Near-Optimal and Practical Algorithms for Graph Scan Statistics with Connectivity Constraints

JOSE CADENA, Dept. of Computer Science and Biocomplexity Institute, Virginia Tech
FENG CHEN, Dept. of Computer Science, University of Albany - SUNY
ANIL VULLIKANTI, Dept. of Computer Science and Biocomplexity Institute, Virginia Tech

One fundamental task in network analysis is detecting "hotspots" or "anomalies" in the network; that is, detecting subgraphs where there is significantly more activity than one would expect given historical data or some baseline process. Scan statistics is one popular approach used for anomalous subgraph detection. This methodology involves maximizing a score function over all connected subgraphs, which is a challenging computational problem. A number of heuristics have been proposed for these problems, but they do not provide any quality guarantees. Here, we propose a framework for designing algorithms for optimizing a large class of scan statistics for networks, subject to connectivity constraints. Our algorithms run in time that scales linearly on the size of the graph and depends on a parameter we call the "effective solution size," while providing rigorous approximation guarantees. In contrast, most prior methods have super-linear running times in terms of graph size. Extensive empirical evidence demonstrates the effectiveness and efficiency of our proposed algorithms in comparison with state-of-the-art methods. Our approach improves on the performance relative to all prior methods, giving up to over 25% increase in the score. Further, our algorithms scale to networks with up to a million nodes, which is 1–2 orders of magnitude larger than all prior applications.

CCS Concepts: • **Computing methodologies** → **Anomaly detection**; • **Networks** → *Network algorithms*;

Additional Key Words and Phrases: Scan statistics, anomalous subgraph detection, graph anomaly detection, parameterized complexity

ACM Reference format:
Jose Cadena, Feng Chen, and Anil Vullikanti. 2019. Near-Optimal and Practical Algorithms for Graph Scan Statistics with Connectivity Constraints. *ACM Trans. Knowl. Discov. Data* 13, 2, Article 20 (April 2019), 33 pages.
https://doi.org/10.1145/3309712

Jose Cadena and Anil Vullikanti were part of Virginia Tech when this work was completed.
The work of Jose Cadena and Anil Vullikanti has been partially supported by the following grants: DTRA CNIMS Contract HDTRA1-11-D-0016-0010, NSF BIG DATA Grant IIS-1633028 and NSF DIBBS Grant ACI-1443054. The work of Feng Chen was supported in part by NSF under Grants IIS-1750911 and IIS-1815696. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. LLNL-JRNL-766366.
Authors' addresses: J. Cadena, 7000 East Ave., Livermore, CA 94550; email: cadenapico1@llnl.gov; F. Chen, UAB 426, 1215 Western Ave, Albany, NY 12222; email: fchen5@albany.edu; A. Vullikanti, 85 Engineer's Way Charlottesville, VA 22904; email: asv9v@virginia.edu.

1556-4681/2019/04-ART20 $15.00
https://doi.org/10.1145/3309712

1 INTRODUCTION

Detecting "hotspots" and "anomalies" is a recurring problem in a wide range of applications, such as social network analysis, epidemiology, finance, and bio-surveillance (Ding et al. 2012; Eberle and Holder 2009). Furthermore, the applications mentioned above involve network abstractions, so anomaly detection in network data has become a very active area of research in recent years—see Akoglu et al. (2015) and Cadena et al. (2018) for surveys on network anomaly detection. A number of methods have been proposed for anomaly detection in networks: these approaches typically formalize anomalies as subgraphs whose properties deviate from some kind of baseline. For instance, Akoglu et al. (2010) use different kinds of metrics within the egonets of nodes, such as the total weight, number of triangles, and principal eigenvalues, to identify anomalous nodes. Hooi et al. (2016) use the density of a subgraph—i.e., the ratio of the number of edges within the subgraph to the subgraph size—to identify anomalous subgraphs. However, the definition of anomalies in graphs and algorithms to find these anomalies tend to be application specific and rely on feature engineering.

One of the few approaches that has a sound and general statistical basis is the paradigm of *scan statistics* (Kulldorff 1997). This is among the most powerful and widely used methods for anomaly detection, not only for graphs, but also for spatio-temporal datasets. Scan statistics is based on hypothesis testing. In this setting, we have a graph $G = (V, E)$, with two kinds of counts—baseline and event counts—associated with each node. The null hypothesis H_0 in the simplest model is that event counts for all nodes are generated proportional to their baseline counts. The alternative hypothesis $H_1(S)$ is that within a subset $S \subseteq V$ of vertices, the counts are generated from a different process, typically at a higher rate than what the null hypothesis assumes. Then, the goal is to find a subgraph that maximizes a "score" function that quantifies the likelihood of $H_1(S)$ compared to H_0—for instance, a likelihood ratio.

Scan statistics were originally developed for spatial data (Kulldorff 1997; McFowland et al. 2013; Neill 2012) with the goal of finding spatial clusters where the rate of incidence of a disease is higher than the average rate in the entire data. Recently, these methods have been extended to network data by considering *connected subgraphs* instead of spatial clusters. The connectivity constraint is important because it ensures that subgraphs reflect changes due to localized in-network processes. More generally, scan statistics require some kind of constraint on the anomalous regions for the formulation to be of practical use. In the graph setting, for instance, without connectivity constraints, the most "anomalous" subset would trivially consist of the nodes with the highest event counts, potentially spread out all over the graph. From a practical point of view, such subgraph is not interpretable or actionable. Second, by mere randomness, it is expected to have several nodes in the graph with much higher event counts than what the null hypothesis assumes. What is interesting is when these high-count nodes are localized—either spatially or by being connected—since this gives us evidence of some underlying anomalous process occurring in that part of the graph.

As an illustration of these concepts and the challenges, consider the snapshots of a toy sensor network in Figure 1. This type of network has been used to detect pollution on a water distribution system (Ostfeld et al. 2008). Each node is a sensor, and an edge represents a water pipe between two sensors. We would like to detect pollution in the network, so that remedial action is taken. However, a sensor could become active due to noisy observations; similarly, a sensor that should be active may fail to detect pollution. For example, at times 2 and 3 in the figure, we observe some active sensors (colored blue), but these are not indicative of pollution. At time 4, on the other hand, we find a large subgraph of adjacent active sensors, which has a low likelihood of being just noise. Note, however, that we may have to include some inactive nodes if this allows us to discover a larger anomalous cluster. This event detection problem can be formalized as follows: given a

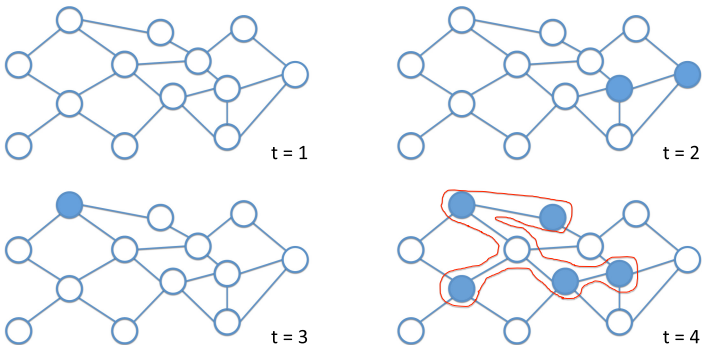

Fig. 1. Anomalous subgraph detection. Four snapshots of a network of sensors. A blue node (sensor) indicates pollution at that part of the network. However, individual sensors may become active due to noise. This is the case at times 2 and 3. However, time 4 shows a large subgraph of active sensors. This event detection problem can be cast as finding a connected subgraph with a high proportion of blue nodes—possibly connected by some white nodes. In this case, we would like to detect the subgraph circled in red at time 4.

sensor graph $G(V, E)$ and a score function $F : 2^V \rightarrow \mathbb{R}$, we want to find a subset of connected nodes $S \subseteq V$, such that $F(S)$ is maximized.

A large number of scan statistics have been developed as a result of the diversity of applications where they have been applied; we briefly describe some of them as follows:

— *Social Science:* Detection of human rights events (Chen and Neill 2015) and civil unrest (Chen and Neill 2014a).
— *Disease surveillance:* Early detection of respiratory disease outbreaks (Neill 2012) and clusters with high incidence of breast cancer (Boscoe et al. 2016).
— *Security:* Network intrusion detection and illicit activities in shipment data (McFowland et al. 2013).

Depending on whether the notion of anomalousness is with respect to an underlying model for the data or historical values, scan statistics can be *parametric* or *non-parametric* (Section 3). Parametric scan statistics assume that counts associated with each node are generated from a parameterized distribution, e.g., Poisson or Normal. One of the most widely used parametric scan statistic is the Kulldorff statistic, which is commonly used in disease surveillance (Kulldorff 1997). Non-parametric scan statistics are defined without assuming an underlying distribution or process on the graph. Instead, they estimate a p-value for each node by comparing its event count with historical data. An example of this type is the Berk–Jones scan statistic (BJ) (Berk and Cohen 1979)—this has been used for civil unrest events and network intrusion detection (Chen and Neill 2014a).

Finding a connected subgraph S which maximizes a function $F(S)$ of this type generalizes *Network Design* problems—this includes well-known graph optimization problems, such as the Steiner Tree problem and its variants, Prize-Collecting Steiner Tree (PCST) and NetWorth, all NP-hard. However, no formal proof of hardness of the common scan statistics (e.g., the BJ-statistic) on networks is known, to the best of our knowledge. Heuristics for these problems have been used for network scan statistics (Bogdanov et al. 2011; Rozenshtein et al. 2014; Speakman et al. 2015, 2013), but they do not give any rigorous guarantees on the solution quality.

1.1 Contributions

Here, we present a *unified algorithmic framework for graph scan statistics with connectivity constraints*. Our contributions are as follows:

(1) We show formally that maximizing the BJ statistic on a network with connectivity constraint is NP-hard. While the problem has some similarity to steiner connectivity, formally proving the hardness of the BJ-statistic is quite challenging because of the non-linear objective function.
(2) A *unified framework for optimizing a large class of parametric and non-parametric scan statistics for networks with connectivity constraints*, which scales linearly in the network size and is a function of a parameter defined as the "effective solution size." We also give rigorous bounds on the solution quality (summarized in Theorem 4.3). In other words, our framework encompasses many different network scan statistics—this contrasts with all prior methods, which are developed for specific statistics; further, our approach also holds for the extensions of these functions with both node and edge weights, which generalize Steiner connectivity problems. *In practice, the effective solution size parameter is very small* (see Section 6.6), making the time complexity of our algorithms better than prior methods, which are super-linear in the network size.
(3) Preprocessing and refinement techniques that reduce the solution size without degrading the quality score beyond a provable constant factor (Section 4.4). The resulting algorithms are able to scale to graphs with over a million nodes in minutes and are significantly faster than state-of-the-art methods, which have only been run on graphs of up to 10^4 nodes.
(4) Significant improvement over the objective scores computed by different baselines, with over 25% improvement in some instances, compared to the best baseline method (Section 6.3). Better objective scores also translate to higher anomaly detection power with 3% improvement on accuracy and F1 score over state-of-the-art methods. Our algorithmic framework has the added advantage that different score functions can be optimized by just modifying the specific objective function *within the same implementation.*

2 RELATED WORK

There is a very large body of work related to our article because of the wide range of applications. Below, we discuss the specific work related to scan statistics; we refer to the comprehensive survey by Akoglu et al. (2015) for a general discussion on other methods for graph anomaly detection.

Although a large number of detection algorithms have been proposed for different kinds of scan statistics in networks, *no computationally tractable algorithms with rigorous guarantees are known for any of the objectives discussed below*, other than the PCST objective. Table 1 compares our method with several state-of-the-art algorithms on supported scoring functions, time complexity, and performance bound. Many of these heuristics have reasonable performance on some datasets, but are not consistent, as we find in our experiments. Since better approximation bounds often imply better detection power, this can be a problem in practice.

2.1 Algorithms for Optimization of Parametric Scan Statistics

These fall into three categories as follows:

(1) *Exact algorithms*, such as exhaustive search over all connected subgraphs (Takahashi et al. 2008), a branch-and-bound method for Kulldorff's spatial scan statistic, and upper level set scan statistic (Speakman et al. 2015). However, *these methods do not scale to graphs with more than 1,000 nodes.*
(2) *Heuristic algorithms*, which include (a) a simulated annealing approach that is based on a concept of "non-compactness" for penalizing clusters (Duczmal et al. 2006), (b) the Additive GraphScan algorithm, which connects clusters based on shortest path distances (Speakman et al. 2013), (c) sparse learning method based on edge-lasso regularization

Table 1. Comparison between Subgraph Detection Methods

Method	Score function	Time Complexity	Performance Bound
MEDEN (Bogdanov et al. 2011)	Linear	$O(nt \log^2 t)$	No
EventTree+ (Rozenshtein et al. 2014)	PCST objective (Johnson et al. 2000)	$O(n^2 \log n)$	2-approximation
AdditiveGraphScan (GS) (Speakman et al. 2013)	Nonlinear	$O(mn + n^2 \log n)$	No
DepthFirstScan (DFS) (Speakman et al. 2015)	Nonlinear	$O(n \cdot 2^d)$	No
EdgeLasso (EL) (Sharpnack et al. 2012)	Quadratic	$O(l \cdot n^3)$	No
GraphLaplacian (GL) (Sharpnack et al. 2013b)	Quadratic	$O(l \cdot n^3)$	No
NPHGS (Chen and Neill 2014a)	Nonlinear	$O(n \log n)$	No
Our algorithms	Linear, Nonlinear	$O(2^k \cdot e^k m \log \frac{n}{\epsilon})$	$(1-\epsilon)$-approximation

n and m are the total numbers of nodes and edges in the input graph, respectively; d is a maximum depth parameter; l is the number of iterations; t is the number of snapshots; k is the solution size of our algorithm and is $\leq$10 in most cases.

(Sharpnack et al. 2012), (d) spectral scan method based on graph Laplacian (GL) regularization (Sharpnack et al. 2013b), and (e) submodular optimization algorithm based on Lovasz extensions (Sharpnack et al. 2013a). *No quality guarantees are known for these methods*, when used for optimizing parametric scan statistics, in general.

(3) *Algorithms based on density or Steiner connectivity*, a semi-definite-programming-based method (Qian et al. 2014), and the use of standard solutions of MAXCUT and PCST (Rozenshtein et al. 2014). These methods work well in practice and give guarantees for the PCST objective (based on (Goemans and Williamson 1997)), but they do not directly optimize a specific scan statistic function.

A number of other methods in this category consider relatively simple graphs, such as lines, lattices or trees, and planar graphs, and optimize over subgraphs of special forms, such as rectangles, balls, or some other low-dimensional parametric shapes (Agarwal et al. 2006; Dai et al. 2010; Kulldorff 1997; Neill 2009). For example, Khezerlou et al. (2017) consider scan statistic optimization over paths for early detection of "gathering events"—where many moving objects move to the same location from different paths. These methods are inapplicable to general graphs and are not reviewed here.

We also note that there has been a lot of work on parametric scan statistic optimization in non-network data (Kulldorff 1997; Neill 2012). An important result due to Neill (2012) is that unconstrained maximization of scan statistics can be performed efficiently.

2.2 Algorithms for Optimization of Non-Parametric Scan Statistics

Although non-parametric scan statistics have been widely used in a variety of pattern detection applications (Donoho and Jin 2015; Jin and Ke 2014), their applications to anomalous cluster detection have only been explored recently. Several papers apply non-parametric scan statistics to detect anomalous clusters in non-graph data (McFowland et al. 2013; Neill 2008; Neill and Lingwall 2007). Chen and Neill presented a fast heuristic algorithm to optimize non-parametric scan statistics on general graphs (Chen and Neill 2014a), with applications to detection of civil unrest, disease outbreaks (Chen and Neill 2014b), and human rights events (Chen and Neill 2015), but this algorithm does not provide worst-case theoretical guarantees.

2.3 Reduction to Variants of Network Design

A common approach for dealing with connectivity requirements is to use algorithms for PCST (Johnson et al. 2000) and other kinds of network design problems. For instance, the EventTree and EventTree+ problems in Rozenshtein et al. (2014) are exactly the PCST problem. The authors propose a simple greedy heuristic. In Bogdanov et al. (2011) and Mongiovì et al. (2013), the authors propose the heaviest dynamic subgraph (HDS) problem and a more general version called significant anomalous regions. Both formulations reduce to the NetWorth objective, which is a complement of PCST. In Chen and Neill (2014a) and Qian et al. (2014), two different scan statistic methods are proposed. These formulations have connections to the Quota Steiner tree problem and Budgeted Steiner tree, respectively. Rigorous guarantees are known for some of these objectives, such as for PCST and the budgeted Steiner tree objective. However, no guarantees are known for the NetWorth objective.

3 PRELIMINARIES

We are given a graph $G = (V, E)$, where V is a set of n vertices or nodes, and $E \subseteq V \times V$ is a set of m edges. Each vertex $v \in V$ has two values associated with it as follows: (1) a *population* count, $b(v)$, which indicates the count that we expect to see at the node v—for instance, the number of people in a county, corresponding to node v and (2) an *event count*, $c(v)$, which indicates how many occurrences of an event of interest are seen at the node—for instance, the number of cases of a disease in a county. These values vary over time, but we will not indicate the time in the notation in order to keep it simple. Our notation is summarized in Table 3.

3.1 Non-Parametric Scan Statistics

Non-parametric scan statistics do not assume an underlying distribution or process on the graph. Instead, they first estimate a p-value for each vertex based on empirical calibration by comparing the current feature ($c(v)$ and $b(v)$) of this vertex with its features in the historical data. The problem of anomaly detection has been formalized as a hypothesis testing problem for testing whether the empirical p-values are uniformly distributed on $[0, 1]$ (Awini et al. 2010; Margai and Henry 2003; Neill 2012; Qian et al. 2014; Sharpnack et al. 2013a; Vaneckova et al. 2010; Zeoli et al. 2014). Let $\alpha \in [0, 1]$ be *significance level* and let $w(v, \alpha)$ denote the *weight* of a node v as a function of α. For a set of nodes S, let $W(S, \alpha) = \sum_{v \in S} w(v, \alpha)$ and $N(S)$ be a function of the cardinality of the set. Then, the score functions can be expressed in the following general form:

$$F(S) = \max_{\alpha \leq \alpha_{\max}} \phi(W(S, \alpha), N(S), \alpha), \tag{1}$$

where ϕ is a function of S and α that depends on the particular score function to be optimized—see Table 2 for examples of ϕ. The significance level α can be optimized between 0 and some constant $\alpha_{\max}$. We use $w(v)$ and $W(S)$ to denote $w(v, \alpha)$ and $W(S, \alpha)$, respectively, whenever α is clear from the context. For clarity, from now on, we consider the specific case when $N(S) = |S|$ is the cardinality of S and $w(v, \alpha)$ is 1 if the p-value of v is less than α, 0 otherwise—we say these nodes are *significant at level* α. Then, $W(S, \alpha)$ is the number of significant nodes in S. An example of a scan statistic with this structure is the BJ scan statistic (Berk and Jones 1979) defined as

$$\max_{\alpha \leq \alpha_{\max}} |S| \cdot KL\left(\frac{W(S, \alpha)}{|S|}, \alpha\right),$$

where $\mathrm{KL}(p, q)$ is Kullback-Leibler (KL) divergence for two Bernoulli distributions with parameters p and q:

$$KL(p, q) = p \log \frac{p}{q} + (1 - p) \log \frac{1 - p}{1 - q}.$$

Table 2. Scan Statistics Functions that can be Optimized with Our Framework

Non-parametric scan statistics (The following definitions are by default, unless otherwise indicated) $F(S) = \max_{\alpha \le \alpha_{\max}} \phi(W(S,\alpha), N(S), \alpha)$, $p(v)$ refers to the p-value of node v, $N(S) = \lvert S \rvert$, $W(S,\alpha) = \sum_{v \in S} I(p(v) \le \alpha)$, where $I(\text{True}) = 1$ and $I(\text{False}) = 1$.		
Name	Original form	General form
Berk–Jones (Berk and Jones 1979)	$F(S) = \max_{\alpha \le \alpha_{\max}} N(S) KL(\frac{W(S,\alpha)}{N(S)}, \alpha)$	$\phi(a,b,\alpha) = b \cdot KL(a/b, \alpha)$, where $KL(x,\alpha) = x \log\left(\frac{x}{\alpha}\right) + (1-x)\log\left(\frac{1-x}{1-\alpha}\right)$
Higher criticism (Donoho and Jin 2004)	$F(S) = \max_{\alpha \le \alpha_{\max}} \frac{W(S,\alpha) - N(S)\alpha}{\sqrt{N(S)\alpha(1-\alpha)}}$	$\phi(a,b,\alpha) = (a - b\cdot\alpha)/\sqrt{b\cdot\alpha(1-\alpha)}$
Kolmogorov–Smirnov (Wilcox 2005)	$F(S) = \max_{\alpha \le \alpha_{\max}} \sqrt{N(S)} \cdot \left(\frac{W(S,\alpha)}{N(S)} - \alpha\right)$	$\phi(a,b,\alpha) = \sqrt{b}\left(\frac{a}{b} - \alpha\right)$
Anderson–Darling (Eicker 1979)	$F(S) = \max_{\alpha \le \alpha_{\max}} \sqrt{N(S)} \cdot \left(\frac{W(S,\alpha)}{N(S)} - \alpha\right) / \sqrt{\frac{W(S,\alpha)}{N(S)} \cdot \left(1 - \frac{W(S,\alpha)}{N(S)}\right)}$	$\phi(a,b,\alpha) = \sqrt{b}\left(\frac{a}{b} - \alpha\right) / \sqrt{\frac{a}{b} \cdot \left(1 - \frac{a}{b}\right)}$
Jager–Wellner (Jager and Wellner 2007)	$F(S) = \max_{\alpha \le \alpha_{\max}} \sqrt{N(S)} \cdot \left(1 - \sqrt{\frac{N_\alpha^-(S)}{N(S)} \cdot \alpha} - \sqrt{(1 - \frac{N_\alpha^-(S)}{N(S)})(1-\alpha)}\right)$	$\phi(a,b,\alpha) = \sqrt{b}\left(1 - \sqrt{\frac{a}{b}\cdot\alpha} - \sqrt{(1-\frac{a}{b})(1-\alpha)}\right)$
Stochastic ordering of p-values (Alves and Yu 2014)	$F(S) = N(S) \int_0^{\alpha_{\max}} \frac{(W(S,\alpha)/N(S) - \alpha)^2}{\alpha(1-\alpha)} \mathrm{d}\alpha$	$\phi(a,b,\alpha) = b \int_0^{\alpha_{\max}} \frac{(a/b - \alpha)^2}{\alpha(1-\alpha)} \mathrm{d}\alpha$
Fisher's test (Fisher 1925)	$F(S) = -\sum_{v \in S} \log p(v) / N(S)$	$W(S,\alpha) = \sum_{v \in S} \log p(v)$, $\phi(a,b,\alpha) = -a/b$
Truncated Fisher's test	$F(S) = \max_{\alpha \le \alpha_{\max}} -\frac{\sum_{v \in S} I(p(v) \le \alpha) \log p(v)}{N(S)}$	$W(S,\alpha) = \sum_{v \in S} I(p(v) \le \alpha) \log p(v)$, $\phi(a,b,\alpha) = -a/b$
Weighted Fisher's test	$F(S) = -\sum_{v \in S} \log(w(v)p(v)) / \sum_{v \in S} w(v)$, where $w(v)$ is the predefined weight of vertex v.	$W(S,\alpha) = \sum_{v \in S} \log(w(v)p(v))$, $N(S) = \sum_{v \in S} w(v)$, $\phi(a,b,\alpha) = -a/b$
Stouffer's test (Stouffer et al. 1949)	$F(S) = -\frac{\sum_{v \in S} \Phi^{-1}(1-p(v))}{\sqrt{N(S)}}$	$W(S,\alpha) = \sum_{v \in S} \Phi^{-1}(1-p(v))$, $\phi(a,b,\alpha) = -a/b$, where $\Phi^{-1}(\cdot)$ refers to the inverse cumulative density function of standard Gaussian distribution
Edgington's test (Edgington 1972)	$F(S) = -\sum_{v \in S} \log p(v) / N(S)$	$W(S,\alpha) = \sum_{v \in S} \log p(v)$, $\phi(a,b,\alpha) = -a/b$
Parametric scan statistics (The following defintions are by default, unless otherwise indicated) $F(S) = g(C(S), B(S))$, $C(S) = \sum_{v \in S} c(v)$, $B(S) = \sum_{v \in S} b(v)$		
Positive elevated mean scan statistic (Qian et al. 2014)	$F(S) = \sum_{i \in S} x_i / \sqrt{N(S)}$	$g(a,b) = a/\sqrt{b}$
Elevated mean scan statistic (Qian et al. 2014)	$F(S) = (\sum_{i \in S} x_i)^2 / N(S)$	$g(a,b) = a^2/b$
Expectation-based Poisson scan statistic (Neill 2012)	$F(S) = C(S)\log(C(S)/B(S)) + B(S) - C(S)$	$g(a,b) = a\log(a/b) + b - a$
Kulldorff scan statistic (Kulldorff 1997)	$F(S) = C(S)\log\left(\frac{C(S)}{B(S)}\right) + (C - C(S))\log\left(\frac{C-C(S)}{B-B(S)}\right) - C\log\left(\frac{C}{B}\right)$, where $C = \sum_{v \in \mathbb{V}} c(v)$ and $B = \sum_{v \in \mathbb{V}} b(v)$.	$g(a,b) = a\log(\frac{a}{b}) + (C-a)\log(\frac{C-a}{B-b}) - C\log\left(\frac{C}{B}\right)$
Expectation-based Gaussian scan statistic (Neill 2012)	$F(S) = (C(S) - B(S))^2/(2B(S))$, where $\sigma(v)$ refers to the standard deviation of $c(v)$ that is calibrated based on its historical observations, $C(S) = \sum_{v \in S}(c(v)b(v))/\sigma(v)^2$, and $B(S) = \sum_{v \in S} b(v)/\sigma(v)^2$	$g(a,b) = (a-b)^2/(2b)$

(Continued)

Table 2. Continued

Expectation-based exponential scan statistic (Neill 2012)	$F(S) = B(S)\log(B(S)/C(S)) + C(S) - B(S)$, where $C(S) = \sum_{v\in S} c(v)/b(v)$, $B(S) = \lvert S\rvert$	$g(a, b) = a\log(a/c) + b - a$
Spatial scan statistic for multinomial data (Jung et al. 2010)	$F(S) = \sum_k \{C_k(S)\log(\frac{C_k(S)}{C(S)}) + (C_k - C_k(S))\log\frac{C_k - C_k(S)}{C - C(S)}\} - \sum_k C_k\log(C_k/C)$, where $C_k(S)$ refers to the count of vertices of category k, $C(S) = \lvert S\rvert$, and $C = \lvert\mathbb{V}\rvert$.	$C(S) = \sum_k \{C_k(S)\log(\frac{C_k(S)}{C(S)}) + (C_k - C_k(S))\log\frac{C_k - C_k(S)}{C - C(S)}\}$, $B(S) = \sum_k C_k\log(C_k/C)$, $g(a, b) = a - b$

Table 3. Definitions and Notation Used in the Article

Term	Description
$b(v)$, $c(v)$	Population and event counts of node v
α, $\alpha_{\max}$	Significance level, maximum significance level
Significant node (at level α)	A node with p value below α
$\mathrm{Nbr}(v)$	Set of neighbors of v
$w(v)$, $w(v, \alpha)$	Weight of node v, based on its p-value and the significance level α
$W(S)$, $W(S, \alpha)$	Denotes $\sum_{v\in S} w(v, \alpha)$
$F(S)$	Any of the functions in Table 2
K	The set $\{1, \ldots, k\}$
$\mathrm{col}(u)$	Color of node u from set K
T	Subset of K (denotes colors)
$M(v, T)$	$\max_S W(S)$, where the maximization is over connected colorful sets $S \subseteq V$, such that $v \in S$ and $\{\mathrm{col}(u) : u \in S\} = T$
ψ_i, $\psi_i(\alpha)$	$\max_{T:\lvert T\rvert=i} M(v, T)$. Maximum weight over connected colorful sets of size i
S_i^*, $S_i^*(\alpha)$	Set with weight ψ_i
$OPT(F, k)$	$\max_{S:\lvert S\rvert\le k} F(S)$, where the maximum is over connected subsets S of size $\le k$

Intuitively, the BJ statistic measures how much the fraction of significant nodes in a set S deviates from α, which is the fraction of significant nodes we expect to see if the p-values were uniformly distributed in V.

3.2 Parametric Scan Statistics

Parametric scan statistics assume that counts observed at each node are generated from some parameterized distribution and formalize anomaly detection as a hypothesis testing problem (Kulldorff 1997; Neill 2012). Common choices are distributions from the exponential family, such as Poisson or Normal. Under the alternative hypothesis $H_1(S)$, an underlying anomalous phenomenon is characterized in the following manner: features of a majority of the vertices are generated from the same background distribution, and features of a small connected subset $S \subseteq V$ of vertices are generated from a different distribution. The goal is to maximize an appropriate scan statistic function $F(S)$, typically a likelihood ratio. These score functions can be expressed as

$$F(S) = g(C(S), B(S)), \tag{2}$$

where $C(S) = \sum_{v \in S} c(v)$, $B(S) = \sum_{v \in S} b(v)$, and the function g is defined depending on the score function considered. A well-known example of this class of functions is the Kulldorff scan statistic, commonly used in disease surveillance (Duczmal et al. 2006; Kulldorff 1997; Kulldorff et al. 2003; Neill 2012) and defined as

$$C(S) \log\left(\frac{C(S)}{B(S)}\right) + (C(V) - C(S)) \log\left(\frac{C(V) - C(S)}{B(V) - B(S)}\right) - B(V) \log\left(\frac{C(V)}{B(V)}\right),$$

Table 2 shows non-parametric and parametric scan statistics that can be optimized using our proposed methods.

Limitations of scan statistics. The suitability of scan statistics depends on the application and the assumptions underlying the dataset. We refer to Margai and Henry (2003), Neill (2012), Kulldorff (1997), and Neill and Lingwall (2007) for a more detailed discussion of the advantages and limitations of these approaches.

3.3 Problem Formulation

From the discussion above, the graph anomaly detection task can be posed as the following constrained optimization problem.

Problem 1. *Given a graph $G = (V, E)$, a scan statistic $F(\cdot)$, and the associated counts for vertices—represented by vectors* **c** *and* **b***—find a connected subset $S \subseteq V$ that maximizes $F(S)$.*

3.4 Hardness

In the absence of any connectivity requirement, many of the scan statistics in Table 2 can be optimized efficiently. In particular, functions like the BJ and Kulldorff scan statistics satisfy a linear ordering property, which leads to a linear time algorithm (Neill 2012). In the presence of connectivity constraints, optimizing scan statistics on graphs becomes much harder. However, no formal proof of hardness is known for any of the common scan statistics, such as the BJ statistic. Here, we show formally that maximizing the BJ statistic with connectivity requirement is NP-hard.

Theorem 3.1. *Maximizing the BJ statistic on a network with connectivity requirements is NP-complete.*

Since the proof is quite complex, we have presented this in the Appendix. We note that the hardness is actually for a decision version of the problem, formalized as problem BJ-D in the Appendix.

3.5 Final Problem Formulation: Scan Statistics with Size Constraint

In light of the NP-completeness in Theorem 3.1, it is unlikely that we would be able to compute the optimal solutions to the score functions efficiently. One approach that has been successfully used to combat computational hardness is *fixed parameter tractable algorithms*: the idea is to find an algorithm whose running time is $O(c^k f(n, m))$, where $f(n, m)$ is a polynomial function of the number of nodes, n, and the number of edges, m, c is a constant, and k is a parameter. In other words, the algorithm is exponential in a parameter other than the input size, but polynomial in the input size. We consider the solution size as a parameter: the objective is to maximize $F(S)$, restricted to sets with $|S| \leq k$, where k is a parameter that represents the *solution size*.

Problem 2. *Given a graph $G = (V, E)$, a scan statistic $F(\cdot)$, associated counts for vertices—represented by vectors* **c** *and* **b***, and a parameter k, find a connected subset $S \subseteq V$ with $|S| \leq k$, that maximizes $F(S)$.*

In Section 4, we develop algorithms for the above problem. In Section 4.4, we show that we can compress specific subsets of nodes into "supernodes," using a process we refer to as *refinement*.

The size of a set S computed in terms of these supernodes will be referred to as the *effective solution size*, and it becomes significantly smaller than the original size of S. Our final algorithms will find solutions with effective solution size at most k.

4 ALGORITHMS FOR NON-PARAMETRIC SCAN STATISTICS

In this section, we present an algorithm for non-parametric functions that are characterized by Equation (1), and then we discuss techniques to scale it without losing the quality guarantees. Our algorithm takes as input a network $G = (V, E)$, a p-value $p(v)$ for each node in the graph, and parameters that we discuss below. The p-values are computed based on modeling assumptions about the statistical process that generates the event counts. For example, for some application, if we assume that all the event counts are generated from a Poisson distribution with parameter λ, the p-value of a node with event count $c(v)$ could be the probability of observing counts at least as extreme as $c(v)$.

Our algorithm relies on two main ideas, namely *monotonicity* and *constraining the solutions*.

4.1 First Idea: Monotonicity

A key observation is that the functions $\phi(W(S, \alpha), N(S), \alpha)$ are monotonically increasing functions of $W(S, \alpha)$ under some conditions, as described below.

Lemma 4.1. *The non-parametric scan statistics functions characterized by Equation (1) are increasing functions of $W(S, \alpha)$ if $\frac{W(S,\alpha)}{N(S)} \geq \alpha$ and $N(S)$ is constant.*

For example, in the BJ statistic from Table 2, the function increases with the number of significant nodes—nodes with p-value less than α. Further, given two node sets of the same size, the set with more significant nodes scores higher according to the BJ statistic. This provides us with a way to optimize $F(S)$ by maximizing the function $\phi(\cdot)$ for sets of fixed size. In the next subsection, we describe how exactly to optimize $F(S)$ for a fixed subgraph size efficiently by constraining the space of possible solutions.

4.2 Second Idea: Constraining the Solutions

We introduce the idea of a *coloring* of the nodes and only consider connected subgraphs in which all nodes have distinct colors. Our approach builds on the color-coding technique of Alon et al. (1995), but it involves several new techniques to scale the algorithm up to graphs with millions of nodes.

Let $K = \{1, 2, \ldots, k\}$ be a set of colors—where k is a parameter—and let $\text{col}(v) \in K$ denote the color for node v. We say that a subgraph induced by set $S \subseteq V$ is *colorful* if $\text{col}(u) \neq \text{col}(v)$, for all $u, v \in S$. For a node v and subset of colors $T \subseteq K$, we let $M(v, T) = \max_S W(S)$, where the maximization is over all connected and colorful sets $S \subseteq V$, such that $v \in S$, $|S| = |T|$, and $\{\text{col}(u) : u \in S\} = T$. In other words, we only consider a set S if each node in the set has a distinct color from T. These definitions are illustrated in Figure 2. $M(v, T)$ can be computed by a dynamic program with a recurrence given in the lemma below.

Lemma 4.2. *Let $M(v, T)$ be defined as above. For any node v and color s, $M(v, \{s\}) = w(v)$ if $col(v) = s$, else $M(v, \{s\}) = -\infty$. If $|T| \geq 2$:*

$$M(v, T) = \max_{\substack{u \in Nbr(v) \\ T_1, T_2 \subseteq T}} \{M(v, T_1) + M(u, T_2)\},$$

where the maximum is over all partitions $T_1 \cup T_2 = T$ of the set T and all neighbors u of v.

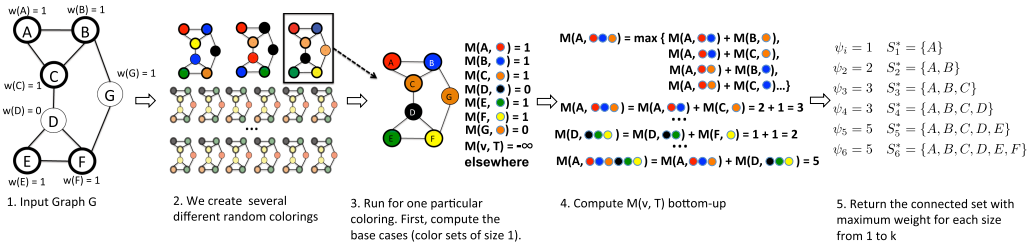

Fig. 2. Example illustrating the MaxWeight procedure for $k = 6$ colors. (1) Nodes A, B, C, E, and F in the input graph have weight 1; D and G have weight 0. (2) We generate many random colorings of the nodes, as per the error parameter ϵ'. (3) For each coloring, we solve the dynamic program given in Lemma 4.2. M(A, {red}) is 1 because node A has weight 1 and its color is red; in other words, there exists a tree that is colorful with respect to {red} and contains node A. For all other colors c, we set M(A, {c}) = $-\infty$. (4) We compute M(v,T) bottom up. M(A, {red, blue, orange}) is maximized by adding M(A, {red, blue}) and M(C, {orange}). The corresponding colorful subtrees have nodes {A, B, C}, {A, B}, and {C}, respectively. In other words, the weight of tree {A, B, C} is the sum of weights of its subtrees {A, B}, and {C}. (5) We return the maximum weight and corresponding subtree of sizes 1 to k. The optimal subtree may not be colorful in one particular coloring, but, over all the random colorings, we will find the optimal subtree for each size up to k with high probability.

Proof. Suppose $M(v,T)$ is achieved for a connected set S, such that $|S| = |T|$ and $\{\text{col}(u) : u \in S\} = T$, with $M(v,T) = \sum_{i \in S} w(i,\alpha)$. We claim that there exists $u \in \text{Nbr}(v)$, and partitions $T = T_1 \cup T_2$, and $S = S_1 \cup S_2$, such that (1) $M(v,T) = M(v,T_1) + M(u,T_2)$, (2) $\{\text{col}(i) : i \in S_1\} = T_1$ and $\{\text{col}(i) : i \in S_2\} = T_2$, and (3) the subsets of nodes S_1 and S_2 are connected. Since S is connected, there exists a tree H that spans S and contains node v. Further, there must exist a node $u \in \text{Nbr}(v)$ such that $(u,v) \in T$, since $|T| = |H| \geq 2$. Let H_1 and H_2 be the trees rooted at nodes v and u, respectively, that result when edge (u,v) is deleted in H. Let S_1 and S_2 denote the sets of nodes in H_1 and H_2, respectively. Let T_1 and T_2 be the colors used by S_1 and S_2, respectively. By construction, we have $M(v,T) = M(v,T_1) + M(u,T_2)$, so that the partitions $S = S_1 \cup S_2$ and $T = T_1 \cup T_2$ satisfy the requirements mentioned earlier. Therefore, the recurrence follows. □

4.3 ColCodeNP

In Algorithm 1, we present ColCodeNP for optimizing non-parametric scan statistics. Recall the notation in Table 3. Let $F(S)$ denote any of the non-parametric functions in Table 2, and let $OPT(F,k) = \max_{S:|S| \leq k} F(S)$, where the maximum is over all connected subsets S of size $\leq k$, for a given $\alpha_{\max}$. Algorithm ColCodeNP takes the size bound k as input, and an *error parameter* ϵ, which indicates the probability of not finding the optimum solution. We describe the main steps of ColCodeNP connecting with the two ideas from above.

- The set A in line 3 of ColCodeNP denotes the set of distinct p-values of the nodes less than $\alpha_{\max}$; it suffices to find the maximum of $\phi(W(S,\alpha), N(S), \alpha)$ for $\alpha \in A$. The **for** loop in lines 4–6 finds the best solution for each $i \in K$ and any given α (by calling MaxWeight in line 6), and the maximum is computed in line 7.
- MaxWeight finds the best solution S_i^* of size i for each $i \in K$ using the idea described in Section 4.2. We show an example in Figure 2 and the pseudocode appears in Algorithm 1. Each iteration of the outer **for** loop in lines 14–20 starts with a random coloring (line 15 in the pseudocode and Step 2 in Figure 2). The inner **for** loop in lines 16–17 computes the base case of the dynamic program from Lemma 4.2; then, we solve the program bottom-up in lines 18–19. These are Steps 3 and 4 in Figure 2.

ALGORITHM 1: ColCodeNP$((G(V,E),\alpha_{\max}),k,\epsilon)$.

1: **Input**: Instance $(G(V,E),\alpha_{max})$, parameters k, ϵ
2: **Output**: Set S^* with score $OPT(F,k)$
3: Let A be the set of p-values of nodes in V below α_{max}
4: **for** $\alpha \in A$
5: Let $\mathbf{w}$ be a weight vector with $w(v) = w(v,\alpha)$
6: $\{S_i^*(\alpha) : i \in K\} =$ MaxWeight$(G(V,E),\mathbf{w},k,\epsilon/n^2)$
7: $S^* = argmax_{i\in K,\alpha\in A} F(S_i^*(\alpha))$
8: **return** S^*
9:
10: **procedure** MaxWeight$(G(V,E),\mathbf{w},k,\epsilon')$
11: **Input**: Instance $(G(V,E),\mathbf{w})$ and parameters k,ϵ'
12: **Output**: $\{S_i^* : i \in K\}$, such that S_i^* has weight ψ_i
13: Let $\psi_i = -\infty$ for all $i \in K$
14: **for** $j = 1$ to $e^k \log(1/\epsilon')$
15: For each node v, pick random color $col(v) \in K$
16: **for** $v \in V, s \in K$
17: $M(v,\{s\}) = w(v)$ if $col(v) = s$; $-\infty$ otherwise
18: **for** $v \in V$ and $T \subseteq K$, with $|T| \geq 2$
19: Use Lemma 4.2 to compute $M(v,T)$
20: **If** $M(v,T) > \psi_{|T|}$ **update** $\psi_{|T|} = M(v,T)$
21: **return** $\{S_i^* : \sum_{v\in S_i^*} w(v) = \psi_i, \text{for } i \in K\}$

— ψ_i keeps track of the maximum weight solution restricted to size i, and it is updated if $M(v,T)$ denotes a better solution for size $|T|$.

Theorem 4.3. *For any non-parametric function $F(\cdot)$ in Table 2, algorithm ColCodeNP returns solution S^* satisfying $\Pr[F(S^*) = OPT(F,k)] \geq 1-\epsilon$, in time $O(2^k e^k |A| m \log(n/\epsilon))$, and using space $O(2^k n)$, where A is the set defined in line 3 of Algorithm 1.*

Before proving this theorem, the following lemma establishes that we can find the set with maximum weight for a particular size i by solving the recurrence from Lemma 4.2 for many different random colorings.

Lemma 4.4. *Let $\epsilon \in (0,1)$ be any constant and define. For any fixed i,α, consider ℓ random colorings of the nodes of graph G using a random color from the set $\{1,\ldots,i\}$ for each node. Let $X_j = \max_{v,T:|T|=i} M(v,T)$ for the jth coloring. Then, $\Pr[\max_j X_j = \psi_i(\alpha)] \geq 1-\epsilon$, if $\ell \geq e^i \log 1/\epsilon$.*

Proof. Let $T = \{1,2,\ldots,i\}$ be a color set and let S_i^* be the node set that achieves ψ_i. For a random coloring of G, the probability that the set S_i^* is colorful is

$$p = \frac{i!}{i^i}.$$

For ℓ random colorings of G, the probability that S_i^* is not colorful in *any* of the colorings is $(1-p)^\ell$. We want this probability to be bounded by some small constant ϵ. Then, the number of random colorings that we should explore can be estimated as follows:

$$(1-p)^\ell = \left(1-\frac{i!}{i^i}\right)^\ell < \left(1-\frac{1}{e^i}\right)^\ell.$$

If we let ℓ be at least $-e^i \log(\epsilon)$, we have

$$\left(1 - \frac{1}{e^i}\right)^{\ell} \leq \left(1 - \frac{1}{e^i}\right)^{-e^i \log(\epsilon)} \leq e^{log(\epsilon)} = \epsilon. \qquad \square$$

Now, we present the proof of Theorem 4.3.

Proof. We start with the proof of correctness of our algorithm, which involves three parts. The first observation is that within the outer for loop in the procedure MaxWeight, for each random coloring col(·), $\max_v M(v, \{1, \ldots, k\})$ is correctly computed. This follows because the algorithm is a dynamic program that computes all $M(v, T)$ for $T \subseteq K$ using the recurrence in Lemma 4.2.

Next, we observe that the algorithm correctly finds $\psi_i(\alpha)$—the maximum weight among sets of size i for a given α—for each i, α, with probability at least $1 - \epsilon/n^2$. The procedure MaxWeight is called with parameter $\epsilon' = \epsilon/n^2$ and $e^k \log(1/\epsilon')$ colorings. Let X_{ij} be the maximum weight found over subsets of size i in the jth random coloring. By Lemma 4.4, $\Pr[\max_j X_{ij} \neq \psi_i(\alpha)] \leq \epsilon' = \epsilon/n^2$. The number of possible choices for α is $|A|$, which satisfies $|A| \leq n$. Therefore, by a union bound, it follows that for all $i, \alpha \in A$, we have $\Pr[\max_j X_{ij} \neq \psi_i(\alpha)] \leq n^2\epsilon' \leq \epsilon$, and the algorithm correctly computes $\psi_i(\alpha)$ for all i, α with probability $1 - \epsilon$.

Finally, for any fixed i, α, by Lemma 4.1, $\phi(W(S), N(S), \alpha)$ is an increasing function of $W(S)$ when $N(S)$ is fixed. This implies that $\max_{S:N(S)=i} \phi(W(S), N(S), \alpha) = \psi_i(\alpha) = F(S_i^*(\alpha))$. Therefore, $\max_{i \in K, \alpha \in A} F(S_i^*(\alpha)) = OPT(F, k)$, and it follows that Algorithm ColCodeNP correctly computes $OPT(F, k)$ with probability at least $1 - \epsilon$.

Next, we consider the space and time complexity. The algorithm maintains the array M, indexed by nodes and all possible color sets, which leads to the space complexity of $O(2^k n)$, since there are at most $2^k - 1$ possible non-empty color sets. The running time is the result of solving the recurrence for each node v, and for each color set T; this requires examining each possible partition $T_1 \cup T_2 = T$, and each neighbor $u \in \text{Nbr}(v)$, which requires time $O(|\text{Nbr}(v)|2^{|T|})$. Therefore, the total running time for each coloring is $O(\sum_v \sum_{i=0}^k |\text{Nbr}(v)|2^i) = O(\sum_v |\text{Nbr}(v)|2^k) = O(2^k m)$. The algorithm considers $O(e^k \log(n^2/\epsilon)) = O(e^k \log(n/\epsilon))$ colorings, so the running time follows. $\square$

4.4 Techniques for Scaling

ColCodeNP has rigorous guarantees on the quality of the solution; however, if applied directly, it would only be feasible to discover small anomalies due to the exponential dependence in k, the solution size. We discuss two techniques to scale ColCodeNP to networks with over a million nodes without losing the approximation guarantees significantly.

4.4.1 Graph Refinement and Effective Solution Size. Graph refinement involves compressing subsets of nodes into "supernodes." The size of a set S after refinement is determined in terms of the nodes and supernodes in it. This new size is called *effective size*, and, in practice, it is significantly smaller than the original size of S.

We observe that neighboring significant nodes can be merged without loss in quality. We illustrate with an example in Figure 3(a). In the figure, orange nodes are significant and have weight 1. The key idea is that any solution containing node A should also include nodes B and C, since $\phi(W(S, \alpha), N(S), \alpha)$ is increasing in the number of significant nodes. In the figure, we replace five nodes of weight 1 for two nodes of weights 3 and 2. The effective size of the subgraph A through F is 3. In Section 6.6, we show that this refinement is very effective in real networks, and *we can usually discover large solutions by setting* $k \leq 10$.

We formalize this idea in the following lemma.

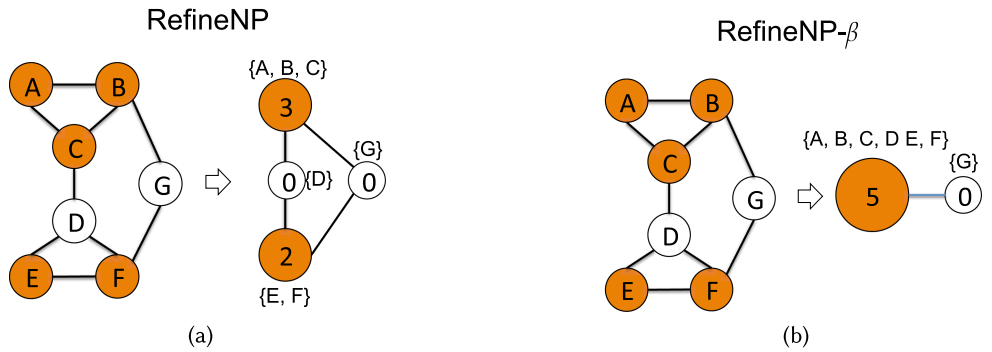

Fig. 3. (a) Graph refinement. Colored nodes are significant. Nodes A, B, and C are merged into a supernode of weight 3, and E and F form a supernode of weight 2. The effective size of the set $\{A, B, C, D, E, F\}$ is 3. (b) Graph refinement with $\beta = 5/6$. By allowing non-significant node D to be merged, the effective size of $\{A, B, C, D, E, F\}$ becomes 1 with an approximation guarantee bounded by β.

Lemma 4.5. *Let $G(V, E)$ be a network, and let $F(\cdot)$ be a non-parametric scan statistic function. Given a set $S \subseteq V$, suppose there exists a significant node $u \notin S$ and an edge $(u, v) \in E$, for some $v \in S$. Then, $F(S \cup \{u\}) \geq F(S)$.*

Proof. Non-parametric scan statistics are increasing on $\frac{W(S)}{N(S)}$. Since u is significant, we have that

$$\frac{W(S \cup \{u\})}{N(S \cup \{u\})} = \frac{W(S) + 1}{N(S) + 1} \geq \frac{W(S)}{N(S)},$$

and the proof follows. □

Exact refinement. An implication of Lemma 4.5 is that we can collapse components of significant nodes into a single node prior to running ColCodeNP. We propose a *graph refinement* to reduce the total number of significant nodes in a graph. Given a network $G(V, E)$, p-values for the nodes in V, and a significance level α, we define $V_1, \ldots, V_r$ as the connected components (or *supernodes*) of G induced by the set of significant nodes. We create a new graph $H(V', E')$, whose node set consists of $V_1, \ldots, V_r$ and the non-significant nodes in G, $V \setminus \bigcup_{i=1}^{r} V_i$. Edges between non-significant nodes are preserved in H, and we put an edge from a non-significant node u to V_i if the edge (u, v) exists, for some $v \in V_i$. Finally, we create a vector of weights, $\mathbf{w}'$. The weight of a node in H is $|V_i|$ for each supernode V_i, and 0 otherwise. This procedure is equivalent to removing all the significant nodes and replacing them with the respective supernode V_i. Figure 3(a) illustrates the procedure. After this preprocessing step, we run procedure MaxWeight for the instance $(H(V', E'), \mathbf{w}')$.

Lemma 4.6. *Let $((G(V, E), \alpha), k, \epsilon)$ be an input to ColCodeNP, and let S^*_G be the solution returned by the algorithm. Similarly, let $((H(V', E'), \alpha), k, \epsilon)$ with weights $\mathbf{w}'$ be the corresponding instance generated by graph refinement, and let S^*_H be the solution returned by ColCodeNP in this instance. Then, $F(S^*_H) = F(S^*_G)$. Furthermore, suppose $|S^*_G| = k$, then, $|S^*_H| = k'$, for some $k' \leq k$, so it is possible to execute ColCodeNP with parameter k' and still obtain $F(S^*_H) = F(S^*_G)$.*

Proof. The lemma follows from Lemma 4.5. □

Approximate refinement. It is possible to further compress the graph by including non-significant nodes into the components described above. As before, we first obtain connected components of significant nodes, $V_1, \ldots, V_r$, but now, we keep adding nodes to a component V_i as long as the number of significant nodes is at least $\beta|V_i|$, where β is a parameter between 0 and 1. We show an

example in Figure 3(b). By doing this, we are able to reduce the number of anomalous supernodes. We may not find the optimal solution now; however, we can control the error with the parameter β.

Figure 3(b) shows an example for $\beta = 5/6$. By allowing non-significant node D to be merged, we are able to combine nodes A through F in a subgraph of effective size 1 and weight 5. Note that when $\beta < 1$, the solution may not be optimal, but Lemma 4.7 describes the effect on the approximation bound.

LEMMA 4.7. *Denote the number of nodes signicant at level α in a set S as $N_\alpha(S)$. Let S^* be the set that maximizes F and let $r(S^*) = \frac{N_\alpha(S^*)}{N(S^*)}$. There is a solution on the instance H with ratio $r(S) \geq \beta r(S^*)$.*

PROOF. We split the set S^* into significant nodes, $N_\alpha(S^*)$, and non-significant nodes, $N_\alpha^-(S^*)$, such that $N(S) = N_\alpha(S^*) + N_\alpha^-(S^*)$. We now show that, in the instance H, there exists a set S with ratio

$$r(S) = \frac{N_\alpha(S)}{N(S)} \geq \frac{N_\alpha(S^*)}{N_\alpha(S^*) + N_\alpha^-(S^*) + \frac{1-\beta}{\beta} N_\alpha(S^*)}.$$

We define S' as the set formed by the supernodes V_i corresponding to the significant nodes in S^*, and we note that $N_\alpha(S') \geq N_\alpha(S^*)$; for simplicity, we assume equality. By construction, the cardinality of S' is $N(S') = N_\alpha(S') + \frac{1-\beta}{\beta} N_\alpha(S') = N_\alpha(S^*) + \frac{1-\beta}{\beta} N_\alpha(S^*)$. Note that the nodes in S' may be disconnected; however, we can connect them using a set of anomalous nodes S'' of size at most $N_\alpha^-(S^*)$. Finally, we form a set $S = S' \cup S''$ that has the desired ratio.

To conclude the proof, we compute $r(S)/r(S^*)$:

$$\frac{r(S)}{r(S^*)} = \frac{\frac{N_\alpha(S)}{N(S)}}{\frac{N_\alpha(S^*)}{N(S^*)}} \geq \frac{\frac{N_\alpha(S^*)}{N(S^*) + \frac{1-\beta}{\beta} N_\alpha(S^*)}}{\frac{N_\alpha(S^*)}{N(S^*)}} = \frac{1}{1 + \frac{1-\beta}{\beta} \times \frac{N_\alpha(S^*)}{N(S^*)}}.$$

Noticing that $\frac{N_\alpha(S^*)}{N(S^*)} \leq 1$, we obtain $r(S) \geq \beta r(S^*)$. □

4.4.2 Low Radius Subgraphs. Let $B_G(v, r)$ (referred to the ball of radius r at v) denote the set of nodes at distance at most r from v in the graph G. It suffices to run the algorithm restricted to the balls centered around significant nodes. The balls are smaller than the full graph, so they can be processed faster; furthermore, they can be processed in parallel.

We use these two techniques to reduce the size of the input graph before running MAXWEIGHT. Our algorithm, FASTCOLCODENP, with these addition is shown in Algorithm 2.

ALGORITHM 2: FASTCOLCODENP$(G(V, E), \alpha_{max}, k, \epsilon, \beta)$.

Input: Instance $(G(V, E), \alpha_{max})$, parameters k, ϵ and β
Output: Set S^* with score $OPT(F, k)$
Let A be the set of p-values of nodes in V below α_{max}
for $\alpha \in A$
  Perform approximate refinement with parameter β.
  Let $H = (V', E')$ be the refined graph with weights $\mathbf{w}'$
  $\{S_i^*(\alpha) : i \in K\} =$ MAXWEIGHT$(H(V', E'), \mathbf{w}', k, \epsilon/n^2)$
$S^* = argmax_{i \in [1,k], \alpha \in A} F(S_i^*(\alpha))$
return S^*

5 ALGORITHMS FOR PARAMETRIC SCAN STATISTICS AND EXTENSIONS

In parametric scan statistics, both $c(v)$ and $b(v)$ are used as arguments to the function, which makes the problem more challenging than for non-parametric functions. Furthermore, there exist other score functions for graph anomaly detection where both nodes and edges have weights (Bogdanov et al. 2011; Rozenshtein et al. 2014). Optimizing such functions reduces to the PCST problem (Johnson et al. 2000), which is NP-Hard. We can extend the methods described above to these settings by keeping additional information in the dynamic program. We propose algorithm FastColCodeP for parametric scan statistics and algorithm ColCodeNW for PCST and its variants.

5.1 Extensions to Parametric Scan Statistics

In Algorithm 3, we describe ColCodeP for parametric scan statistics maximization. For these functions, each node v of the input graph has two weights associated with it: $c(v)$ and $b(v)$. Therefore, we need a more general algorithm than ColCodeNP. Analogous to Lemma 4.1, we make use of the following property:

Lemma 5.1. *The parametric scan statistics functions characterized by Equation (2) are increasing functions of $C(S)$ if $C(S) > B(S)$ and $B(S)$ is constant.*

Given a graph $G(V, E)$ and vectors $\mathbf{c}$ and $\mathbf{b}$, let $M(v, T, j)$ be the maximum value $C(S)$ over all connected subsets S, such that (1) $v \in S$, (2) S is colorful with respect to T, and (3) $B(S) = j$. Here, j ranges from 1 to $B(V)$. $M(v, T, j)$ can be computed by a dynamic program with the following recurrence.

Lemma 5.2. *Let $M(v, T, j)$ be defined as above. For any node v and color s, $M(v, \{s\}, j) = c(v)$ if $col(v) = s$ and $b(v) = j$, else $M(v, \{s\}, j) = -\infty$. If $|T| \geq 2$,*

$$M(v, T, j) = \max_{\substack{u \in Nbr(v) \\ T_1, T_2 \subseteq T \\ j_1 + j_2 = j}} \{M(v, T_1, j_1) + M(u, T_2, j_2)\},$$

where the maximum is over all partitions $T_1 \cup T_2$ of the set T, all integers j_1, j_2 with $j_1 + j_2 = j$ and all neighbors u of v.

Proof. Suppose $M(v, T, j)$ is achieved for a connected set S, such that $|S| = |T|$, $\{\text{col}(u) : u \in S\} = T$, and $B(S) = j$ with $M(v, T, j) = \sum_{i \in S} c(s)$. We claim that there exists $u \in \text{Nbr}(v)$, integers j_1, j_2: $j = j_1 + j_2$, and partitions $T = T_1 \cup T_2$, and $S = S_1 \cup S_2$, such that (1) $M(v, T, B) = M(v, T_1, j_1) + M(u, T_2, j_2)$, (2) $\{\text{col}(i) : i \in S_1\} = T_1$ and $\{\text{col}(i) : i \in S_2\} = T_2$, and (3) the node sets S_1 and S_2 are connected. Since S is connected, there exists a tree H that spans S rooted at node v. Further, there must exist a node $u \in \text{Nbr}(v)$ such that $(u, v) \in T$, since $|T| = |H| \geq 2$. Let H_1 and H_2 be the trees rooted at nodes v and u, respectively, that results when edge (u, v) is deleted in H. Let S_1 and S_2 denote the sets of nodes in H_1 and H_2, respectively. Let T_1 and T_2 be the colors used by S_1 and S_2, respectively. By construction, we have $\psi(v, T, j) = \psi(v, T_1, j_1) + \psi(u, T_2, j_2)$, so that the partitions $S = S_1 \cup S_2$ and $T = T_1 \cup T_2$ satisfy the requirements mentioned earlier. Therefore, the recurrence follows. □

5.1.1 ColCodeP. Our algorithm for maximizing parametric scan statistics, ColCodeP, is presented in Algorithm 3. The procedure MaxWeightP uses Lemma 5.2 to compute $\psi_i = \max_{T:|T|=i} M(v, T, j)$ for all $i \leq k$.

Theorem 5.3. *Let $F(\cdot)$ be any of the parametric scan statistics in Table 2, and let $OPT(F, k) = \max_{S:|S| \leq k} F(S)$, where the maximum is over all connected subsets S of size $\leq k$. ColCodeP returns*

ALGORITHM 3: ColCodeP$((G(V,E),\mathbf{C},\mathbf{B}),k,\epsilon)$.

Input: Instance $(G(V,E),\mathbf{C},\mathbf{B})$, parameters k and ϵ
Output: Set S^* with score $OPT(F,k)$
$\{S_i^* : i \in K\}$ = MaxWeightP$(G(V,E),\mathbf{C},\mathbf{B},k,\epsilon/n)$
$S^* = argmax_{i\in[1,k]}F(S_i^*)$
return S^*

procedure MaxWeightP$(G(V,E),\mathbf{C},\mathbf{B},k,\epsilon')$
Input: Instance $(G(V,E),\mathbf{C},\mathbf{B})$ and parameter k
Output: Set S_i^* with maximal weight ψ_i for all $i \in [1,k]$
Let $\psi_i = -\infty$ for all $i \in [1,k]$
for $t = 1$ to $e^k \log(1/\epsilon')$
  For each node v, pick random color $col(v) \in K$
  for $v \in V, s \in K, j \le B(V)$
    $M(v,\{s\},j) = c(v)$ if $col(v) = s$ and $j = b(v)$; $-\infty$ otherwise
  for $v \in V, T \subseteq K$, with $|T| \ge 2, j \le B(V)$
    Use Lemma 5.2 to compute $M(v,T,j)$
    If $M(v,T,j) > \psi_{|T|}$ **update** $\psi_{|T|} = M(v,T,j)$
return $\{S_i^* : \sum_{v\in S_i^*} c(v) = \psi_i, \text{for } i \in K\}$

solution S^* satisfying $\Pr[F(S^*) = OPT(F,k)] \ge 1-\epsilon$, *in time* $O(2^k e^k m B_{max}^2 \log(n/\epsilon))$, *and using space* $O(2^k n B_{max})$, *where* $B_{max} = B(V)$.

Proof. We prove below that the procedure MaxWeightP correctly returns ψ_i for $i \le k$, within the required time and space bounds. From Lemma 5.1, it follows that $\max_{i\in[1,k]}\psi_i = \max_{|S|\le k} g(C(S),B(S))$. The correctness of the algorithm follows from the proof of the recurrence in Lemma 5.2 and the bound on the success probability from Lemma 4.4. The algorithm maintains the array M, indexed by nodes, all possible color sets, and all integers in $[1,B_{\max}]$, which leads to the space complexity of $O(2^k n B_{\max})$, since there are at most 2^{k-1} possible non-empty color sets. Lemma 4.4 considers the probability of error, i.e., $\Pr[\max_j X_j \ne \psi_i]$ for one value of i. We need to consider this for k possible values. Therefore, taking $\epsilon' = \epsilon/n$ in Lemma 4.4 ensures that for each i, the probability $\Pr[\max_j X_j \ne \psi_i] \le \epsilon/n^2$, so that for all i,α, the algorithm correctly finds ψ_i. The running time is the result of solving the recurrence for each node v, for each color set T, and value B; this requires examining each possible partition $T_1 \cup T_2 = T$, $B_1 + B_2 = B$, and each neighbor $u \in \mathrm{Nbr}(v)$, which requires time $O(|\mathrm{Nbr}(v)|B2^{|T|})$. Therefore, the total running time for each coloring is $O(\sum_v \sum_{i=0}^k \sum_{j=1}^{B_{\max}} |\mathrm{Nbr}(v)|2^i j) = O(\sum_v |\mathrm{Nbr}(v)|2^k B_{\max}^2) = O(2^k m B_{\max}^2)$. Since the algorithm considers $O(e^k \log n/\epsilon)$ colorings, the running time bound follows. □

We note that by approximating $B(S)$ within a factor of $(1+\delta)$, the running time in Theorem 5.3 can be improved to $O(2^k e^k m \frac{n^2}{\delta} \log(n/\epsilon))$, while losing a constant factor in terms of the approximation. We define $\mu = \delta B_{\max}/n$, for some $\delta > 0$. Then, we define a vector $\mathbf{b'}$, where $b'(v) = \lfloor b(v)/\mu \rfloor$, for each node v. By invoking ColCodeP on the instance $(G = (V,E), \mathbf{c'}, \mathbf{b'})$, we obtain a $(1+\delta)$ approximation on the weight of S^*.

5.1.2 Scaling. There is a notion of graph refinement for parametric scan statistics analogous to the one presented for non-parametric functions. We note that the parametric functions in Table 2 are increasing on the ratio $r(S) = C(S)/B(S)$. Therefore, if we are given a set S with score $F(S)$, we

can increase the score of the set by adding any node that is a neighbor of a node already in S as long as $r(S)$ does not decrease. This idea is formalized in the following lemma.

Lemma 5.4. *Let $G(V, E)$ be a network, and let $F(\cdot)$ be a parametric scan statistic function. Given a set $S \subseteq V$, suppose there exists an node $u \notin S$ and an edge $(u, v) \in E$, for some $v \in S$, such that $C(S \cup \{u\})/B(S \cup \{u\}) \geq C(S)/B(S)$, then, $F(S \cup \{u\}) \geq F(S)$.*

Exact refinement. For parametric scan statistics, the exact refinement consists of merging nodes into components as long as the ratio $r(S)$ of the component does not decrease. Given a network $G(V, E)$ and event ($c(v)$) and population ($b(v)$) counts for every $v \in V$, we maintain a list of components or supernodes $\mathcal{V} = V_1, \ldots, V_r$ and the graph $H(V', E')$ induced by those. There is an edge between V_i and V_j if there exist nodes $v_i \in V_i$ and $v_j \in V_j$, such that v_i and v_j are neighbors in G. Initially, every node is its own component. The exact refinement iterates through the list of current components trying to merge them until no more merges are possible. In each iteration, we first sort the current components in descending order of ratio $r(S)$. Then, in that order, we merge a component with its neighbors if the ratio does not decrease. After this exact refinement, we run Algorithm 3 for the instance $(H(V', E'), \mathbf{c}', \mathbf{b}')$, where $\mathbf{c}'$ and $\mathbf{b}'$ are the event and population counts of the supernodes, respectively.

Approximate refinement. We can obtain larger components by allowing merges that decrease the ratio of the component. Our approximate refinement procedure takes two parameters: $\beta \in [0, 1]$ and $\delta > 1$. We allow a component V_i to grow as long as two constraints are not violated:

(1) *Population size constraint.* The population size of V_i is at most δ times the smallest population size in the component: $B(V_i) \leq \delta \min_{v \in V_i} b(v)$.
(2) *Event size constraint.* The event size of V_i is at least $\beta\delta$ times the largest event size in the component: $\frac{C(V_i)}{\delta} \geq \beta \max_{v \in V_i} c(v)$.

Lemma 5.5. *Let S^* be the set that maximizes a parametric scan statistic G and let $r(S^*) = \frac{C(S^*)}{B(S^*)}$. There is a solution S on the instance H constructed as above with ratio $r(S) \geq \beta r(S^*)$.*

Proof. Let S be the set formed by the supernodes V_i containing the nodes in S^*:

$$S = \{V_i : (v \in S^*) \wedge (v \in V_i)\}.$$

For simplicity, and without loss of generality, let us assume that each node of S^* is in a separate component in H. Then, we can analyze the ratio $r(S)/r(S^*)$:

$$\frac{r(S)}{r(S^*)} = \frac{\frac{C(V_1)+\cdots+C(V_{|S|})}{B(V_1)+\cdots+B(V_{|S|})}}{\frac{c(v_1)+\cdots+c(v_{|S|})}{b(v_1)+\cdots+b(v_{|S|})}}.$$

By the population size constraint δ,

$$\frac{\frac{C(V_1)+\cdots+C(V_{|S|})}{B(V_1)+\cdots+B(V_{|S|})}}{\frac{c(v_1)+\cdots+c(v_{|S|})}{b(v_1)+\cdots+b(v_{|S|})}} \geq \frac{\frac{C(V_1)+\cdots+C(V_{|S|})}{\delta b(v_1)+\cdots+\delta b(v_{|S|})}}{\frac{c(v_1)+\cdots+c(v_{|S|})}{b(v_1)+\cdots+b(v_{|S|})}} = \frac{\frac{C(V_1)+\cdots+C(V_{|S|})}{\delta}}{c(v_1)+\cdots+c(v_{|S|})}.$$

And, by the event size constraint β, we have

$$\begin{aligned}\frac{\frac{C(V_1)+\cdots+C(V_{|S|})}{\delta}}{c(v_1)+\cdots+c(v_{|S|})} &= \frac{\frac{C(V_1)}{\delta}+\cdots+\frac{C(V_{|S|})}{\delta}}{c(v_1)+\cdots+c(v_{|S|})} \\ &\geq \frac{\beta c(v_1)+\cdots+\beta c(v_{|S|})}{c(v_1)+\cdots+c(v_{|S|})} = \beta,\end{aligned}$$

so we conclude that $r(S) \geq \beta r(S^*)$. □

5.2 Functions with Node and Edge Weights

Both the heaviest subgraph (Bogdanov et al. 2011) and EventTree+ problems (Rozenshtein et al. 2014) reduce to PCST with NetWorth objective (Johnson et al. 2000). In PCST, we are given a graph $G(V, E)$ with non-negative node prizes, $\boldsymbol{\pi}$, and non-negative edge costs, $\mathbf{w}$, and the goal is to find a tree $S(V(S), E(S))$ that maximizes the NetWorth objective:

$$W(S) = \sum_{v \in V(S)} \pi(v) - \sum_{e \in E(S)} w(e).$$

Using our framework, we can design an algorithm to find a tree with maximal NetWorth and size up to k, where k is a parameter. This implies an algorithm for HS and EventTree+.

Let $M(v, T) = \max_S W(S)$, where the maximization is over all connected and colorful sets $S \subseteq V$, such that $v \in S$, $|S| = |T|$, and $\{\text{col}(u) : u \in S\} = T$. $M(v, T)$ can be computed by a dynamic program:

Lemma 5.6. *Let $M(v, T)$ be defined as above. For any node v and color s, $M(v, \{s\}) = \pi(v)$ if $col(v) = s$, else $M(v, \{s\}) = -\infty$. If $|T| \geq 2$:*

$$M(v, T) = \max_{\substack{u \in Nbr(v) \\ T_1, T_2 \subseteq T}} \{M(v, T_1) + \max\{M(u, T_2) - w(v, u), 0\}\},$$

where the maximum is over all partitions $T_1 \cup T_2$ of the set T and all neighbors u of v.

Proof. The proof is analogous to that of Lemma 4.2, but this time we have to account for the weight of the edge connecting two subsets in the recursive step. Suppose $M(v, T)$ is achieved for a connected set S, such that $|S| = |T|$ and $\{\text{col}(u) : u \in S\} = T$, with $M(v, T) = \sum_{i \in S} \pi(i)$. We claim that there exists $u \in \text{Nbr}(v)$, and partitions $T = T_1 \cup T_2$, and $S = S_1 \cup S_2$, such that (1) $M(v, T) = M(v, T_1) + M(u, T_2) - w(v, u)$, (2) $\{\text{col}(i) : i \in S_1\} = T_1$ and $\{\text{col}(i) : i \in S_2\} = T_2$, and (3) $G[S_1]$ and $G[S_2]$ are connected. Since $G[S]$ is connected, there exists a tree H that spans $G[S]$, rooted at node v. Further, there must exist a node $u \in \text{Nbr}(v)$ such that $(u, v) \in T$, since $|T| = |H| \geq 2$. Let H_1 and H_2 be the trees rooted at nodes v and u, respectively, that result when edge (u, v) is deleted in H. Let S_1 and S_2 denote the sets of nodes in H_1 and H_2, respectively. Let T_1 and T_2 be the colors used by S_1 and S_2, respectively. By construction, we have $M(v, T) = M(v, T_1) + \max\{M(u, T_2) - w(v, u), 0\}$, where the second term is negative if the weight $w(v, u)$ is greater than the NetWorth $M(u, T_2)$, in which case $M(v, T) = M(v, T_1)$. The partitions $S = S_1 \cup S_2$ and $T = T_1 \cup T_2$ satisfy the requirements mentioned earlier. Therefore, the recurrence follows. □

6 EXPERIMENTS

Our experiments address the following questions.

(1) *Optimization power.* Do our algorithms find high-scoring subgraphs in real networks and synthetic benchmarks? How do they compare with existing methods? (Section 6.3).

(2) *Event detection power.* Do our algorithms correctly identify anomalous subgraphs? How do precision and recall compare with baselines? (Section 6.4).

(3) *Scalability.* How do our algorithms scale to networks with more than 10^5 nodes? (Section 6.5).

(4) *Performance in real datasets.* How does the performance in real datasets compare with the worst case bounds? (Section 6.6).

For brevity, we focus on one scan statistic from each class as illustrative examples: (1) BJ statistic (Berk and Jones 1979) with $\alpha_{\max} = 0.15$, (2) positively elevated mean statistic (EMS) (Qian et al. 2014), and (3) the Heaviest Subgraph (HS) (Bogdanov et al. 2011) and EventTree+ (Rozenshtein et al. 2014) functions, as examples of non-parametric, parametric, and generalized functions with edge weights, respectively.

Table 4. Datasets Used in Our Experiments

Dataset	Description	Nodes	Edges	Instances
Datasets with real events				
CitHepPh	Citation network	11,895	76,284	4
NEast	Network of counties in Northeastern USA	245	683	10,000
Traffic	Traffic network of Los Angeles Country, CA	1,870	1,993	1,488
Twitter	Follower network collected through Twitter API	2,645	17,108	182
BWSN	Battle of the water sensors	12,527	14,831	22
PCST	Benchmark for the prize collecting Steiner tree problem	100 to 400	284 to 1,576	34
Datasets with planted anomalies for scalability experiments				
Email-EuAll	Email network	224,832	340,795	1
Higgs–Retweet	Retweet network	223,833	307,884	1
RoadNet-PA	Traffic network of Pennsylvania	1,088,092	1,541,898	1
Random	Erdos–Renyi graphs of	100 to	$O(100)$ to	5
	with 100 to 1,000,000 nodes	1,000,000	$O(1{,}000{,}000)$	

6.1 Datasets

We use datasets from different domains, including social networks, infrastructure networks, and standard synthetic benchmarks. Most of these datasets contain multiple instances, corresponding to snapshots of the networks at different times. A brief summary of the datasets is provided in Table 4.

CitHepPh.[1] This is a network of scientific collaborations between authors of papers submitted to the High Energy Physics—Phenomenology category of arXiv. The p-value of each node v for a specific snapshot was calculated as the ratio of nodes in the current graph snapshot whose citations are greater than or equal to the citations of this node.

NEast (Kulldorff et al. 2003). The Northeastern USA Benchmark is a well-known dataset in the spatial scan statistics community. Each node v represents one of 245 counties with a population size $b(v)$ and a synthetically generated number of infected people $c(v)$. Under the null hypothesis of "no anomalous cluster," the infected people are uniformly distributed in the counties with probability proportional to the population. That is, the number of infected people at node v follow a Poisson distribution with parameter $\lambda \cdot b(v)$, where $\lambda = C(V)/B(V)$ is the average infection rate in the graph. For the non-parametric scan statistics experiments, we define the p-value of a node as the probability of observing counts at least as extreme as $c(v)$.

Traffic.[2] The highway network of Los Angeles County, California and its activity on May, 2014. Nodes in the graph are sensors that record traffic statistics, such as average speed and the number of vehicles passing through. We assume a normal distribution for the average speed recorded by each sensor. In each snapshot t, the p-value of a node v is the cumulative distribution function of

[1] https://snap.stanford.edu/data/cit-HepPh.html.
[2] http://pems.dot.ca.gov/.

a normal distribution with mean $x_v^{[1,t-1]}$ and standard deviation $\sigma_v^{[1,t-1]}$, where $x_v^{[1,t-1]}$ and $\sigma_v^{[1,t-1]}$ are, respectively, the sample mean and standard deviation for node v from snapshots 1 to $t-1$.

Twitter. A sample of the follower graph of Venezuela collected between July 1, 2013 and December 31, 2013. We assign p-values based on the tweeting behavior of the users. Formally, let x_u^t be the number of tweets generated by node u at time t; we model x_u^t as a draw from a Poisson distribution with parameter λ_u. We take a Bayesian approach and consider λ_u to be drawn from a Gamma distribution with parameters α_u and β_u. These parameters are updated as we see new data every snapshot. The p-value of a node at time t is its posterior probability $p(n_e^t|n_e^{[1,t-1]})$, which follows a negative binomial distribution by our choice of prior.

Battle of the Water Sensor Networks (BWSN) (Ostfeld et al. 2008). This dataset is a benchmark originally used to evaluate different sensor network designs in terms of early detection of contaminants in a water system. The dataset includes "ground truth" subgraphs representing parts of the network that are contaminated, which we use for evaluation in Section 6.4.

PCST (Johnson et al. 2000). A standard benchmark to evaluate algorithms for the PCST Problem. We use the "K" instances of the benchmark to evaluate methods that reduce to PCST (Section 6.3.3).

In addition, we also consider three large networks (i.e., over 10^5 nodes) from the SNAP repository (Leskovec and Krevl 2014) to evaluate the scalability and performance-runtime tradeoff of our proposed methods (Section 6.5). Since we do not have data of events in these networks, we plant events according to the statistical assumptions of the BJ scan statistic. We select a set of seed nodes and their neighbors in the graph. With probability α, each node has p-value less than 0.05. The remaning nodes have p-value uniformly distributed in $[0, 1]$.

6.2 Baseline Methods

We compare our proposed algorithm with the following state-of-the-art methods for scan statistics. A summary of these algorithms can be found in Section 2 and Table 1.

(1) `NPHGS` (Chen and Neill 2014a): A local search heuristic for optimizing the BJ scan statistic.
(2) `AdditiveGraphScan` (GS) (Speakman et al. 2013) and `DepthFirstScan` (DFS) (Speakman et al. 2015): The state-of-the-art algorithms for optimizing parametric scan statistics that satisfy the linear-time-subset-scanning property. The EMS statistic also belongs to this category.
(3) GL (Sharpnack et al. 2013b) and `EdgeLasso` (Sharpnack et al. 2012): The representative methods for anomalous subgraph detection that optimize their own specific score functions, but are often considered as baseline methods in connected subgraph detection papers (Qian et al. 2014).
(4) `EventTree+` (Rozenshtein et al. 2014) and `MEDEN` (Bogdanov et al. 2011): For optimizing anomaly score functions with node and edge weights.

Parameter Tuning. The methods under evaluation, including ours, depend on user-specified parameters. We set k to 10 or below for our algorithms. When possible, we use the values prescribed by the authors of the method; this is the case with `NPHGS`, `EventTree+`, and `MEDEN`. In the case of `GL` and `EdgeLasso`, we tune the parameters separately for each dataset. In particular, we take a sample of 20 instances for each dataset and choose a parameter from $\{0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1\}$ that maximizes the average score. Notice that the parameter for each dataset may be different.

6.3 Optimization Power

6.3.1 Non-Parametric Scan Statistics. We compare FastColCodeNP to other algorithms on the BJ statistic. In Table 5, we report the average BJ score obtained by each method, where the average

Table 5. Non-Parametric Scan Statistics Optimization

	Berk–Jones scan statistic					
	FastColCodeNP	GS	EL	GL	DFS	NPHGS
CitHepPh	**1138.011**	1135.029	559.176	1118.352	1130.874	1118.353
NEast	**68.525**	64.214	23.366	23.211	55.541	17.696
Traffic	**128.740**	128.722	116.732	125.824	14.412	121.632
Twitter	**1722.790**	1722.388	1722.388	1722.388	1720.243	1457.410
BWSN	**602.164**	599.972	530.850	530.457	536.200	531.280

Our algorithm FastColCodeNP obtains higher scores than previous methods in real-world datasets.

Table 6. Parametric Scan Statistics Optimization

	Elevated mean scan statistic			
	FastColCodeP	GS	EL	GL
CitHepPh	**43.611**	8.578	14.959	41.830
NEast	41.903	**42.164**	5.570	7.607
Traffic	**11.763**	9.920	4.526	8.752
Twitter	**23.019**	11.337	22.660	19.110
BWSN	**109.097**	21.64	108.933	107.459

We evaluate different methods with respect to the EMS. In almost all datasets, FastColCodeP has better performace than existing methods.

is taken over all the instances in each dataset. We observe that FastColCodeNP achieves higher scores than all other methods. The difference in score is more pronounced in the NEast dataset, where FastColCodeNP more than doubles the score of `EdgeLasso` (EL) and GL. We also note that `AdditiveGraphScan` has performance close to our algorithm, which is reasonable, since this method uses a sophisticated heuristic for Steiner connectivity problems.

6.3.2 Parametric Scan Statistics. Next, we compare FastColCodeP to other methods with respect to the EMS function. Table 6 shows the average score for different datasets. We find that FastColCodeP has the best performance in all datasets, except for NEast, where `AdditiveGraphScan` (GS) scores sligthly higher.

6.3.3 Functions with Node and Edge Weights. We also test our algorithm on two objective functions for event detection that consider edge weights in addition to node weights: the HS (Bogdanov et al. 2011) and `EventTree+` (Rozenshtein et al. 2014) problems. The methods proposed in these two works are `MEDEN` (Bogdanov et al. 2011) and `GreedyT` (Rozenshtein et al. 2014), respectively. Both problem formulations reduce to the PCST problem (Johnson et al. 2000). We use our framework to design an algorithm for the PCST objective; we call this algorithm ColCodeNW, and we compare it in terms of objective score to `MEDEN` and `GreedyT` on the PCST benchmark of (Johnson et al. 2000). For `MEDEN`, we convert the PCST instances to HS instances and run the Top-Down heuristic described in (Bogdanov et al. 2011). For `GreedyT`, we convert the instances to a complete graph where an edge between two nodes has weight equal to the shortest path between the nodes, as described in (Rozenshtein et al. 2014). Figure 4 shows the scores of the two heuristics relative to the score of ColCodeNW. Our algorithm finds subgraphs of higher quality than the heuristics. The score is as much as four times higher compared to `GreedyT` and 1.5 times higher compared to `MEDEN`.

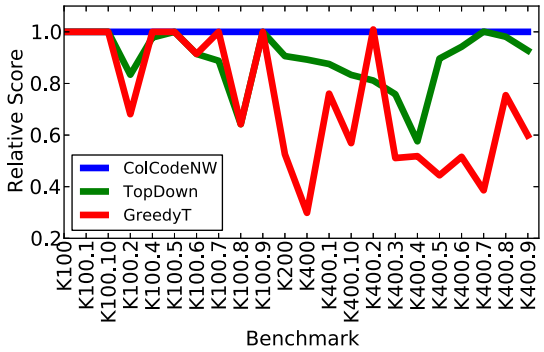

Fig. 4. PCST objective score in a set of hard PCST benchmarks. ColCodeNW finds solutions of higher quality than state-of-the-art heuristics.

Table 7. Average Precision, Recall, F1 Score, Accuracy, and Objective Value at Different Levels of Noise

	Precision				Recall				F1 score				Accuracy				$F(S)$			
				FastCol				FastCol				FastCol				FastCol				FastCol
	GS	EL	GL	CodeNP	GS	EL	GL	CodeNP	GS	EL	GL	CodeNP	GS	EL	GL	CodeNP	GS	EL	GL	CodeNP
0%	**0.980**	0.999	0.901	0.977	0.943	0.856	0.856	**0.955**	0.948	0.895	0.820	**0.952**	0.966	0.855	0.820	**0.973**	599.972	530.850	530.457	**602.164**
2%	**0.974**	0.991	0.995	0.973	0.967	0.796	0.772	**0.975**	**0.970**	0.854	0.842	0.957	0.946	0.789	0.769	**0.950**	579.197	427.984	437.783	**580.977**
4%	0.945	0.985	0.984	**0.966**	0.955	0.687	0.663	**0.971**	0.952	0.775	0.757	**0.963**	0.912	0.678	0.652	**0.929**	565.363	393.930	387.914	**571.231**
6%	**0.959**	0.964	0.973	0.954	0.937	0.567	0.542	**0.953**	0.946	0.683	0.664	**0.953**	0.901	0.558	0.536	**0.912**	522.694	318.593	300.118	**531.497**
8%	0.928	0.960	0.966	**0.931**	0.888	0.561	0.502	**0.919**	0.905	0.670	0.626	**0.923**	0.830	0.544	0.490	**0.860**	483.127	315.720	291.227	**491.657**

6.4 Event Detection Power

Now, we evaluate FastColCodeNP in terms of event detection power. We use the ground truth provided with the BWSN dataset, and we evaluate in terms of accuracy, precision, recall, and the F1 score. Let R be the set of nodes in the anomalous subgraphs and let S be the detected subgraph; then, we define

(1) $\text{Accuracy}(R,S) = \frac{|R \cap S|}{|R \cup S|}$,
(2) $\text{Precision}(R,S) = \frac{|R \cap S|}{|S|}$,
(3) $\text{Recall}(R,S) = \frac{|R \cap S|}{|R|}$, and
(4) $\text{F1 score} = 2\left(\frac{\text{Precision}(R,S) \cdot \text{Recall}(R,S)}{\text{Precision}(R,S) + \text{Recall}(R,S)}\right)$.

In order to assess the performance of our method under noise, we introduce a random percentage of uniform noise in each instance. For a given *noise level* l, each non-significant node becomes significant with probability l.

In Table 7, we compare the performance of FastColCodeNP with other algorithms at different noise levels. As in the previous section, we observe that our algorithm achieves higher objective scores compared to other methods, even when noise is present. As for the event detection power, FastColCodeNP has higher accuracy and recall for all the noise levels and higher F1 score at almost every level. Finally, we note that the results in this section provide evidence that *better objective scores also lead to better detection power, thus the importance of algorithms with good theoretical bounds.*

Table 8. Performace-Runtime Tradeoff in Large Datasets for Non-Parametric Scan Statistic Evaluation

	Berk–Jones scan statistic (Time in seconds)					
	FastColCodeNP	GS	EL	GL	DFS	NPHGS
Email-EuAll	**420.46** (2,376)	420.46 (20,254)	275.08 (679)	–	392.53 (3,671)	275.08 (10)
Higgs-Retweet	**839.18** (1,015)	839.18 (32,340)	421.16 (585)	–	721.70 (3,213)	421.16 (5)
RoadNet-PA	**24.66** (22)	–	24.66 (7,919)	–	21.22 (1,584)	13.28 (15)

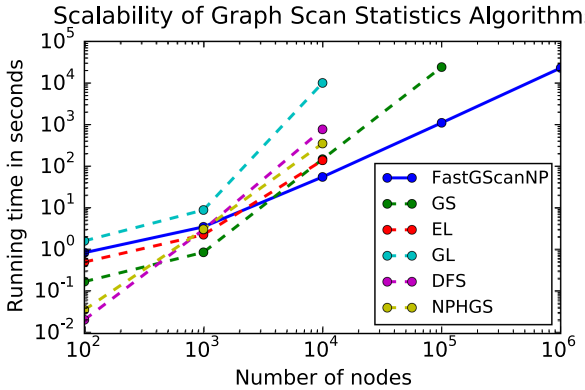

Fig. 5. Comparison of running time as a function of number of nodes in Erdos–Renyi graphs.

6.5 Scalability

With the techniques presented in Section 4.4, we are able to run FastColCodeNP in networks with over one million nodes. We note that in previous work only networks of up to 80,000 nodes have been considered. In Table 8, we compare our algorithm to existing methods in terms of objective score and running time. First, we compare FastColCodeNP to the best-performing heuristic: `AdditiveGraphScan` (GS). We note that both methods achieve the same score in all datasets, but the running time of the latter is one order of magnitude larger, and it did not complete after 10 hours for the road network. Second, we observe that GL does not scale to large networks due to the fact high time and space complexity of constrained quadratic programming methods. We also note that our algorithm is much faster on the road network than on the other two because nodes in this planar network have low degree, thus making our low-radius technique more effective. Finally, `EdgeLasso`, `DFS`, and `NPHGS` are faster than FastColCodeNP; however, the scores obtained are significantly lower than using our algorithm.

In Figure 5, we show the scalability of all the algorithms we consider as a function of graph size in the Random dataset. FastColCodeNP is faster than other methods for graph of size 10^4 and above, and it was the only algorithm to run to completion on graphs with one million nodes within 24 hours.

6.6 Performance Guarantees in Real Datasets

Graph refinement. Our graph refinement operation reduces the number of significant nodes in real datasets to less than a third of the original number. In Table 9, we report the number of significant nodes before and after graph refinement for $\beta \in \{1, 0.95, 0.90\}$. Each dataset initially contains hundreds of significant nodes, with Twitter being close to 1,000. However, after graph refinement, this number goes down to less than 100. With approximate refinement, we are able to further reduce the effective number of significant nodes, down to a single digit in most cases. Because there are

Table 9. Number of Significant Nodes with Graph Refinement

Dataset	$W(S,\alpha)$ ($\alpha = 0.15$)	Graph refinement		
		$\beta = 1$	$\beta = 0.95$	$\beta = 0.90$
CitHepPh	624	20	3	1
NEast	65	24	24	21
Traffic	157	61	61	60
Twitter	991	80	2	1
BWSN	333	4	2	2

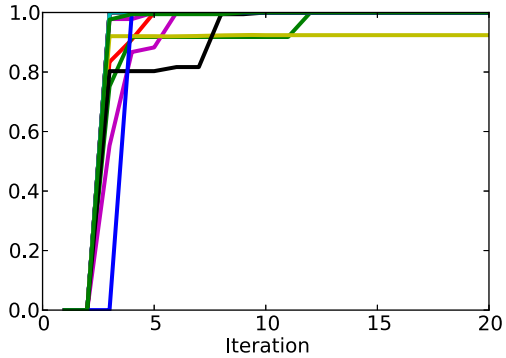

Fig. 6. Score of the solution found (normalized by that of the optimum) as a function of the number of iterations.

only a few significant supernodes, solutions of high score are small. In fact, if we set k to 10 or less in FastColCodeNP, we are able to discover solutions of higher scores than previous methods (Section 6.3).

Convergence in few iterations. Algorithm FastColCodeNP uses $\ell = e^k \log n^2/\epsilon$ random colorings to guarantee a solution with probability $1 - \epsilon$. In practice, we find the number of colorings needed is much smaller—this is shown in Figure 6 for the PCST benchmark. Each line in the plot represents the solution obtained for one instance of size 100; the y-axis shows the objective value obtained normalized by the objective obtained in the last iteration of the algorithm. The theoretical bound requires 3,285 random colorings to guarantee 95% probability. However, the best solution is found in less than 20 iterations for all instances, and sooner for most of them. We observed similar results in the other networks in Table 4.

7 APPLICATION

We use our methods for event detection in Twitter follower graphs of Mexico and Venezuela. We model interactions—i.e., retweets and replies—between users as Poisson counts. The weight of an edge is given by $-\log(p(n_e^t|n_e^{[1,t-1]})/\mu$, where $p(n_e^t|n_e^{[1,t-1]})$ is the posterior probability of seeing n_e^t interactions given the counts in the previous days, and $\mu = 0.05$ is a significance threshold. This weighing function (Mongiovì et al. 2013) is positive-increasing if the posterior probability is less than μ and negative-decreasing otherwise. After assigning weights to edges, we solve the HDS problem (Bogdanov et al. 2011) using algorithm ColCodeNW (See Section 6.3.3). For Venezuela, the temporal anomalous subgraph spans the time period between January 4, 2014 and March 31, 2014, which was a time of nationwide protests against the central government of that country. We also extracted the tweets corresponding to the heavy subgraphs, and we find that these tweets

(a)

(b)

(c)

(d)

Fig. 7. Examples of events found in the heavy subgraphs for Venezuela (top) and Mexico (bottom) using our methods.

contain chatter about important national events. Figure 7(a) shows one such event. The predominant terms of the tweets relate to a protest organized in the city of Tachira demanding the liberation of local students, who had been put in jail in the previous days for protesting. For Mexico, the two most common words in the extracted tweets form the phrase "energy reform" referring to a bill recently proposed by Mexican president Enrique Pena Nieto that would have a significant impact in the economy of the country (Figure 7(c)).

8 CONCLUSIONS

Anomaly detection is a fundamental task in network analysis with a large number of applications to different domains, and scan statistics is one particular methodology that has been widely applied for this task.

The detection power of methods based on scan statistics is reliant on the degree to which we can optimize a given "anomalousness" function over connected subgraphs. This connectivity constraint makes the problem very challenging, compared to the looser constraint of spatial adjacency that has been studied extensively. In fact, here, we show strong connections to the classical Steiner Tree problem in graph theory.

In order to tackle the computational complexity, recent papers have proposed various heuristics for scan statistics on graphs, which have been shown to have good empirical performance while, at the same time, being very efficient and scalable. However, one notable drawback of heuristics is that they seldom offer rigorous theoretical guarantees on the quality of the solutions discovered. Because of the lack of guarantees, heuristics may perform poorly in some datasets, and this will affect the ability to detect anomalous events in the graph. Furthermore, the lack of guarantees also

makes it challenging to compare two competing heuristics. It is not clear whether one method is always superior to the other or whether both are better in different parts of the problem space.

Instead, we propose algorithms based on parameterized complexity. We find that fixed parameter tractability is a powerful, but underexplored approach for designing efficient algorithms, and it is likely to be useful in other graph mining applications. We present a unified framework for optimizing a broad class of graph scan statistics with connectivity constraints, with the following novel characteristics: (1) it gives rigorous guarantees for a large class of parametric and non-parametric score functions, and (2) it can be scaled to large graphs with over a million nodes.

Our methods will lead to direct improvements in performance quality for other uses of graph scan statistics in different applications.

APPENDIX

A HARDNESS OF MAXIMIZING BERK–JONES STATISTIC

We present the proof of hardness for maximizing the BJ statistic over connected subgraphs. A similar style of proof can be used to establish hardness results for other functions in Table 2. Through the results below, we formalize connections between scan statistic optimization and Steiner connectivity, which is of both theoretical and practical interest.

The BJ statistic for a subset of nodes S is defined as

$$F(S) = \max_{\alpha \leq \alpha_{max}} |S| \cdot KL(W(S,\alpha)/|S|, \alpha).$$

Here, each node has a p-value, $p(v) \in [0,1]$, $W(S,\alpha)$ is the number of nodes in S with p-value at most α, and $\alpha_{\max}$ is a parameter. $KL(\cdot,\cdot)$ is the truncated KL-divergence:

$$\mathrm{KL}(\delta,\gamma) = \begin{cases} 0 & 0 \leq \delta < \gamma \\ \delta \log\left(\frac{\delta}{\gamma}\right) + (1-\delta)\log\left(\frac{1-\delta}{1-\gamma}\right) & \gamma \leq \delta < 1 \\ \log\left(\frac{1}{\gamma}\right) & \delta = 1 \end{cases}.$$

Sometimes, we will only be interested in the number of nodes with p-value at most α and not in the particular set being evaluated. In those cases, we will consider the following version of the BJ statistic:

$$F'(i,j,\alpha) = (i+j) \cdot KL(i/(i+j),\alpha) = \begin{cases} 0 & 0 \leq i/(i+j) < \alpha \\ i\log\left(\frac{i/(i+j)}{\alpha}\right) + j\log\left(\frac{j/(i+j)}{1-\alpha}\right) & \alpha \leq i/(i+j) < 1 \\ i\log\left(\frac{1}{\alpha}\right) & i/(i+j) = 1 \end{cases}.$$

Notice that $F(\cdot)$ can be computed from $F'(\cdot)$ by letting $F(S) = \max_{\alpha \leq \alpha_{\max}} F'(W(S,\alpha), |S| - W(S,\alpha), \alpha)$.

Lemma A.1. *The function $F'(i,j,\alpha)$ is increasing on i and decreasing on j when $\frac{i}{i+j}$ is in the interval $(\alpha, 1)$.*

Proof. The derivative of $F'(\cdot)$ with respect to i is

$$\frac{\partial F'}{\partial i} = \log\left(\frac{i}{i+j}\right) - \log(\alpha),$$

which is greater than 0 in the desired interval. Similarly, the derivative of $F'(\cdot)$ with respect to j is

$$\frac{\partial F'}{\partial j} = \log\left(\frac{j}{i+j}\right) - \log(1-\alpha),$$

which is less than 0 in the interval. □

In the decision version of anomalous subgraph detection, we want to find whether or not there is a connected subgraph with objective value $F(\cdot)$ at least some lower bound.

Problem 3 (BJ-Decide (BJ-D)). ***Given:*** *an undirected graph $G = (V, E)$ with p-values, $p(v) \in [0, 1]$, a parameter α_{max}, and a parameter $\tau \geq 0$.* ***Decide:*** *Is there a connected set of nodes $S \subseteq V$, such that $F(S) \geq \tau$?*

For the proof, we will consider the node version of the Steiner Tree problem.

Problem 4 (Steiner Tree (ST)). ***Given:*** *an undirected graph $G = (V, E)$, a set of terminals $T \subset V$, such that $|T| = a$, and a parameter $b \geq 0$.* ***Decide:*** *Is there a connected set of nodes $S \subseteq V$, such that $|S| \leq a + b$ and $T \subseteq S$? i.e., Is it possible to connect all the terminals using at most b non-terminal nodes?*

Proof. (of Theorem 3.1) BJ-D is the decision version of this problem. First, we note that **BJ-D is in NP**. Given any set of nodes S, we can verify that the nodes are connected in time polynomial in the size of the input graph, and we can evaluate $F(S)$ in time $O(|S|^2)$ to check whether or not $F(S) \geq \tau$. The $O(|S|^2)$ bound comes from the facts that (1) for a fixed α, we have to compute $W(S, \alpha)$, which involves checking whether each node has p-value at most α or not—this takes $O(|S|)$ time and (2) we have to evaluate the function for at most $|S|$ different values of α.

Now, we show that *ST is polynomial-time reducible to BJ-D.* Given an instance $(G = (V, E), T, b)$ of ST, we construct an instance $(H = (V', E'), p, \alpha_{\max}, \tau)$ of BJ-D as follows. Let n be the number of nodes in G; H is going to be a graph with the same nodes and edges as G, but, in addition, we are going to attach $n^2 - 1$ new vertices to each terminal node. We will refer to these new vertices as *spokes*. Formally, for a terminal $t \in T$, let $S^t = \{s_1^t, \ldots, s_{n^2-1}^t\}$; then, $V' = V \cup \{S^t | \forall t \in T\}$ and $E' = E \cup \{(t, s_1^t), \ldots, (t, s_{n^2-1}^t) | \forall t \in T\}$. For the p-values, the terminals and the spokes have $p(v) = 1/2$; all other nodes have p-value 1. Finally, we let $\alpha_{\max} = 1/2$ and $\tau = F'(an^2, b, 1/2)$, where a is the number of terminals and b is the parameter in the instance of ST. This reduction is illustrated in Figure 8. □

Claim. *There is a Steiner tree S_G with b non-terminal nodes in G if and only if there is a connected set of nodes S_H with BJ score τ in H.*

Proof. The *first direction*—i.e., S_G implies S_H—is straightforward. By construction of H, S_G is a connected set of nodes in H. Let S_H be the set formed by S_G and all the spokes in the graph. That is, $S_H = S_G \cup \{S^t | \forall t \in T\}$. S_H has $b + a + a(n^2 - 1)$ vertices—i.e., the non-terminals, the terminals, and the spokes. Out of those, an^2 nodes have p-value $1/2$ and the remaining b have p-value 1. Therefore, this subgraph has BJ score of

$$F(S_H) = \max_{\alpha \leq (1/2)} F'(an^2, b, \alpha) = F'(an^2, b, 1/2) = \tau.$$

To prove the *converse*, notice that a connected subgraph S_H with $F(S_H) \geq \tau$ has at most b nodes with p-value greater than $1/2$. By construction, these b nodes correspond to non-terminal nodes in G. All that is left to show is all the terminals are included in S_H, as this implies that G contains a connected graph with all the terminals and at most b non-terminals.

To see that S_H must in fact have all the terminal nodes, consider the highest scoring subgraph that *does not* include some terminal. In the best possible case, we would be able to connect the remaining $(a - 1)$ terminals and their respective spokes without using any Steiner nodes. Such subgraph would have a score of

$$F'((a - 1)n^2, 0, 1/2) = (a - 1)n^2 \log(2).$$

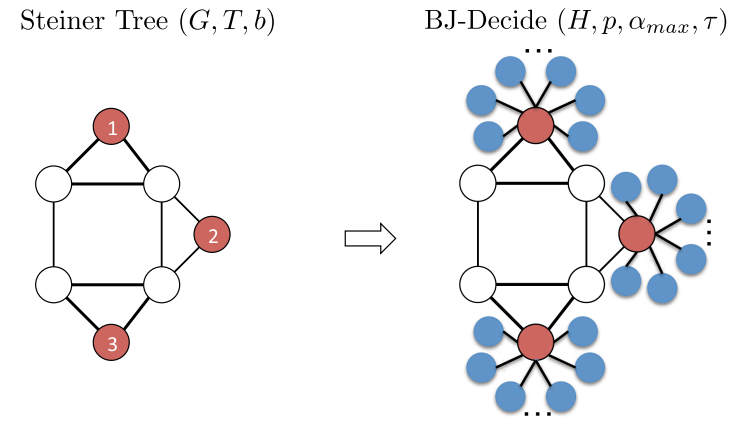

Fig. 8. An example of the reduction in Theorem 3.1. For each terminal in the instance of Steiner Tree, we add $n^2 - 1$ spokes to the corresponding node in the instance of BJ-Decide. There is a Steiner Tree containing at most b non-terminal nodes in G if and only if there is a connected subgraph with BJ objective at least $\tau = F'(an^2, b, 1/2)$ in H.

We want to compare this value to τ, which is given by

$$\begin{aligned}\tau &= F'(an^2, b, 1/2)\\ &\geq F'(an^2, n-a, 1/2)\\ &= an^2 \log\left(2 \times \frac{an^2}{an^2+n-a}\right) + (n-a)\log\left(2 \times \frac{n-a}{an^2+n-a}\right),\end{aligned}$$

where the inequality holds because (1) $F'(\cdot)$ is decreasing on b (Lemma A.1) and (2) in the worst case, we need to use all the vertices in G to connect the terminals—i.e., $(n-a)$ non-terminal nodes. Now, we have a lower bound on τ and an upper bound on the best score that does not have all the terminals. We want to show that

$$an^2 \log\left(2 \times \frac{an^2}{an^2+n-a}\right) + (n-a)\log\left(2 \times \frac{n-a}{an^2+n-a}\right) > (a-1)n^2\log(2) \tag{3}$$

$$an^2 \log\left(\frac{an^2}{an^2+n-a}\right) + (n-a)\log\left(\frac{n-a}{an^2+n-a}\right) > (n^2+n-a)\log(1/2), \tag{4}$$

where we get inequality (4) by moving the $an^2\log(2)$ and $(n-a)\log(2)$ terms to the right-hand side and rearranging. We show that the following two inequalities, which together imply (4), hold for sufficiently large n.

$$an^2 \log\left(\frac{an^2}{an^2+n-a}\right) > \frac{1}{2}(n^2+n-a)\log(1/2) \tag{5}$$

$$(n-a)\log\left(\frac{n-a}{an^2+n-a}\right) > \frac{1}{2}(n^2+n-a)\log(1/2). \tag{6}$$

Proving (5):

$$an^2 \log\left(\frac{an^2}{an^2+n-a}\right) > \frac{1}{2}(n^2+n-a)\log(1/2)$$
$$an^2 \log\left(\frac{an^2+n-a}{an^2}\right) < \frac{1}{2}(n^2+n-a)\log(2)$$
$$\log\left(1+\frac{n-a}{an^2}\right) < \frac{1}{2a}(1+\frac{n-a}{n^2})\log(2)$$
$$1+\frac{n-a}{an^2} < 2^{\frac{1}{2a}(1+\frac{n-a}{n^2})}.$$

Because $1 < a < n$, we can express a as βn, such that $1 > \beta > 1/n$. Replacing, we obtain

$$1+\frac{n-\beta n}{(\beta n)n^2} < 2^{\frac{1}{2}(\frac{1}{\beta n}+\frac{n-\beta n}{(\beta n)n^2})} \tag{7}$$
$$1+\frac{1-\beta}{\beta n^2} < 2^{\frac{1}{2}(\frac{1}{\beta n}+\frac{1-\beta}{\beta n^2})} \tag{8}$$
$$1+\frac{\delta}{n^2} < 2^{\frac{1}{2}(\frac{1}{\beta n}+\frac{\delta}{n^2})} \qquad \left(\text{let } \delta = \frac{1-\beta}{\beta}\right) \tag{9}$$
$$2^{\frac{1}{2}(1-\frac{1}{\beta n})}\left(1+\frac{\delta}{n^2}\right) < 2^{\frac{1}{2}(1+\frac{\delta}{n^2})} \qquad \text{(multiply both sides by } 2^{\frac{1}{2}(1-\frac{1}{\beta n})}). \tag{10}$$

Let $\epsilon = \log_2(1+\delta/n^2)$, so that $1+\delta/n^2 = 2^\epsilon$; then, inequality (10) holds for

$$2^{\frac{1}{2}(1-\frac{1}{\beta n})}2^\epsilon < 2^{\frac{1}{2}}2^\epsilon \tag{11}$$
$$\frac{1}{2}\left(1-\frac{1}{\beta n}\right)+\epsilon < 2^{\epsilon-1} \tag{12}$$
$$1-\frac{1}{\beta n}+2\epsilon < 2^\epsilon. \tag{13}$$

Substituting $1+\delta/n^2 = 2^\epsilon$ into (13), we obtain

$$1-\frac{1}{\beta n}+2\epsilon < 1+\frac{\delta}{n^2}$$
$$2\log\left(1+\frac{\delta}{n^2}\right) < \frac{\delta}{n^2}+\frac{1}{\beta n}.$$

Since $(1-\beta) < 1$, we prove the stronger inequality $2\log(1+\delta/n^2) < \frac{\delta}{n^2}+\frac{\delta}{n}$ by analyzing the growth rate of both functions:

$$\lim_{n\to\infty}\frac{\frac{\delta}{n^2}+\frac{\delta}{n}}{2\log(1+\frac{\delta}{n^2})} = \lim_{n\to\infty}\frac{-\frac{2\delta}{n^3}-\frac{\delta}{n^2}}{2\frac{1}{1+\frac{\delta}{n^2}}\frac{-2\delta}{n^3}} \qquad \text{(taking the derivative of both functions)}$$
$$= \lim_{n\to\infty}\frac{2+n}{\frac{4}{1+\frac{\delta}{n^2}}} \qquad \text{(taking out } n^3 \text{ and } \delta)$$
$$= \lim_{n\to\infty}\frac{2+n}{\frac{4n^2}{n^2+\delta}}$$
$$= \lim_{n\to\infty} n/4 = \infty.$$

Proving (6):

$$(n-a)\log\left(\frac{n-a}{an^2+n-a}\right) > \frac{1}{2}(n^2+n-a)\log(1/2)$$
$$(n-a)\log\left(\frac{an^2+n-a}{n-a}\right) < \frac{1}{2}(n^2+n-a)\log(2)$$
$$\log\left(1+\frac{an^2}{n-a}\right) < \frac{1}{2}\left(1+\frac{n^2}{n-a}\right)\log(2)$$
$$1+\frac{an^2}{n-a} < 2^{\frac{1}{2}(1+\frac{n^2}{n-a})}.$$

By letting $a = \beta n$ for $1 > \beta > 1/n$,

$$1+\frac{(\beta n)n^2}{n-\beta n} < 2^{\frac{1}{2}(1+\frac{n^2}{n-\beta n})} \tag{14}$$
$$1+\frac{\beta n^2}{1-\beta} < 2^{\frac{1}{2}(1+\frac{n}{1-\beta})} \tag{15}$$
$$1+\delta n^2 < 2^{\frac{1}{2}(1+\frac{n}{1-\beta})} \qquad \left(\text{let } \delta = \frac{\beta}{1-\beta}\right) \tag{16}$$
$$2^{\frac{1}{2}(\delta n^2-\frac{n}{1-\beta})}(1+\delta n^2) < 2^{\frac{1}{2}(1+\delta n^2)} \qquad \text{(multiply both sides by } 2^{\frac{1}{2}(\delta n^2-\frac{n}{1-\beta})}\text{)}. \tag{17}$$

Let $1+\delta n^2 = 2^\epsilon$. Then, inequality (17) holds for

$$2^{\frac{1}{2}(\delta n^2-\frac{n}{1-\beta})}2^\epsilon < 2^{\frac{1}{2}2^\epsilon} \tag{18}$$
$$\frac{1}{2}\left(\delta n^2-\frac{n}{1-\beta}\right)+\epsilon < 2^{\epsilon-1} \tag{19}$$
$$\delta n^2-\frac{n}{1-\beta}+2\epsilon < 2^\epsilon. \tag{20}$$

Substituting $1+\delta n^2 = 2^\epsilon$ into (20), we obtain

$$\delta n^2-\frac{n}{1-\beta}+2\epsilon < (1+\delta n^2)$$
$$n > (2\epsilon-1)(1-\beta)$$
$$n > (2\log_2(1+\delta n^2)-1)(1-\beta).$$

The stricter inequality $n > (2\log_2(1+\delta n^2)-1)$ is true for sufficiently large n, since the function on the left of the ">" sign grows faster. □

We have shown that BJ-D is in NP and that ST is polynomial-time reducible to BJ-D. This completes the proof. □

ACKNOWLEDGMENTS

The authors would like to thank Elizabeth Tran for revising earlier versions of this work.

REFERENCES

Deepak Agarwal, Andrew McGregor, Jeff M. Phillips, Suresh Venkatasubramanian, and Zhengyuan Zhu. 2006. Spatial scan statistics: Approximations and performance study. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. Oddball: Spotting anomalies in weighted graphs. In *Proceedings of the Advances in Knowledge Discovery and Data Mining*. Springer, 410–421.

Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: A survey. *Data Mining and Knowledge Discovery* 29, 3 (2015), 626–688.

Noga Alon, Raphael Yuster, and Uri Zwick. 1995. Color-coding. *Journal of the ACM* 42, 4 (1995), 844–856.

Gelio Alves and Yi-Kuo Yu. 2014. Accuracy evaluation of the unified *p*-value from combining correlated *p*-values. *PLoS ONE* 9, 3 (2014), e91225.

E. Awini, P. Mattah, O. Sankoh, and M. Gyapong. 2010. Spatial variations in childhood mortalities at the Dodowa Health and Demographic Surveillance System site of the INDEPTH network in Ghana. *Tropical Medicine & International Health* 15, 5 (2010), 520–528.

Robert H. Berk and Arthur Cohen. 1979. Asymptotically optimal methods of combining tests. *Journal of the American Statistical Association* 74, 368 (1979), 812–814.

R. H. Berk and D. H. Jones. 1979. Goodness-of-fit test statistics that dominate the Kolmogorov statistics. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 47, 1 (1979), 47–59.

P. Bogdanov, M. Mongiovì, and A. Singh. 2011. Mining heavy subgraphs in time-evolving networks. In *IEEE 11th International Conference on Data Mining*. IEEE, 81–90.

Francis P. Boscoe, Thomas O. Talbot, and Martin Kulldorff. 2016. Public domain small-area cancer incidence data for New York state, 2005–2009. *Geospatial Health* 11, 1 (2016), 304.

Jose Cadena, Feng Chen, and Anil Vullikanti. 2018. Graph anomaly detection based on Steiner connectivity and density. *Proceedings of IEEE* 106, 5 (2018), 829–845. DOI: https://doi.org/10.1109/JPROC.2018.2813311

Feng Chen and Daniel Neill. 2014b. Non-parametric scan statistics for disease outbreak detection on Twitter. *Online Journal of Public Health Informatics* 6, 1 (2014), e155.

Feng Chen and Daniel Neill. 2014a. 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*.

Feng Chen and Daniel B. Neill. 2015. Human rights event detection from heterogeneous social media graphs. *Big Data* 3, 1 (2015), 34–40.

Jing Dai, Feng Chen, Sambit Sahu, and Milind Naphade. 2010. Regional behavior change detection via local spatial scan. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 490–493.

Qi Ding, Natallia Katenka, Paul Barford, Eric D. Kolaczyk, and Mark Crovella. 2012. Intrusion as (anti)social communication: Characterization and detection. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 886–894.

David Donoho and Jiashun Jin. 2004. Higher criticism for detecting sparse heterogeneous mixtures. *The Annals of Statistics* 32, 3 (2004), 962–994.

David Donoho and Jiashun Jin. 2015. Special Invited Paper: Higher criticism for large-scale inference, especially for rare and weak effects. *Statistical Science* 30, 1 (2015), 1–25.

Luiz Duczmal, Martin Kulldorff, and Lan Huang. 2006. Evaluation of spatial scan statistics for irregularly shaped clusters. *Journal of Computational and Graphical Statistics* 15, 2 (2006), 428–442.

W. Eberle and L. Holder. 2009. Graph-based approaches to insider threat detection. In *Proceedings of the Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*.

E. S. Edgington. 1972. An additive method for combining probability values from independent experiments. *The Journal of Psychology* 80, 2 (1972), 351–363.

F. Eicker. 1979. The asymptotic distribution of the suprema of the standardized empirical processes. *The Annals of Statistics* 7, 1 (1979), 116–138.

R. A. Fisher. 1925. *Statistical Methods for Research Workers*. Edinburgh Oliver & Boyd.

Michel X. Goemans and David P. Williamson. 1995. A general approximation technique for constrained forest problems. *SIAM Journal on Computing* 24, 2 (1995), 296–317.

Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. FRAUDAR: Bounding graph fraud in the face of camouflage. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, 895–904. DOI: https://doi.org/10.1145/2939672.2939747

Leah Jager and Jon A. Wellner. 2007. Goodness-of-fit tests via phi-divergences. *The Annals of Statistics* 35, 5 (2007), 2018–2053.

Jiashun Jin and Tracy Ke. 2016. Rare and weak effects in large-scale inference: Methods and phase diagrams. *Statistica Sinica* (2016), 1–34.

D. Johnson, M. Minkoff, and S. Phillips. 2000. The prize collecting Steiner tree problem: Theory and practice. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*.

Inkyung Jung, Martin Kulldorff, and Otukei John Richard. 2010. A spatial scan statistic for multinomial data. *Statistics in Medicine* 29, 18 (2010), 1910–1918.

Amin Vahedian Khezerlou, Xun Zhou, Lufan Li, Zubair Shafiq, Alex X. Liu, and Fan Zhang. 2017. A traffic flow approach to early detection of gathering events: Comprehensive results. *ACM Transactions on Intelligent Systems and Technology* 8, 6 (2017), 74.

Martin Kulldorff. 1997. A spatial scan statistic. *Communications in Statistics: Theory and Methods* 26, 6 (1997), 1481–1496.

Martin Kulldorff, Toshiro Tango, and Peter J. Park. 2003. Power comparisons for disease clustering tests. *Computational Statistics & Data Analysis* 42, 4 (2003), 665–684.

Jure Leskovec and Andrej Krevl. 2014. SNAP datasets: Stanford large network dataset collection. Retrieved from http://snap.stanford.edu/data.

Florence Margai and Norah Henry. 2003. A community-based assessment of learning disabilities using environmental and contextual risk factors. *Social Science & Medicine* 56, 5 (2003), 1073–1085.

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

Misael Mongiovi, Petko Bogdanov, Razvan Ranca, Evangelos E. Papalexakis, Christos Faloutsos, and Ambuj K. Singh. 2013. Netspot: Spotting significant anomalous regions on dynamic networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 28–36.

Daniel B. Neill. 2008. Fast and flexible outbreak detection by linear-time subset scanning. *Advances in Disease Surveillance* 5 (2008), 48.

Daniel B. Neill. 2009. An empirical comparison of spatial scan statistics for outbreak detection. *International Journal of Health Geographics* 8, 1 (2009), 20.

Daniel B. Neill. 2012. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74, 2 (2012), 337–360.

Daniel B. Neill and Jeff Lingwall. 2007. A nonparametric scan statistic for multivariate disease surveillance. *Advances in Disease Surveillance* 4 (2007), 106.

Avi Ostfeld, James G. Uber, Elad Salomons, Jonathan W. Berry, William E. Hart, Cindy A. Phillips, Jean-Paul Watson et al. 2008. The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms. *Journal of Water Resources Planning and Management* 134, 6 (2008), 556–568.

Jing Qian, Venkatesh Saligrama, and Yuting Chen. 2014. Connected sub-graph detection. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*.

Polina Rozenshtein, Aris Anagnostopoulos, Aristides Gionis, and Nikolaj Tatti. 2014. Event detection in activity networks. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* ACM, 1176–1185.

James Sharpnack, Akshay Krishnamurthy, and Aarti Singh. 2013a. Near-optimal anomaly detection in graphs using Lovasz extended scan statistic. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*.

James Sharpnack, Aarti Singh, and Alessandro Rinaldo. 2012. Sparsistency of the edge Lasso over graphs. In *Proceedings of the Artificial Intelligence and Statistics*.

James Sharpnack, Aarti Singh, and Alessandro Rinaldo. 2013b. Changepoint detection over graphs with the spectral scan statistic. In *Procedings of the Artificial Intelligence and Statistics*.

Skyler Speakman et al. 2015. Scalable detection of anomalous patterns with connectivity constraints. *Journal of Computational and Graphical Statistics* 24, 4 (2015), 1014–1033.

Skyler Speakman, Yating Zhang, and Daniel B. Neill. 2013. Dynamic pattern detection with temporal consistency and connectivity constraints. In *Proceedings of the 13th International Conference on Data Mining*. IEEE, 697–706

Samuel A. Stouffer, Edward A. Suchman, Leland C. DeVinney, Shirley A. Star, and Robin M. Williams. 1949. *The American Soldier. Adjustment During Army Life*. Princeton University Press.

Kunihiko Takahashi, Martin Kulldorff, Toshiro Tango, and Katherine Yih. 2008. A flexibly shaped space-time scan statistic for disease outbreak detection and monitoring. *International Journal of Health Geographics* 7, 1 (2008), 14.

Pavla Vaneckova, Paul J. Beggs, and Carol R. Jacobson. 2010. Spatial analysis of heat-related mortality among the elderly between 1993 and 2004 in Sydney, Australia. *Social Science & Medicine* 70, 2 (2010), 293–304.

R. Wilcox. 2005. Kolmogorov–Smirnov test. *Encyclopedia of Biostatistics*, P. Armitage and T. Colton (Eds.). DOI: 10.1002/0470011815.b2a15064

April M. Zeoli, Jesenia M. Pizarro, Sue C. Grady, and Christopher Melde. 2014. Homicide as infectious disease: Using public health methods to investigate the diffusion of homicide. *Justice quarterly* 31, 3 (2014), 609–632.

Received October 2017; revised November 2018; accepted January 2019