^*)||_2$.

As $||\mathbf{x}^* - \mathbf{b}||_2 \le ||\mathbf{x}^*_\Omega - \mathbf{b}_\Omega||_2 + ||\mathbf{x}^*_{\Omega^c} - \mathbf{b}_{\Omega^c}||_2$, compute the upper bound.

$$||\mathbf{x}^* - \mathbf{b}||_2 \le \frac{||\mathbf{x}^*_{\Omega^c} - \mathbf{b}_{\Omega^c}||_2}{1 - \delta} + \frac{\xi||\nabla_\Omega\varphi(\mathbf{x}^*)||_2}{1 - \delta} \tag{10}$$

Let $\Phi = supp(\mathbf{x}^*) \in \mathcal{M}(\mathbb{Q})$. As $\Omega = supp(\mathbf{x}^i - \eta\nabla_\Gamma\varphi(\mathbf{x}^i))$, we obtain the fact as follows. Eliminate the intersection $\Phi \cap \Omega$, we obtain

$$||(\mathbf{x}^i - \eta\nabla_\Gamma\varphi(\mathbf{x}^i))_\Phi||_2 \le ||(\mathbf{x}^i - \eta\nabla_\Gamma\varphi(\mathbf{x}^i))_\Omega||_2$$

$$||(\mathbf{x}^i - \eta\nabla_\Gamma\varphi(\mathbf{x}^i))_{\Phi\setminus\Omega}||_2 \le ||(\mathbf{x}^i - \eta\nabla_\Gamma\varphi(\mathbf{x}^i))_{\Omega\setminus\Phi}||_2$$

For the right-hand formula, we have

$$\begin{aligned} &||(\mathbf{x}^i - \eta\nabla_\Gamma\varphi(\mathbf{x}^i))_{\Omega\setminus\Phi}||_2 \le \\ &||(\mathbf{x}^i - \mathbf{x}^* - \eta\nabla_\Gamma\varphi(\mathbf{x}^i) + \eta\nabla_\Gamma\varphi(\mathbf{x}^*))_{\Omega\setminus\Phi}||_2 + \eta||\nabla_{\Gamma\cup\Omega}\varphi(\mathbf{x}^*)||_2 \end{aligned}$$

where $||\mathbf{x}^*_{\Omega\setminus\Phi}||_2 = 0$. For the left-hand formula, we have

$$\begin{aligned} &||(\mathbf{x}^i - \eta\nabla_\Gamma\varphi(\mathbf{x}^i))_{\Phi\setminus\Omega}||_2 \ge -\eta||\nabla_{\Gamma\cup\Omega}\varphi(\mathbf{x}^*)||_2 + \\ &||(\mathbf{x}^i - \mathbf{x}^* - \eta\nabla_\Gamma\varphi(\mathbf{x}^i) + \eta\nabla_\Gamma\varphi(\mathbf{x}^*))_{\Phi\setminus\Omega} + (\mathbf{x}^* - \mathbf{b})_{\Omega^c}||_2 \end{aligned}$$

where $\mathbf{x}^*_{\Phi\setminus\Omega} = \mathbf{x}^*_{\Omega^c}$ and $\mathbf{b}_{\Omega^c} = \mathbf{0}$. Let $\Pi = \Phi \cup \Omega - \Phi \cap \Omega$ be the symmetric difference of the sets Φ and Ω. We have that

$$\begin{aligned} &||(\mathbf{x}^* - \mathbf{b})_{\Omega^c}||_2 \\ &\le \sqrt{2}||(\mathbf{x}^i - \mathbf{x}^* - \eta\nabla_\Gamma\varphi(\mathbf{x}^i) + \eta\nabla_\Gamma\varphi(\mathbf{x}^*))_\Pi||_2 + 2\eta||\nabla_I\varphi(\mathbf{x}^*)||_2 \\ &\le \sqrt{2}||(\mathbf{x}^i - \mathbf{x}^* - \xi\nabla_\Gamma\varphi(\mathbf{x}^i) + \xi\nabla_\Gamma\varphi(\mathbf{x}^*))_\Pi||_2 + \\ &\quad \sqrt{2}(\xi - \eta)||(\nabla_\Gamma\varphi(\mathbf{x}^i) - \nabla_\Gamma\varphi(\mathbf{x}^*))_\Pi||_2 + 2\eta||\nabla_I\varphi(\mathbf{x}^*)||_2 \\ &\le \sqrt{2}||r^i_{\Gamma^c}||_2 + \sqrt{2}||(r^i_\Gamma - \xi\nabla_\Gamma\varphi(\mathbf{x}^i) + \xi\nabla_\Gamma\varphi(\mathbf{x}^*))_\Pi||_2 + \\ &\quad \sqrt{2}\big((\xi - \eta)(1 + \delta)/\xi\big)||r^i||_2 + (2\sqrt{2}(\xi - \eta) + 2\eta)||\nabla_I\varphi(\mathbf{x}^*)||_2 \\ &\le \sqrt{2}||r^i_{\Gamma^c}||_2 + \sqrt{2}\big((2 - \eta/\xi)\delta + 1 - \eta/\xi\big)||r^i||_2 + \\ &\quad 2\big(\sqrt{2}\xi + (1 - \sqrt{2}\eta)\big)||\nabla_I\varphi(\mathbf{x}^*)||_2 \end{aligned}$$

where the first inequality follows from the fact that, for $a \ge 0$ and $b \ge 0$,

$$(\sqrt{a} + \sqrt{b})^2 \le a + b + 2\sqrt{ab} \le 2(a + b)$$

Next analyze the upper bound of $||r^i_{\Gamma^c}||_2$.

$$\begin{aligned} ||\nabla_\Gamma\varphi(\mathbf{x}^i)||_2 &\ge ||\nabla_\Gamma\varphi(\mathbf{x}^i) - \nabla_\Gamma\varphi(\mathbf{x}^*)||_2 - ||\nabla_\Gamma\varphi(\mathbf{x}^*)||_2 \\ &\ge (1 - \delta)||r^i||_2/\xi - ||\nabla_I\varphi(\mathbf{x}^*)||_2 \qquad \triangleright \text{ condition-WRSC} \end{aligned}$$

Let $\Psi = supp(r^i)$. Inequalities on the other side,

$$\begin{aligned} &||\nabla_\Gamma\varphi(\mathbf{x}^i)||_2 \le (1/\xi)||\xi\nabla_\Gamma\varphi(\mathbf{x}^i) - \xi\nabla_\Gamma\varphi(\mathbf{x}^*)||_2 + ||\nabla_\Gamma\varphi(\mathbf{x}^*)||_2 \\ &= (1/\xi)\cdot||\xi\nabla_\Gamma\varphi(\mathbf{x}^i) - \xi\nabla_\Gamma\varphi(\mathbf{x}^*) - r^i_\Gamma + r^i_\Gamma||_2 + ||\nabla_\Gamma\varphi(\mathbf{x}^*)||_2 \\ &\le (1/\xi)\cdot||\xi\nabla_{\Gamma\cup\Psi}\varphi(\mathbf{x}^i) - \xi\nabla_{\Gamma\cup\Psi}\varphi(\mathbf{x}^*) - r^i_{\Gamma\cup\Psi}||_2 + \\ &\quad (1/\xi)\cdot||r^i_\Gamma||_2 + ||\nabla_\Gamma\varphi(\mathbf{x}^*)||_2 \\ &\le (\delta/\xi)\cdot||r^i||_2 + (1/\xi)\cdot||r^i_\Gamma||_2 + ||\nabla_I\varphi(\mathbf{x}^*)||_2 \end{aligned}$$

where the third inequality satisfies the condition-WRSC and $r^i_{\Gamma\cup\Psi} = r^i$. With the two bounds, we obtain the inequality.

$$||r^i_\Gamma||_2 \ge (1 - 2\delta)||r^i||_2 - 2\xi||\nabla_I\varphi(\mathbf{x}^*)||_2$$

By applying the similar algebraic techniques in [Hegde *et al.*, 2014] Page 11, we obtain

$$||r^i_{\Gamma^c}||_2 \le 2\sqrt{\delta - \delta^2}||r^i||_2 + \left(\frac{2\xi}{1 - 2\delta} + \frac{(1 - 2\delta)\xi}{\sqrt{\delta - \delta^2}}\right)||\nabla_I\varphi(\mathbf{x}^*)||_2$$

Combining above inequalities, we prove this theorem. □

B Proof of Theorem 2

Proof. The i-th solution $\mathbf{x}^i$ in Algorithm 1 satisfies

$$\begin{aligned} ||\mathbf{x}^* - \mathbf{x}^i||_2 &\le \alpha^i||\mathbf{x}^*||_2 + \frac{\beta}{1 - \alpha}||\nabla_I\varphi(\mathbf{x}^*)||_2 \\ &\le (1 + \frac{\beta}{1 - \alpha})||\nabla_I\varphi(\mathbf{x}^*)||_2 \end{aligned} \tag{11}$$

After the $t = \lceil \log\big(||\mathbf{x}^*||_2/||\nabla_I\varphi(\mathbf{x}^*)||_2\big)/\log(1/\alpha)\rceil$ iterations, Graph-TPP returns a desired estimate $\hat{\mathbf{x}}$. As T is the time cost of execution for one iterate in Algorithm 1, and the total iterations are $\lceil \log\big(||\mathbf{x}^*||_2/||\nabla_I\varphi(\mathbf{x}^*)||_2\big)/\log(1/\alpha)\rceil$, we prove this theorem over. □

[2] Generate *sql injection attacks* by Common Vulnerability.

[3] Generate *FckEditor* attacks by a vulnerable FckEditor version.

References

[Bach *et al.*, 2011] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2011.

[Bahmani *et al.*, 2013] Sohail Bahmani, Bhiksha Raj, and Petros T. Boufounos. Greedy sparsity-constrained optimization. *JMLR*, 14(1):807–841, 2013.

[Bahmani *et al.*, 2016] Sohail Bahmani, Petros T. Boufounos, and Bhiksha Raj. Learning model-based sparsity via projected gradient descent. *IEEE Trans. Inf. Theory*, 62(4):2092–2099, 2016.

[Chen and Neill, 2014] Feng Chen and Daniel B. Neill. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 1166–1175, 2014.

[Chen and Zhou, 2016] Feng Chen and Baojian Zhou. A generalized matching pursuit approach for graph-structured sparsity. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, 9-15 July 2016*, pages 1389–1395, 2016.

[Choi *et al.*, 2009] Hyunsang Choi, Heejo Lee, and Hyogon Kim. Botgad: detecting botnets by capturing group activities in network traffic. In *Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE*, page 2. ACM, 2009.

[Du and Yang, 2011] Haitao Du and Shanchieh Jay Yang. Discovering collaborative cyber attack patterns using social network analysis. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 129–136. Springer, 2011.

[Gionis *et al.*, 2015] Aristides Gionis, Michael Mathioudakis, and Antti Ukkonen. Bump hunting in the dark: Local discrepancy maximization on graphs. In *31st IEEE International Conference on Data Engineering, ICDE 2015, April 13-17, 2015*, pages 1155–1166, 2015.

[Gupta *et al.*, 2014] Manish Gupta, Jing Gao, Xifeng Yan, Hasan Cam, and Jiawei Han. Top-k interesting subgraph discovery in information networks. In *2014 IEEE 30th International Conference on Data Engineering*, pages 820–831. IEEE, 2014.

[Hegde *et al.*, 2014] Chinmay Hegde, Piotr Indyk, and Ludwig Schmidt. Approximation-tolerant model-based compressive sensing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, January 5-7, 2014*, pages 1544–1561, 2014.

[Hegde *et al.*, 2015] Chinmay Hegde, Piotr Indyk, and Ludwig Schmidt. A nearly-linear time framework for graph-structured sparsity. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 928–937, 2015.

[Jain *et al.*, 2014] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Dec. 8-13 2014, Montreal, Quebec, Canada*, pages 685–693, 2014.

[Kim *et al.*, 2015] Jinha Kim, Hyungyu Shin, Wook-Shin Han, Sungpack Hong, and Hassan Chafi. Taming subgraph isomorphism for rdf query processing. *Proceedings of the VLDB Endowment*, 8(11):1238–1249, 2015.

[McFowland *et al.*, 2013] Edward McFowland, Skyler Speakman, and Daniel B Neill. Fast generalized subset scan for anomalous pattern detection. *Journal of Machine Learning Research*, 14(1):1533–1561, 2013.

[Neill, 2009] Daniel B Neill. An empirical comparison of spatial scan statistics for outbreak detection. *International journal of health geographics*, 8(1):1, 2009.

[Pan *et al.*, 2013] Bei Pan, Yu Zheng, David Wilkie, and Cyrus Shahabi. Crowd sensing of traffic anomalies based on human mobility and social media. 2013.

[Patil *et al.*, 2003] GP Patil, C Taillie, et al. Geographic and network surveillance via scan statistics for critical area detection. *Statistical Science*, 18(4):457–465, 2003.

[Qian *et al.*, 2014] Jing Qian, Venkatesh Saligrama, and Yuting Chen. Connected sub-graph detection. In *AISTATS '14, Iceland, April 22-25, 2014*, pages 796–804, 2014.

[Tong *et al.*, 2007] Hanghang Tong, Christos Faloutsos, Brian Gallagher, and Tina Eliassi-Rad. Fast best-effort pattern matching in large attributed graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, August 12-15, 2007*, pages 737–746, 2007.

[Wu *et al.*, 2016] Nannan Wu, Feng Chen, Jianxin Li, Baojian Zhou, and Naren Ramakrishnan. Efficient nonparametric subgraph detection using tree shaped priors. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1352–1358, 2016.

[Yang *et al.*, 2014] Shengqi Yang, Yinghui Wu, Huan Sun, and Xifeng Yan. Schemaless and structureless graph querying. *Proceedings of the VLDB Endowment*, 7(7):565–576, 2014.

[Yang *et al.*, 2016] Shengqi Yang, Fangqiu Han, Yinghui Wu, and Xifeng Yan. Fast top-k search in knowledge graphs. In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pages 990–1001, 2016.

[Yuan and Liu, 2014] Xiao-Tong Yuan and Qingshan Liu. Newton greedy pursuit: A quadratic approximation method for sparsity-constrained optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4122–4129, 2014.

[Yuan *et al.*, 2014] Xiaotong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 127–135, 2014.