Tradeoffs between Density and Size in Extracting Dense Subgraphs: A Unified Framework

Zhefeng Wang*, Lingyang Chu†, Jian Pei†, Abdullah Al-Barakati‡ and Enhong Chen*
*School of Computer Science and Technology
University of Science and Technology of China, Hefei, China
Email: zhefwang@mail.ustc.edu.cn, cheneh@ustc.edu.cn
†School of Computing Science
Simon Fraser University, Burnaby, Canada
Email: lca117@sfu.ca, jpei@cs.sfu.ca
‡King Abdulaziz University, Jeddah, Saudi Arabia
Email: aaalbarakati@kau.edu.sa

***Abstract*—Extracting dense subgraphs is an important step in many graph related applications. There is a challenging struggle in exploring the tradeoffs between density and size in subgraphs extracted. More often than not, different methods aim at different specific tradeoffs between the two factors. To the best of our knowledge, no existing method can allow a user to explore the full spectrum of the tradeoffs using a single parameter. In this paper, we investigate this problem systematically. First, since the existing studies cannot find highly compact dense subgraphs, we formulate the problem of finding very dense but relatively small subgraphs. Second, we connect our problem with the existing methods and propose a unified framework that can explore the tradeoffs between density and size of dense subgraphs extracted using a hyper-parameter. We give theoretical upper and lower bounds on the hyper-parameter so that the range where the unified framework can produce non-trivial subgraphs is determined. Third, we develop an efficient quadratic programming method for the unified framework, which is a generalization and extension to the existing methods. We show that optimizing the unified framework is essentially a relaxation of the maximization of a family of density functions. Last, we report a systematic empirical study to verify our findings.**

I. Introduction

A dense subgraph is a set of vertices in a graph that are highly connected to each other. Extracting dense subgraphs is a key step in many graph related applications. For example, in social networks, a dense subgraph may correspond to a closely connected community [1], [2], [3]. In bioinformatics, extracting dense subgraphs can be used to find regulatory motifs in genomic DNA [4] and reveal correlations among genes [5]. In multimedia analysis, a dense subgraph can represent a robust common visual pattern in similar images [6]. In machine learning, the one-class clustering problem [7] can be essentially tackled by finding dense subgraphs.

There are two competing factors in dense subgraph extraction: density and size. If one wants a higher density, then the sizes of the corresponding subgraphs tend to be smaller. If one wants to extract larger subgraphs, she/he has to compromise in density requirement. In many applications one has to explore tradeoffs between these two factors. For example, it is interesting to find dense subgraphs in a friendship social network, which help to capture groups of people who may be influenced together. At the same time, according to the famous Dunbar's number theory [8], an individual can only maintain stable relationships with a limited number of people (e.g., 150). Thus, one may want to extract smaller and more compact subgraphs, which correspond to closely connected people. Here, we need to constrain the size of dense subgraphs and expect a higher density. Although many algorithms have been proposed to extract dense subgraphs [9], [10], [11], a fundamental challenge remains – how can one explore the possible tradeoffs between density and size of the extracted subgraphs conveniently?

The existing methods provide some limited capability in exploring the tradeoffs between density and size. Specifically, [12], [13], [14] find dense subgraphs maximizing quadratic function $\mathbf{x}^\top A\mathbf{x}$. In this setting, one cannot control the size of the extracted dense subgraphs. Pavan and Pelillo [15] proposed a different objective function $\mathbf{x}^\top(A-\alpha I)\mathbf{x}$ to extract dense subgraphs that are not as dense as but larger than those found in [12], [13], [14], where one can use the hyper-parameter $\alpha > 0$ to control the size of extracted subgraphs to some extent. The larger the value of α, the larger the size of the extracted subgraphs and the lower the density. However, it remains untouched how to extract dense subgraphs of density higher than those found in [12], [13], [14], that is, those maximizing $\mathbf{x}^\top A\mathbf{x}$. Moreover, to the best of our knowledge, there does not exist a unified framework or a unified method to explore various tradeoffs between density and size.

In this paper, we systematically investigate the tradeoffs between density and size in extracting dense subgraphs, and make several contributions.

- First, we introduce a new objective function $\mathbf{x}^\top(A+\alpha I)\mathbf{x}$ to address the need of finding subgraphs smaller and denser than those found in [12], [13], [14]. By increasing the hyper-parameter $\alpha > 0$, we can find smaller and denser subgraphs.

Corresponding author: Enhong Chen

IEEE/ACM ASONAM 2016, August 18-21, 2016, San Frascisco, CA, USA
978-1-5090-2846-7/16/$31.00

Objective	$\mathbf{x}^\top(A+\alpha I)\mathbf{x}$	$\mathbf{x}^\top A\mathbf{x}$	$\mathbf{x}^\top(A-\alpha I)\mathbf{x}$
Relative density	high	medium	low
Relative size	small	medium	large

TABLE I
TRADEOFFS BETWEEN DENSITY AND SIZE USING DIFFERENT OBJECTIVE FUNCTIONS.

- Second, based on the similarity among [12], [13], [14], [15] and this study, shown in Table I, we propose a natural unified framework using objective function $\mathbf{x}^\top(A+\beta I)\mathbf{x}$, where β can be negative, 0, or positive. We provide theoretical upper and lower bounds of the hyper-parameter in the unified framework and derive the meaningful range of the hyper-parameter so that solutions to the unified framework correspond to non-trivial dense subgraphs.
- Third, to develop an efficient method for the unified framework, we take the quadratic programming approach and analyze the KKT conditions of the problem. We extend the SEA method [14] to tackle the unified framework. Moreover, we show that optimizing the proposed unified framework is essentially a relaxation of the maximization of a family of density functions.
- Finally, we evaluate the performance of the proposed framework on four data sets. The experimental results confirm the effectiveness of our framework.

The rest of the paper is organized as follows. Section II reviews related work. Section 3 introduces a new objective function and proposes the unified framework. Section 4 develops a quadratic programming method. Section 5 shows that optimizing the proposed unified framework is essentially a relaxation of the maximization of a family of density functions. Section 6 reports the experimental results. Section 7 concludes the paper.

II. RELATED WORK

Given a graph $G=(V,E)$ and a subset of vertices $S\subseteq V$, let G_S be the corresponding vertex-induced subgraph. Finding a dense subgraph is to find a set of vertices S that maximizes the edge density of G_S.

A straightforward way to measure edge density is by $e(S)/\binom{|S|}{2}$, where $e(S)$ is the number of edges in G_S. However, directly optimizing the density in this way leads to trivial results, since a single edge has the maximum density. Thus, many efforts have been made to formulate alternative edge density so that the maximization can lead to dense subgraphs of a non-trivial size. Using different definitions of edge density, different types of dense subgraphs can be extracted, such as densest subgraphs [16], [17] and quasi-cliques [18], [19]. The densest subgraph problem [16], [17] is to find a set of vertices S that maximize $e(S)/|S|$, and the corresponding edge density definition is $e(S)/|S|$, too. A set of vertices S is an α-quasi-clique [18], [19] if $e(S)\geq\alpha\binom{|S|}{2}$. Essentially, α-quasi-cliques are subgraphs whose edge densities are greater than or equal to α. In this paper, we show that the unified framework is a relaxation of the maximization of a family of interesting and popularly used edge density functions.

In addition to extracting dense subgraphs through maximizing different kinds of edge density functions, there are some works that find dense subgraphs by exploring various cohesive sub-structures of graphs. Along this line, many notions of such sub-structures were proposed, such as k-plex [20], k-core [21] and k-truss [22], [23]. The degree of each vertex within a k-plex of s vertices is $s-k$. The k-core is the largest subgraph in which each vertex has a degree at least k within the subgraph. The k-truss is the largest subgraph in which every edge is involved in at least $(k-2)$ triangles within the subgraph. To some extent, these sub-structures are relaxations of cliques (i.e., a subset of vertices such that its induced subgraph is complete) [24]. Thus, finding these sub-structures is to extract dense subgraphs. Instead of directly extracting cohesive sub-structures, Tsourakakis [25] introduced k-clique densest subgraph problem, which aims to find subgraphs that maximize the density of specific sub-structures, such as triangles when $k=3$. This problem generalizes the well studied densest subgraph problem which is a special case for $k=2$. Then Mitzenmacher *et al.* [26] developed an efficient sampling method for this problem. All these works are based on some specific sub-structures of graphs, which makes them different from our work that focuses on exploring the tradeoff between size and density in extracting dense subgraphs with no requirements of their structures.

In addition to the traditional graph theory approaches, *mathematical programming methods* are popularly used in extracting dense subgraphs. Papailiopoulos *et al.* [27] developed a numerical method for densest k-subgraphs problem [28] that finds dense subgraphs via low-rank bilinear optimization. Motzkin and Straus [12] proved that finding a maximal clique in an unweighted graph is equivalent to finding the maxima of the quadratic function $\mathbf{x}^\top A\mathbf{x}$ on the simplex. Pavan and Pelillo [13] further extended the result to weighted graphs and introduced a novel graph-theoretic concept called dominant set that is a generalization of maximal cliques in the context of weighted graphs and equivalent to maximal clique in the unweighted case [29]. They showed that finding dominant sets in weighted graphs is exactly finding local maxima of the quadratic function $\mathbf{x}^\top A\mathbf{x}$ on the simplex. Liu *et al.* [14] proposed a highly efficient algorithm called SEA to solve the problem. The main drawback of this method is that it cannot control the size of the extracted dense subgraphs. Pavan and Pelillo [15] introduced another formulation $\mathbf{x}^\top(A-\alpha I)\mathbf{x}$ to extract larger and sparser clusters. But their method cannot find denser and smaller clusters. We will propose a new objective function $\mathbf{x}^\top(A+\alpha I)\mathbf{x}$ to address the issue.

Although the proposed formulation $\mathbf{x}^\top(A+\alpha I)\mathbf{x}$ in this paper is very similar to the work of Pavan and Pelillo [15] (i.e., $\mathbf{x}^\top(A-\alpha I)\mathbf{x}$) in form, the contribution of our work is substantial in the following aspects.

1) The change of sign in the formulation makes a significant difference in the objectives. While Pavan *et al.* [15] focused on finding larger subgraphs with compromised

density, we focus on finding smaller dense subgraphs with even higher density.

2) Our work is a solid theoretical supplement for the line of previous methods proposed by Pavan *et al.* [13], [15] and Liu *et al.* [14]. While those previous works only focused on various dense subgraph mining applications, we organize the family of those works and ours into a unified framework and also provide an in-depth theoretical analysis.
3) We further reveal the intriguing one-to-one relationship between our general model and the family of traditional edge density functions on unweighted graphs, such as clique and quasi-clique [16], [17], [18], [19].

In summary, this work makes novel and solid theoretical contributions to the class of mathematical programming methods [13], [15], [14] that focus on dense subgraph detection on weighted graphs.

III. A Unified Framework

Consider a graph $G = (V, E, w)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of vertices, $E \subseteq V \times V$ is a set of edges, $w : E \to \mathbb{R}^+$ is a weight function over the set of edges, and $\mathbb{R}$ is the set of real numbers. Let A be the adjacency matrix of graph G such that $A_{i,j} = w(v_i, v_j)$ if $(v_i, v_j) \in E$, and $A_{i,j} = 0$ otherwise. In this paper, we only consider simple graphs, that is $A_{i,i} = 0$.

Let $U = \{1, 2, \ldots, n\}$ be the set of indices of the vertices in V. For any subset $S \subseteq U$, the corresponding vertex induced subgraph is $G_S = (V_S, E_S)$, where $V_S = \{v_i \mid i \in S\}$ and $E_S = \{(v_i, v_j) \mid i \in S, j \in S, \text{and } (v_i, v_j) \in E\}$.

Denote by $\mathbf{x}$ an n-dimensional vector associated with the vertices of a subgraph, where the components of $\mathbf{x}$ represent the participation of vertices in the subgraph: if $\mathbf{x}_i$ has a large value, vertex v_i is strongly associated with the subgraph, and vice versa. We define the support of $\mathbf{x}$ as the set of indices corresponding to its non-zero components, that is, $\delta(\mathbf{x}) = \{i \mid \mathbf{x}_i \neq 0\}$. Consider the simplex

$$\triangle = \{\mathbf{x} \in \mathbb{R}^n \mid \forall i = 1, \ldots, n, \mathbf{x}_i \geq 0 \text{ and } \sum_i \mathbf{x}_i = 1\}.$$

To extract subgraphs that are smaller than those maximizing function $\mathbf{x}^\top A\mathbf{x}$ [13], we are interested in the following optimization problem.

$$\begin{aligned} &\text{Maximize } \mathbf{x}^\top (A + \alpha I)\mathbf{x} \\ &s.t.\ \mathbf{x} \in \triangle \end{aligned} \tag{1}$$

where $\alpha > 0$ is a hyper-parameter. We can control the size of the extracted dense subgraphs by setting the hyper-parameter α. Once we obtain a solution to the problem, we can extract the corresponding dense subgraph $G_{\delta(\mathbf{x})}$. We will refer to Eq. 1 as Problem I in this paper.

As mentioned in Section II, there exist two related problems. One of them is the following optimization problem [13], referred to as Problem II in this paper.

$$\begin{aligned} &\text{Maximize } \mathbf{x}^\top A\mathbf{x} \\ &s.t.\ \mathbf{x} \in \triangle \end{aligned} \tag{2}$$

The other one is the following [15], referred to as Problem III in this paper.

$$\begin{aligned} &\text{Maximize } \mathbf{x}^\top (A - \alpha I)\mathbf{x} \\ &s.t.\ \mathbf{x} \in \triangle \end{aligned} \tag{3}$$

where $\alpha > 0$ is a hyper-parameter.

Apparently, the three problems are very related to each other, as deliberated in Table I. Comparing to Problem II, Problems I and III have a hyper-parameter α to control the size of the extracted dense subgraphs. Moreover, the signs before the hyper-parameters in Problem I and Problem III are complementary. Naturally, we can unify the three problems into the following framework.

$$\begin{aligned} &\text{Maximize } \mathbf{x}^\top (A + \beta I)\mathbf{x} \\ &s.t.\ \mathbf{x} \in \triangle \end{aligned} \tag{4}$$

where $\beta \in \mathbb{R}$ is a hyper-parameter.

The unified framework corresponds to Problems I, II and III when $\beta < 0$, $\beta = 0$ and $\beta > 0$, respectively. In general, the extracted dense subgraphs become smaller when β increases.

IV. A Quadratic Programming Method

In this section, we develop a method for the unified framework using quadratic programming. We first discuss the range of the hyper-parameter, and then analyze the Karush-Kuhn-Tucker conditions of the unified framework. The quadratic programming method is an extension of SEA [14].

Given a subset $S \subseteq U$, the surface of $\triangle$ with respect to S is $\triangle_S = \{\mathbf{x} \in \triangle \mid \delta(\mathbf{x}) \subset S\}$, and the relative interior is $\text{int}(\triangle_S) = \{\mathbf{x} \in \triangle \mid \delta(\mathbf{x}) = S\}$. We denote by $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$, respectively, the largest and smallest eigenvalues of matrix A.

A. Bounds of the Hyper-parameter β

In the unified framework, we use hyper-parameter β to control tradeoff between the size and density of the extracted subgraphs. A too large or too small value of β may lead to trivial results, such as a single vertex or the whole graph. What should be the meaningful range of β?

We provide an upper bound of β as follows.

Theorem 1. *if $\beta > A_{\max}$, where $A_{\max}$ is the largest element of A, the solution $\mathbf{x}$ of Eq. 4 has only one non-zero component, that is, $|\delta(\mathbf{x})| = 1$ and the dense subgraph extracted has only one single node.*

Proof. Let B be the transformed adjacency matrix $A + \beta(I - \mathbf{e}\mathbf{e}^\top)$, where $\mathbf{e}$ is a column vector whose components are all 1's. Since $\mathbf{x} \in \triangle$, maximizing $\mathbf{x}^\top (A + \beta I)\mathbf{x}$ is equivalent to maximizing $\mathbf{x}^\top B\mathbf{x}$. We can see that

$$B_{i,j} = \begin{cases} A_{i,j} - \beta & \text{if } i \neq j \\ A_{i,j} & \text{if } i = j \end{cases}$$

Thus, $\mathbf{x}^\top B\mathbf{x}$ can be written as

$$\sum_{i \neq j} \mathbf{x}_i \mathbf{x}_j B_{i,j} + \sum_i \mathbf{x}_i^2 B_{i,i} \tag{5}$$

Since $A_{i,i} = 0$, the second term of Eq. 5 is always equal to 0. If $\beta > A_{\max}$, then $B_{i,j} < 0$ when $i \neq j$. Thus, if $|\delta(\mathbf{x})| > 1$, the first item must be negative, which decreases the value of the objective function. It follows that the solution $\mathbf{x}$ of Eq. 4 only has one non-zero component. □

The following statement gives a lower bound of β.

Theorem 2 ([15]). *if $\beta < -\lambda_{\max}(A)$, the objective function is a strictly concave function, and the only solution $\mathbf{x}$ of Eq. 4 belongs to int($\triangle_V$), that is, $\delta(\mathbf{x}) = U$.*

According to the above two theorems, we can get the range for β that can lead to non-trivial results. The range is $[-\lambda_{\max}(A), A_{\max}]$. Furthermore, we have the following.

Theorem 3. *Given a solution $\mathbf{x}$ of Eq. 4, for $k \in U$, if*

$$\sum_{j \in \delta(\mathbf{x})} \mathbf{x}_j B_{k,j} < 0,$$

then $k \notin \delta(\mathbf{x})$, where $B = A + \beta(I - \mathbf{e}\mathbf{e}^\top)$ is the transformed adjacency matrix.

Proof. $\mathbf{x}^\top B\mathbf{x}$ can be rewritten to

$$2\sum_{j \neq k} \mathbf{x}_k \mathbf{x}_j B_{k,j} + \sum_{i,j \neq k} \mathbf{x}_i \mathbf{x}_j B_{i,j} + \mathbf{x}_k^2 B_{k,k}$$

For a specific k, the second term is a constant and the third term is equal to 0. Thus, for $\mathbf{x}_k$, the objective function is a linear function. When $\sum_{j \in \delta(\mathbf{x})} \mathbf{x}_j B_{k,j} < 0$, $\mathbf{x}_k$ should be 0; otherwise the objective function increases when we set $\mathbf{x}_k = 0$ and increase another component $\mathbf{x}_m$ that satisfies the condition $\sum_{j \in \delta(\mathbf{x})} \mathbf{x}_j B_{m,j} \geq 0$. This contradicts the fact that $\mathbf{x}$ is a solution of Eq. 4. □

Theorem 3 explains how the hyper-parameter β works when β is positive. $B_{i,j}$ becomes negative when $A_{i,j}$ is close to 0. As a result, for a vertex v_k, if there are sufficiently many vertices that are weakly connected or not connected to v_k, that is, the corresponding $A_{k,j}$ values are close to 0, then v_k is not added to the subgraph. To fully understand the effect of hyper-parameter β in the range from negative to positive, we carry out a thorough analysis on the KKT conditions for the unified framework in the next subsection.

B. KKT Conditions for the Unified Framework

Given a specific value of β, Eq. 4 is a quadratic programming problem. Therefore, we first analyze its Karush-Kuhn-Tucker (KKT) conditions.

For a local maximal $\mathbf{x}^*$, there exist $n+1$ real constants $\mu_1, \ldots, \mu_n$ and λ, $\mu_i \geq 0$ for all $i = 1, \ldots, n$, such that

$$\begin{aligned} &(A\mathbf{x}^*)_i + \beta\mathbf{x}_i^* - \lambda + \mu_i = 0 \\ &\sum_{i=1}^{n} \mathbf{x}_i^* \mu_i = 0 \end{aligned} \tag{6}$$

Since both $\mathbf{x}_i^*$ and μ_i are non-negative, the second condition means that $i \in \delta(\mathbf{x})$ implies $\mu_i = 0$. Thus, Eq. 6 can be rewritten as

$$(A\mathbf{x}^*)_i + \beta\mathbf{x}_i^* \begin{cases} = \lambda & \text{if } i \in \delta(\mathbf{x}^*) \\ \leq \lambda & \text{otherwise} \end{cases}$$

At the same time, it is easy to derive from the first condition that $\lambda = \mathbf{x}^{*\top}(A + \beta I)\mathbf{x}^*$. Then, we can further rewrite the KKT conditions as

$$(A\mathbf{x}^*)_i \begin{cases} = \mathbf{x}^{*\top} A\mathbf{x}^* + \beta(\mathbf{x}^{*\top}\mathbf{x}^* - \mathbf{x}_i^*) & \text{if } i \in \delta(\mathbf{x}^*) \\ \leq \mathbf{x}^{*\top} A\mathbf{x}^* + \beta\mathbf{x}^{*\top}\mathbf{x}^* & \text{otherwise} \end{cases} \tag{7}$$

When $\beta = 0$, the righthand side of Eq. 7 has only one term $\mathbf{x}^{*\top} A\mathbf{x}^*$ in both cases. When $\beta \neq 0$, the righthand side has two terms, the extra term being $\beta\mathbf{x}^{*\top}\mathbf{x}^*$ in the second case of Eq. 7. This means that there are more indices not in $\delta(\mathbf{x})$ when $\beta > 0$, and consequently the size of the corresponding subgraph $G_{\delta(\mathbf{x})}$ is smaller. Similarly, when $\beta < 0$, there are more vertices included in the result subgraph. The following two theorems provide the formal statements.

Theorem 4. *For any subgraph $S \subset U$, let A_S be the corresponding adjacency matrix. There exists at least one KKT point for Eq. 4 in $\triangle_S$ if $\beta \geq \max_{\mathbf{x} \in \triangle_S} \max_{i \notin \delta(\mathbf{x})} \frac{(A\mathbf{x})_i}{\mathbf{x}^\top\mathbf{x}} - \lambda_{\min}(A_S)$.*

Proof. Denote by $\mathbf{x}_S$ the vector obtained from $\mathbf{x}$ by dropping all the components in $U \setminus S$. We define

$$\gamma(\mathbf{x}) = \max_{i \notin \delta(\mathbf{x})} \frac{(A\mathbf{x})_i - \mathbf{x}^\top A\mathbf{x}}{\mathbf{x}^\top\mathbf{x}}.$$

For $\mathbf{x} \in \triangle_S$, we have

$$\begin{aligned} \gamma(\mathbf{x}) &= \max_{i \notin \delta(\mathbf{x})} \frac{(A\mathbf{x})_i}{\mathbf{x}^\top\mathbf{x}} - \frac{\mathbf{x}_S^\top A_S \mathbf{x}_S}{x_S^\top \mathbf{x}_S} \\ &\leq \max_{\mathbf{x} \in \triangle_S} \Big\{ \max_{i \notin \delta(\mathbf{x})} \frac{(A\mathbf{x})_i}{\mathbf{x}^\top\mathbf{x}} - \frac{\mathbf{x}_S^\top A_S \mathbf{x}_S}{\mathbf{x}_S^\top \mathbf{x}_S} \Big\} \\ &\leq \max_{\mathbf{x} \in \triangle_S} \max_{i \notin \delta(\mathbf{x})} \frac{(A\mathbf{x})_i}{\mathbf{x}^\top\mathbf{x}} - \min_{\mathbf{x}_S} \frac{\mathbf{x}_S^\top A_S \mathbf{x}_S}{\mathbf{x}_S^\top \mathbf{x}_S} \\ &\leq \max_{\mathbf{x} \in \triangle_S} \max_{i \notin \delta(\mathbf{x})} \frac{(A\mathbf{x})_i}{\mathbf{x}^\top\mathbf{x}} - \lambda_{\min}(A_S) \end{aligned}$$

where the last inequality follows the Rayleigh-Ritz theorem [30]. Since $\beta \geq \max_{\mathbf{x} \in \triangle_S} \max_{i \notin \delta(\mathbf{x})} \frac{(A\mathbf{x})_i}{\mathbf{x}^\top\mathbf{x}} - \lambda_{min}(A_S)$, $\beta \geq \gamma(\mathbf{x})$ holds for all $\mathbf{x} \in \triangle_S$. It follows that $(A\mathbf{x})_i \leq \mathbf{x}^\top A\mathbf{x} + \beta\mathbf{x}^\top\mathbf{x}$ holds for all $\mathbf{x} \in \triangle_S$ and $i \notin \delta(\mathbf{x})$. As a result, if $\mathbf{x}_S^*$ is a KKT point of the subgraph G_S, then $\mathbf{x}^*$ must be a KKT point of the graph G, where $\mathbf{x}^*$ is obtained from $\mathbf{x}_S^*$ by assigning 0's to the components whose indices are not in S. It is obviously that $\mathbf{x}^*$ is in $\triangle_S$. □

Theorem 5 ([15]). *If $\beta < -\lambda_{\max}(A_S)$, then there is no point in int($\triangle_S$) that is a local maxima of Eq. 4.*

The two theorems indicate how the hyper-parameter affects the size of the extracted subgraph. For a subset $S \subset U$, if

β is less than $-\lambda_{\max}(A_S)$, we cannot find a KKT point in the corresponding subgraph G_S, and thus have to enlarge the search scope. It means that a small value of β favors large subgraphs when $\beta < 0$. When $\beta > 0$, if β is no less than $\max_{\mathbf{x}\in\triangle_S} \max_{i\notin\delta(\mathbf{x})} \frac{(A\mathbf{x})_i}{\mathbf{x}^\top\mathbf{x}} - \lambda_{\min}(A_S)$, there must be at least one KKT point in the corresponding subgraph G_S. Therefore, the size of the extracted subgraph is not greater than $|S|$.

C. A Generalized Method

Since Eq. 4 is a standard quadratic programming problem, many optimization methods can be applied to solve it. Liu *et al.* [14] proposed an efficient algorithm called SEA to solve Problem II. We extend it to the unified problem Eq. 4.

SEA is an iterative algorithm. Each iteration of the algorithm has two phases: the shrink phase and the expansion phase. In the shrink phase, it finds a KKT point $\mathbf{x}_S^*$ of the current subgraph G_S. In the expansion phase, some related vertices are added and form a new subgraph $G_{S'}$. The algorithm terminates when no more vertex can be added in the expansion phase.

In the shrink phase, SEA utilizes the replicator dynamics algorithm [31] to find the KKT point $\mathbf{x}_S^*$ of subgraph G_S. For the unified framework, we change the update formula in t-th iteration to

$$(\mathbf{x}_S)_i(t+1) = (\mathbf{x}_S)_i(t)\frac{((A_S+\beta I)\mathbf{x}_S(t))_i}{\mathbf{x}_S(t)^\top(A_S+\beta I)\mathbf{x}_S(t)}$$

where A_S is the corresponding adjacency matrix of the subgraph G_S.

In the expansion phase, the algorithm extends $\mathbf{x}_S^*$ to $\mathbf{x}^*$ by assigning 0's to the components whose indices are not in S. Let $C = \{i \mid (A\mathbf{x}^*)_i > \mathbf{x}^{*\top}A\mathbf{x}^* + \beta(\mathbf{x}^{*\top}\mathbf{x}^* - \mathbf{x}_i^*)\}$ be the indices of the components that violate the KKT conditions. If C is empty, then $\mathbf{x}^*$ is already a KKT point of G. If not, $\mathbf{x}^*$ is updated and the corresponding vertices are added to the subgraph G_S. For the unified framework, we change the update formula to

$$\begin{aligned}
\gamma_i &= \begin{cases} 0 & i \notin C \\ (A\mathbf{x}^*)_i + \beta\mathbf{x}_i^* - \mathbf{x}^{*\top}(A+\beta I)\mathbf{x}^* & i \in C \end{cases} \\
\mathbf{b}_i &= \begin{cases} -\mathbf{x}_i^*\sum_i \gamma_i & i \in \delta(\mathbf{x}^*) \\ \gamma_i & i \notin \delta(\mathbf{x}^*) \end{cases} \\
\mathbf{x}_{new} &= \mathbf{x}^* + l\mathbf{b}
\end{aligned} \tag{8}$$

where l is the step size.

The optimal step size can be decided by maximizing the following difference.

$$\begin{aligned}
f(\mathbf{x}^*+l\mathbf{b}) - f(\mathbf{x}^*) =& (\mathbf{x}^*+l\mathbf{b})^\top(A+\beta I)(\mathbf{x}^*+l\mathbf{b}) \\
&- \mathbf{x}^{*\top}(A+\beta I)\mathbf{x}^* \\
=& \mathbf{b}^\top(A+\beta I)\mathbf{b}l^2 + 2\mathbf{b}^\top(A+\beta I)\mathbf{x}^*l \\
=& \mathbf{b}^\top(A+\beta I)\mathbf{b}l^2 + 2\sum_i \gamma_i^2 l
\end{aligned}$$

Thus, the optimal step size l^* is

$$l^* = \begin{cases} \frac{1}{\sum_i \gamma_i} & \text{if} \quad \mathbf{b}^\top(A+\beta I)\mathbf{b} \geq 0 \\ \min(\frac{1}{\sum_i \gamma_i}, -\frac{\sum_i \gamma_i^2}{\mathbf{b}^\top(A+\beta I)\mathbf{b}}) & \text{otherwise} \end{cases}$$

When $\mathbf{x}^*$ is updated, the support $\delta(\mathbf{x}^*)$ is also expanded to its neighbors and forms the new subgraph $G_{\delta(\mathbf{x}_{new})}$.

To understand the effect of β, let us analyze the expansion phase of the algorithm. From Eq. 8, we can see that the newly added vertices are in the intersection of C and $\{i \mid i \notin \delta(\mathbf{x}^*)\}$. Thus, comparing to the situation when $\beta = 0$, $(A\mathbf{x}^*)_i$ is more likely greater than $\mathbf{x}^{*\top}A\mathbf{x}^* + \beta(\mathbf{x}^{*\top}\mathbf{x}^* - \mathbf{x}_i^*)$ when β is negative. As a consequence, there will be more elements in C. A similar analysis can be applied to the situation when β is positive. Thus, there are more elements in the final $\delta(\mathbf{x}^*)$ when β decreases. It follows that Problem III finds larger subgraphs than Problem II does, and Problem I finds smaller subgraphs than Problem II does. We will verify this relation in the experiments.

V. Extracting Dense Subgraphs as Relaxation of the Maximization of a Family of Density Functions

In this section, we show that the unified framework is a relaxation of the maximization of a family of density functions.

Theorem 6. *The unified framework Eq. 4 is a relaxation of the maximization of the edge density function*

$$\frac{1}{|S|^2}\left[e(S) - \beta\binom{|S|}{2}\right] \tag{9}$$

where $e(S)$ is the number of edges in G_S if G_S is unweighted, and the sum of weights of edges when G_S is weighted.

Proof. Let $k = |S|$, and $\mathbf{b}$ be a binary indicator vector where $\mathbf{b}_i = 1$ when the vertex v_i belongs to S and 0 otherwise. Then, we can rewrite Eq. 9 to

$$\frac{1}{k^2}\left(\frac{\mathbf{b}^\top A\mathbf{b}}{2} - \beta\frac{k(k-1)}{2}\right) \tag{10}$$

where A is the adjacency matrix of the graph. Let $\mathbf{x} = \frac{\mathbf{b}}{k}$. We can rewrite Eq. 10 to

$$\frac{1}{2}\left[\mathbf{x}^\top A\mathbf{x} - \beta(1-\frac{1}{k})\right] \tag{11}$$

Since $\mathbf{x}^\top\mathbf{x} = \frac{1}{k}$, we can further rewrite Eq. 11 to

$$\frac{1}{2}\left[\mathbf{x}^\top(A+\beta I)\mathbf{x} - \beta\right]$$

As a result, we can rewrite the maximization of the edge density in Eq. 9 as

$$\begin{aligned}
&\text{Maximize } \mathbf{x}^\top(A+\beta I)\mathbf{x} \\
&s.t.\ \mathbf{x} \in \triangle_D
\end{aligned}$$

where

$$\triangle_D = \{\mathbf{x} \in \triangle \mid \exists k > 0, \forall i = 1,\ldots,n, \mathbf{x}_i = \frac{1}{k} \text{ or } \mathbf{x}_i = 0\}.$$

Immediately, we can obtain the unified framework by relaxing $\mathbf{x}$ from the discrete set $\triangle_D$ to the continues space $\triangle$. □

Please note that the edge density function in Eq. 9 in fact represents a family with respect to hyper-parameter β. When $\beta = 0$, the edge density function is $e(S)/|S|^2$, which is a minor revision of the edge density $e(S)/\binom{|S|}{2}$. Correspondingly, Problem II is used to extract dense subgraphs. When $\beta \neq 0$, there is an extra term in the numerator of Eq. 9. The extra term can be viewed as a regularization. When $\beta < 0$, the density function favors subgraphs of large size by adding extra weight to each possible edge. Correspondingly, Problem III finds larger subgraphs. When $\beta > 0$, the density function penalizes large subgraph by subtracting some weight for each possible edge. Correspondingly, Problem I finds smaller subgraphs.

To make it more clear, we can rewrite Eq. 9 as

$$\frac{e(S)}{|S|^2} + \frac{\beta}{2} \cdot \frac{1}{|S|} - \frac{\beta}{2} \tag{12}$$

If we omit the last term $\beta/2$, which is a constant when β is fixed, the difference between Eq. 12 and $e(S)/|S|^2$ is the second term, which is a regularization penalizing large $|S|$ when $\beta > 0$ and favoring large $|S|$ when $\beta < 0$. β is the coefficient to control the importance of the regularization term. Furthermore, we can rewrite the edge density $e(S)/\binom{|S|}{2}$ to

$$\frac{e(S)}{\binom{|S|}{2}} = \frac{2e(S)}{|S|(|S|-1)} = 2 \times \left(\frac{e(S)}{|S|^2} + \frac{e(S)}{|S|^2(|S|-1)} \right)$$

We can omit the factor 2 here since it does not affect the maximization result. Then, we can rewrite Eq. 12 to

$$\frac{e(S)}{|S|(|S|-1)} - \frac{e(S)}{|S|(|S|-1)} \cdot \frac{1}{|S|} + \frac{\beta}{2} \cdot \frac{1}{|S|} - \frac{\beta}{2} \tag{13}$$

From Eq. 13, we can see that the unified framework is essentially a multi-objective optimization problem. The first term is exactly the edge density. The second term $-\frac{e(S)}{|S|(|S|-1)} \frac{1}{|S|}$ increases when $|S|$ increases. When $\beta = 0$, there is a tradeoff between density and size. This is the reason why maximizing $e(S)/|S|^2$ can avoid trivial results. When $\beta > 0$, the third term decreases when $|S|$ increases. It partially cancels the effect of the second term and the framework becomes more biased on edge density. When $\beta < 0$, the third term increases when $|S|$ increases. It strengthens the effect of the second term and the framework puts less weight on edge density. This clearly manifests why the unified framework can explore the tradeoffs between density and size in extracting dense subgraphs.

VI. Experimental Results

In this section, we test the unified framework using two synthetic data sets that were used in the previous studies and two real data sets. The two synthetic data sets are **synData1** [32] and **synData2** [33]. The two real world data sets are **ego-FB** [34] and **NDI** [33]. Data set **ego-FB** consists of 10 social circles (ego-nets) from Facebook. Data set **NDI** contains images crawled from the web and the edge weights are the similarities between images. Table II shows the statistics of the data sets.

Data set	Type	#Vertices	#Edges / Total weight
SynData1	unweighted	10,000	2270003
SynData2	weighted	10,000	3046376.28
ego-FB	unweighted	4,039	88234
NDI	weighted	14,604	2.00×10^8

TABLE II
Statistics about the data sets

To validate the effectiveness of the unified framework, we measure the size and density of the extracted subgraphs. We adopt the conventional density function $e(S)/\binom{|S|}{2}$ to measure the density.

We implemented the algorithm in MATLAB R2015a. All experiments were conducted on a PC computer with a 3.4GHz Intel Core i7-3770 processor and 16 GB memory, running Microsoft Windows 7.

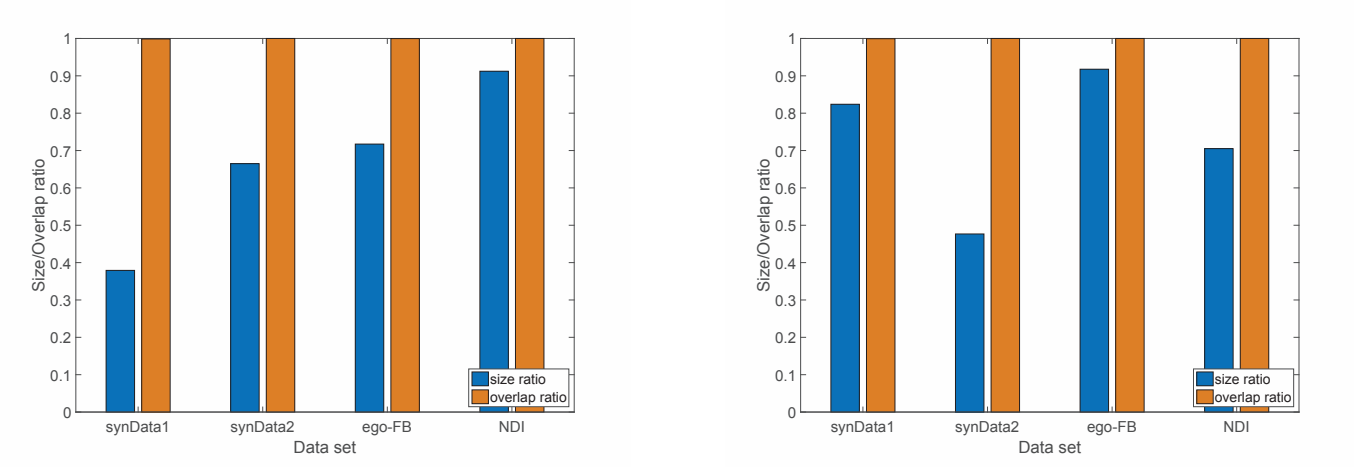
A. Subgraph Extraction from the Same Area

We first examine how the unified framework extracts dense subgraphs from the same area in a graph. We initialized the algorithm randomly and set $\beta = -5$, which returned a larger but less dense subgraph. Then, we used the vector $\mathbf{x}$ output by the algorithm, representing the subgraph extracted using $\beta = -5$, to initialize the algorithm again and set $\beta = 0$. In this way, we guide the algorithm using $\beta = 0$ to explore the subgraph extracted when $\beta = -5$. The second run of the algorithm returned a subgraph in the same area of the extracted subgraph using a larger value of β. Similarly, we used the vector $\mathbf{x}$ obtained as such to initialize the algorithm again and set $\beta = 0.5$. We compared the subgraphs extracted in these three runs. We repeated the above experiments 30 times and report in Figures 1 the mean of the size ratio and the overlap ratios of the extracted subgraphs. When we compare the dense subgraphs extracted using two different values of β, the size ratio is the number of vertices in the subgraph extracted using a larger value of β over that using a smaller value of β. The results in Figures 1 clearly show that when β increases, the size of the subgraphs extracted shrinks, since the size ratio is always smaller than 1.

To verify that the algorithm in fact searched in the same area of the graph, we calculated the overlap ratio, which is the number of common vertices in those two subgraphs against the number of vertices in the subgraph extracted using a larger value of β (that is, the smaller subgraph extracted). This ratio measures how much the subgraph extracted using a larger value of β is included in the subgraph extracted using a smaller value of β. The overlap ratio is very close to 1 in all cases, which clearly indicates that the smaller subgraphs extracted using a larger value of β in fact are extracted from the larger but less dense subgraphs using a smaller value of β.

B. Average Density and Size with Random Initialization

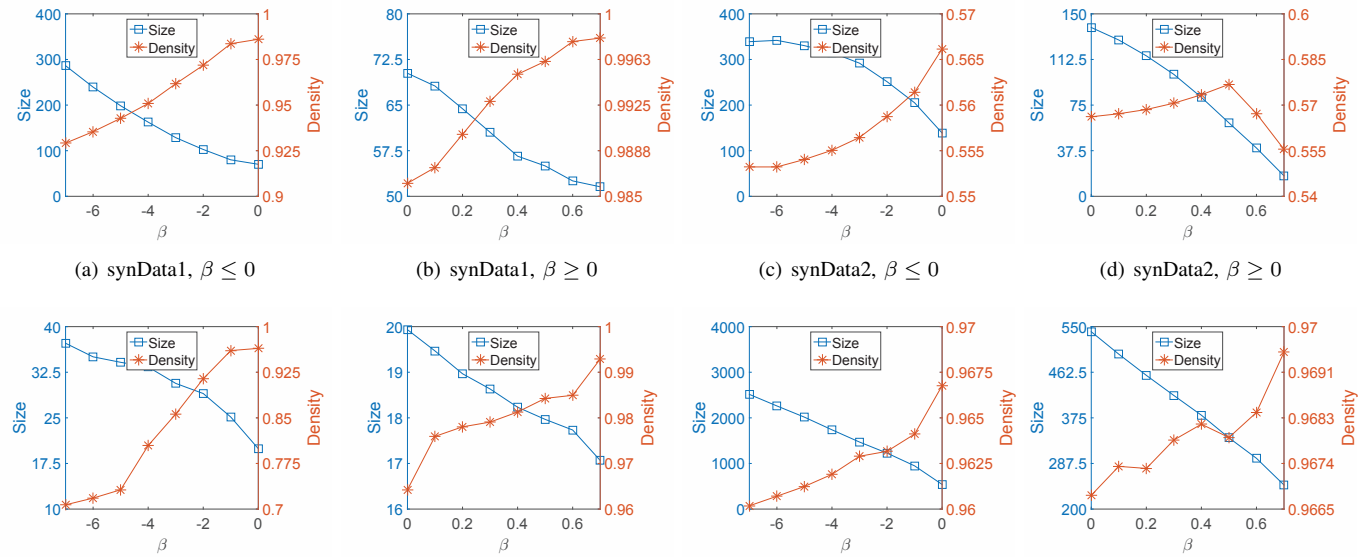
To understand statistically how the algorithm may extract dense subgraphs using different values of β in different areas of a graph, we ran the algorithm using random initialization



(a) Size ratio and overlap ratio when $\beta = 0$ and $\beta = -5$

(b) Size ratio and overlap ratio when $\beta = 0.5$ and $\beta = 0$

Fig. 1. Size ratio and overlap ratio on four data sets



(a) synData1, $\beta \leq 0$

(b) synData1, $\beta \geq 0$

(c) synData2, $\beta \leq 0$

(d) synData2, $\beta \geq 0$

(e) ego-FB, $\beta \leq 0$

(f) ego-FB, $\beta \geq 0$

(g) NDI, $\beta \leq 0$

(h) NDI, $\beta \geq 0$

Fig. 2. Average size and density of the extracted subgraphs on four data sets

30 times for each β value, and compared the average density and size. The results are shown in Figure 2. Since $\lambda_{\max}$ is much greater than $A_{\max}$, we took different step sizes when $\beta \leq 0$ and $\beta \geq 0$.

The results show that the subgraphs extracted using $\beta > 0$ are smaller than those extracted using $\beta = 0$. This confirms that Problem I indeed finds smaller subgraphs than Problem II does. When $\beta < 0$, the extracted subgraphs are bigger than those extracted using $\beta = 0$. It is consistent with the result in [15]. Combining the results using $\beta \leq 0$ and $\beta \geq 0$, we can see that the size of extracted subgraphs decreases when β increases, which is consistent with our theoretical analysis.

Next we analyze the densities of the extracted subgraphs. When $\beta \leq 0$, the density increases when β increases and the size of the extracted subgraph increases. When $\beta \geq 0$, the average density does not monotonically increase. This is because the average density was calculated from 30 runs using random initial values, which can be regarded as a random sample of 30 local maxima. We notice that, in Figures 2(b), 2(d), 2(f) and 2(h) where $\beta \geq 0$, all the average densities on the four data sets do not change much when β increases. Interestingly, when the extracted subgraphs become smaller, the density does not have to increase. For example, for a subgraph with three vertices and two edges, whose density is $\frac{2}{3}$, we can increase the density to $\frac{5}{6}$ if we add a vertex that is connected to all the three vertices.

VII. Conclusions

In this paper, we systematically investigate the tradeoffs between density and size in dense subgraph extraction. We give a new objective function that can find very dense but small subgraphs. We present a unified framework where a hyper-parameter is used to explore possible tradeoffs between density and size. We analyze the properties of the unified framework thoroughly, and develop a quadratic programming algorithm. Our experimental results on real data sets and synthetic data sets verify the effectiveness of our method. As future work we are interested in finding hierarchies of dense subgraphs.

VIII. Acknowledgement

This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the Science and Technology Program for Public Wellbeing (Grant No. 2013GS340302), the National High Technology Research and Development Program of China (Grant No. 2014AA015203), the Fundamental Research Funds for the Central Universities of China (Grant No. WK2350000001), an NSERC Discovery grant, the Canada Research Chair program, and a sponsored research grant from Huawei Technologies Co. Ltd. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[2] J. Chen and Y. Saad, "Dense subgraph extraction with application to community detection," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 7, pp. 1216–1230, 2012.

[3] L. Chu, Z. Wang, J. Pei, J. Wang, Z. Zhao, and E. Chen, "Finding gangs in war from signed networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

[4] E. Fratkin, B. T. Naughton, D. L. Brutlag, and S. Batzoglou, "Motifcut: regulatory motifs finding with maximum density subgraphs," *Bioinformatics*, vol. 22, no. 14, pp. e150–e157, 2006.

[5] M. A. Langston, L. Lin, X. Peng, N. E. Baldwin, C. T. Symons, B. Zhang, and J. R. Snoddy, "A combinatorial approach to the analysis of differential gene expression data," in *Methods of Microarray Data Analysis*. Springer, 2005, pp. 223–238.

[6] L. Chu, S. Jiang, S. Wang, Y. Zhang, and Q. Huang, "Robust spatial consistency graph model for partial duplicate image retrieval," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1982–1996, 2013.

[7] K. Crammer, P. P. Talukdar, and F. Pereira, "A rate-distortion one-class model and its applications to clustering," in *Proceedings of the 25th International Conference on Machine Learning*. ACM, 2008, pp. 184–191.

[8] R. I. Dunbar, "Neocortex size as a constraint on group size in primates," *Journal of Human Evolution*, vol. 22, no. 6, pp. 469–493, 1992.

[9] Q. Ouyang, P. D. Kaplan, S. Liu, and A. Libchaber, "Dna solution of the maximal clique problem," *Science*, vol. 278, no. 5337, pp. 446–449, 1997.

[10] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[11] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, p. 066111, 2004.

[12] T. S. Motzkin and E. G. Straus, "Maxima for graphs and a new proof of a theorem of turán," *Canad. J. Math*, vol. 17, no. 4, pp. 533–540, 1965.

[13] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 1, pp. 167–172, 2007.

[14] H. Liu, L. J. Latecki, and S. Yan, "Fast detection of dense subgraphs with iterative shrinking and expansion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 9, pp. 2131–2142, 2013.

[15] M. Pavan and M. Pelillo, "Dominant sets and hierarchical clustering," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 362–369.

[16] A. V. Goldberg, *Finding a maximum density subgraph*. University of California Berkeley, CA, 1984.

[17] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, "Greedily finding a dense subgraph," *Journal of Algorithms*, vol. 34, no. 2, pp. 203–221, 2000.

[18] J. Abello, M. G. Resende, and S. Sudarsky, "Massive quasi-clique detection," in *LATIN 2002: Theoretical Informatics*. Springer, 2002, pp. 598–612.

[19] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, "Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013, pp. 104–112.

[20] S. B. Seidman and B. L. Foster, "A graph-theoretic generalization of the clique concept*," *Journal of Mathematical sociology*, vol. 6, no. 1, pp. 139–154, 1978.

[21] S. B. Seidman, "Network structure and minimum degree," *Social networks*, vol. 5, no. 3, pp. 269–287, 1983.

[22] J. Cohen, "Trusses: Cohesive subgraphs for social network analysis," *National Security Agency Technical Report*, p. 16, 2008.

[23] J. Wang and J. Cheng, "Truss decomposition in massive networks," *Proceedings of the VLDB Endowment*, vol. 5, no. 9, pp. 812–823, 2012.

[24] R. D. Luce and A. D. Perry, "A method of matrix analysis of group structure," *Psychometrika*, vol. 14, no. 2, pp. 95–116, 1949.

[25] C. Tsourakakis, "The k-clique densest subgraph problem," in *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1122–1132.

[26] M. Mitzenmacher, J. Pachocki, R. Peng, C. Tsourakakis, and S. C. Xu, "Scalable large near-clique detection in large-scale networks via sampling," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 815–824.

[27] D. Papailiopoulos, I. Mitliagkas, A. Dimakis, and C. Caramanis, "Finding dense subgraphs via low-rank bilinear optimization," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1890–1898.

[28] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan, "Detecting high log-densities: an o (n $1/4$) approximation for densest k-subgraph," in *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM, 2010, pp. 201–210.

[29] M. Pavan, M. Pelillo, and E. Jabara, "On the combinatorics of standard quadratic optimization," *Forthcoming*, vol. 490, pp. 492–493, 2003.

[30] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[31] J. W. Weibull, *Evolutionary game theory*. MIT press, 1997.

[32] K.-Y. Chiang, J. J. Whang, and I. S. Dhillon, "Scalable clustering of signed networks using balance normalized cut," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 615–624.

[33] L. Chu, S. Wang, S. Liu, Q. Huang, and J. Pei, "Alid: scalable dominant cluster detection," *Proceedings of the VLDB Endowment*, vol. 8, no. 8, pp. 826–837, 2015.

[34] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 539–547.