March 31, 2016

RT0972
Computer Science; Network 22 pages

Research Report

Axioms of Density: How to Define and Detect the Densest Subgraph

Hiroki Yanagisawa and Satoshi Hara

IBM Research - Tokyo
IBM Japan, Ltd.
19-21, Hakozaki-cho, Nihombashi, Chuoh-ku
Tokyo 103-8501, Japan

Axioms of Density: How to Define and Detect the Densest Subgraph

Hiroki Yanagisawa and Satoshi Hara
IBM Research - Tokyo
{yanagis,satohara}@jp.ibm.com

Abstract

Detecting the densest subgraph is one of the most important problems in graph mining and has a variety of applications. Although there are many possible metrics for subgraph density, there is no consensus on which density metric we should use. In this paper, we provide formal guidelines to choose an appropriate density metric using four axioms. These axioms capture the necessary conditions any density metric should satisfy to conform with our intuition. We investigate the existing density metrics to see whether or not they satisfy these four axioms and determine which ones violate at least one of the four. In addition, we suggest a new density metric, the discounted average degree, which is an extension of the average degree metric and which satisfies all four axioms. We also show how to obtain an optimum densest subgraph for small graphs using typical density metrics, including our new density metric, by using mixed integer quadratic programming. Finally, we develop a new heuristic algorithm to quickly obtain a good approximate solution for large graphs. Our computational experiments on real-world graphs showed that our new heuristic algorithm performs best in terms of the quality of the solutions.

Keywords. densest subgraph, graph mining, integer programming, and local search.

1 Introduction

Detecting the densest subgraph is one of the most important problems in graph mining and has a variety of applications in such fields as social network analysis and biology. For example, in social network analysis, a network may represent interactions between people and we may be seeking a well-connected community by looking for a dense subgraph in the interaction network. In another example, a graph may be used to represent interactions between molecules (such as proteins and DNA in biology) and we can find biologically meaningful sets of entities by detecting dense subgraphs from the full graph [11]. More examples appear in the survey articles (such as [12]).

Given an unweighted graph $G = (V, E)$ and a density metric $f(S)$ defined on $S \subseteq V$, we define the densest subgraph problem as to find a set of vertices $S \subseteq V$ that maximizes the value of $f(S)$. The density metric $f(S)$ is defined such that it takes a higher value when the induced subgraph $G[S] = (S, E[S])$ is denser, where $E[S]$ is the set of edges $(u, v) \in E$ such that both u and v are contained in S. In this paper, $e[S]$ denotes the number of edges in $E[S]$. For simplicity, we consider only unweighted graphs, but it is easy to extend our approach to edge-weighted graphs.

When we solve the densest subgraph problem, there are many definitions that could be used for the density metric $f(S)$, but there is no consensus on which metric is best. Since a clique is a typical dense graph, where a clique is defined as a set of vertices in which each pair of vertices has an edge that connects them, one may consider that the density metric should be designed to measure the degree of similarity to a clique. One of the natural definitions for such a metric is the *clique ratio* $f(S) = e[S]/\binom{|S|}{2} = 2e[S]/|S|(|S|-1)$. However, we often obtain a spurious result if we solve the densest subgraph problem using this metric, because it cannot distinguish among cliques of different sizes. For example, with this metric, a clique with seven vertices and a clique with three vertices (that is, a triangle) would have the same density (of 1). Since even a single edge is a clique of size two, we can trivially obtain the maximum value for this density metric by picking any single edge. Therefore, the clique ratio is unsuitable as a density metric for the densest subgraph problem. In addition, the clique ratio metric has a tradeoff when choosing between small and

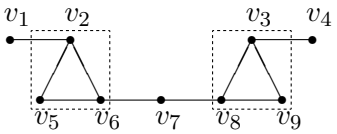

Figure 1: Optimum solution of the densest subgraph problem in which the average degree metric of the graph is $\{v_2, v_3, v_5, v_6, v_8, v_9\}$.

large graphs. For example, if we compare a clique with 12 vertices and 66 edges and a near-clique with 15 vertices and 102 edges (that is, a clique of size 15 with only three edges missing), the clique ratio favors the clique of size 12 rather than the near-clique of size 15. Although some people would perhaps prefer the smaller clique, it would generally be better to choose the larger near-clique as the densest subgraph considering the typical motivations behind the densest subgraph problem. In many applications, our goal is to extract a large well-connected set of entities, which means it is unreasonable to choose the smaller clique simply because the larger near-clique is missing a few edges. Even if there exists a situation when the smaller clique should be preferred over the larger near-clique, the density metric should have a parameter that can control their relative preferences.

Another natural definition for the density metric is the *average degree* [9, 10] $f(S) = e[S]/|S|$. In contrast to the clique ratio, this metric can distinguish among the sizes of the cliques and gives higher scores for larger cliques. However, it too involves a tradeoff in choosing between small and large graphs different from the one affecting the clique ratio metric. Here, suppose that we have a graph with 100 vertices and 500 edges and another clique with 10 vertices and 45 edges. The average degree metric of the size 100 graph is 5, and the average degree metric of the size 10 clique is 4.5, so the size 100 graph is regarded as denser than the size 10 clique when using this metric. However, most people would think that the larger graph is loosely connected and the smaller one is denser than the larger one. This problem has already been described in the literature [19]. In addition, the average degree metric has another problem in the connectivity of the output graphs. For example, when we are given a graph with nine vertices and ten edges as illustrated in Fig. 1, we may obtain a disconnected subgraph $S^* = \{v_2, v_3, v_5, v_6, v_8, v_9\}$ as the densest subgraph for the average degree metric, since there is no other subgraph whose density is strictly greater than S^*. This result is not in line with our intuition that any densest subgraph should be connected.

Because these two natural density metrics have limitations, developers use various metrics for various applications. For example, the authors of [19] suggest using a quasi-clique metric $f(S) = e[S] - \alpha\binom{|S|}{2}$ with the parameter α, Tsourakakis [18] uses a triangle density metric $f(S) = t[S]/|S|$, where $t[S]$ represents the number of triangles in $G[S]$, and the authors of [1] use yet another density metric. There are no established guidelines for choosing an appropriate density metric, and there is no consensus on which metric should be used.

Our first contribution is a set of guidelines to choose an appropriate density metric for the densest subgraph problem. We have defined four axioms that summarize the necessary conditions any density metric should satisfy so as to conform to our intuition.

- **Concentration Axiom**: When we compare two subgraphs that have the same number of vertices, we prefer the one that contains the larger number of edges.
- **Size Axiom**: When we compare two subgraphs that have the same number of edges, we prefer the smaller one.
- **Clique Axiom**: We prefer a larger clique to a smaller clique.
- **Connectivity Axiom**: We prefer a connected subgraph to a disconnected subgraph.

In Sec. 2, we formalize these statements as formal and decidable ones. Note that some of the axioms similar to our definitions were discussed in [3, 19], but ours is the first attempt to discuss these four axioms altogether to evaluate various density metrics. In Sec. 3, we evaluate each of these metrics as to whether or not it satisfies the four axioms, and show that some such as the clique ratio and average degree violate at least one axiom. Our axiomatic approach for evaluating the various metrics is similar in spirit to the axiomatic approaches for measuring entropy [5] or for measuring the importance of the individual vertices in graphs [2, 4].

Our second contribution is to propose a new density metric, the *discounted average degree* $f(S) = e[S]/|S|^\beta$ where β is a parameter such that $1 < \beta \leq 2$. This density metric satisfies all four of the axioms and is a natural extension of the average degree metric; they coincide when we set $\beta = 1$. The parameter β is used to specify the preference between a small clique and a larger near-clique. That is, the output subgraph tends to be a small clique if we use a large β and tends to be a large near-clique otherwise.

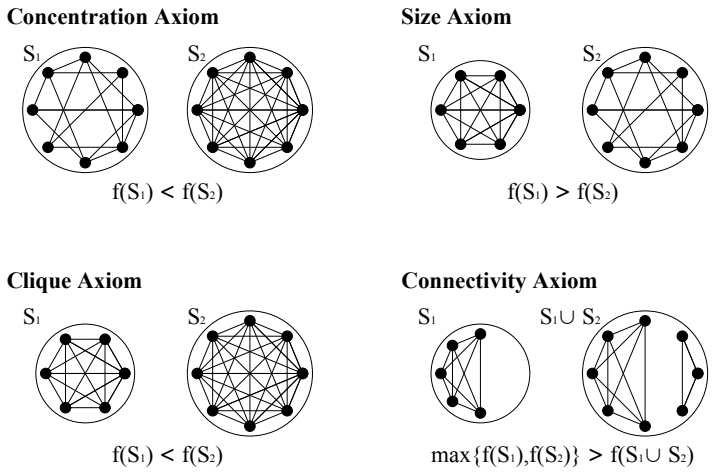

Figure 2: Illustrations of the four axioms of density

Our third contribution is an exact algorithm to obtain the optimum solution for the densest subgraph when using a density metric of the form $f(S) = e[S]/h(|S|)$ or $f(S) = e[S] - h(|S|)$, where $h(x)$ is a convex increasing function with respect to x. This form includes most of the density metrics such as the quasi-clique metric $f(S) = e[S] - \alpha\binom{|S|}{2}$ [19] and our discounted average degree metric $f(S) = e[S]/|S|^{\beta}$. Our exact algorithm uses mixed integer quadratic programming and so it takes exponential time in the worst case, but recent developments in state-of-the-art mixed integer programming solvers allow us to compute the optimum solutions for small real-world graphs (graphs with up to 1000 vertices) in a reasonable time. Regarding the computational complexity of the densest subgraph problem, the authors of [19] conjecture that it is NP-hard to obtain the optimum solution when using the quasi-clique metric; we conjecture that it is also NP-hard when using our new metric. In Sec. 8, we give evidence that supports this conjecture by showing the NP-hardness of optimizing the densest subgraph problem using our new density metric with $\beta = 2$. While the exact algorithm is too slow for large graphs, one of the advantages of having an exact algorithm is that it allows us to evaluate various metrics using small graphs without concerns about the approximate performance of a heuristic algorithm. In Sec. 6, we compare the optimum solutions of the quasi-clique metric and our discounted average degree metric, and show that ours has a broader range of control over the output subgraphs.

Our fourth contribution is a new heuristic algorithm that quickly finds an approximate solution for large graphs. This algorithm can be viewed as a local search algorithm, which consists of expanding and shrinking phases, and the combination of which is a key ingredient that enables the algorithm to search for a better solution. Our experimental results show that this algorithm outputs a higher quality solution than the current best algorithms do [6, 19]. Moreover, the solutions it obtained are almost optimum for all of the small real-world graphs we tested. Therefore, we believe that our new heuristic algorithm can be a complement to our slow but exact algorithm in practice.

Finally, we briefly discuss how to choose the parameters of the density metrics in Sec. 7.

2 Axioms of Density

While there are many density metrics we can use for the densest subgraph problem, there is no consensus on which one is the best. Here, we set out some basic properties that any density metric should exhibit as formal and provable axioms. Specifically, we present a concentration axiom, a size axiom, a clique axiom, and a connectivity axiom (Figure 2 illustrates them).

The first axiom is to favor a subgraph containing more edges if the other parameters are the same.

Axiom 2.1 *(Concentration Axiom). Let S_1 and S_2 be two subgraphs such that $G[S_1]$ has n vertices and m_1 edges, and $G[S_2]$ has n vertices and m_2 edges. Then the density metric f should satisfy $f(S_1) < f(S_2)$ whenever $n \geq 2$ and $m_2 > m_1 > 0$.*

The second axiom is to favor a subgraph with fewer vertices if the other parameters are the same.

Axiom 2.2 *(Size Axiom). Let S_1 and S_2 be two subgraphs such that $G[S_1]$ has n_1 vertices and m edges, and $G[S_2]$ has n_2 vertices and m edges. Then the density metric f should satisfy $f(S_1) > f(S_2)$ whenever $n_2 > n_1 \geq 2$ and $m > 0$.*

The third axiom is to favor a larger clique when we compare multiple cliques.

Axiom 2.3 *(Clique Axiom). Let S_1 and S_2 be two subgraphs such that $G[S_1]$ is a clique of size n_1, and $G[S_2]$ is a clique of size n_2. Then the density metric f should satisfy $f(S_1) < f(S_2)$ whenever $n_2 > n_1 \geq 2$.*

The fourth axiom is to favor a connected subgraph rather than a disconnected one.

Axiom 2.4 *(Connectivity Axiom). Let S_1 and S_2 be two subgraphs such that $G[S_1]$ has n_1 vertices and m_1 edges, $G[S_2]$ has n_2 vertices and m_2 edges, and $G[S_1]$ and $G[S_2]$ are disconnected. Then the density metric f should satisfy $\max\{f(S_1), f(S_2)\} > f(S_1 \cup S_2)$ whenever $n_1 \geq 2$, $n_2 \geq 2$, $m_1 > 0$, and $m_2 > 0$.*

As can be easily seen, these four axioms are related to three basic graph characteristics: the number of vertices, the number of edges, and the connectivity. The concentration, size, and connectivity axioms each focus on one of these characteristics, whereas the clique axiom is different from the other three in that it simultaneously considers the number of vertices and the number of edges. In general, we cannot hope for unanimous agreement when choosing between two subgraphs if we focus on two characteristics simultaneously (for example, choosing between a small clique versus a large near-clique). However, we believe that the clique axiom should be widely accepted because it avoids the problem with the clique ratio metric for the densest subgraph problem.

Note that our intention in recommending these axioms is to avoid obtaining counterintuitive results rather than to prohibit using a density metric that violates one or more axioms. Our belief is that we should at least be aware which of the axioms are or are not satisfied when we choose a density metric.

3 Density Metrics

This section described our new density metric

- **Discounted average degree** $f(S) = e[S]/|S|^{\beta}$

and the previously studied density metrics,

- **Average degree** $f(S) = e[S]/|S|$
- **Clique ratio** $f(S) = e[S]/\binom{|S|}{2}$
- **Quasi-clique** $f(S) = e[S] - \alpha\binom{|S|}{2}$
- **Triangle density** $f(S) = t[S]/|S|$.

We can also consider another natural definition of the density metric that uses the notion of a *dense graph*. In fact, there are many notions about dense graphs, most of which are distinguished by the sorts of parameters they use. For example, k-core [16] denotes a class of graphs in which the degree of every vertex is at least k, where k is a parameter. By fixing k, we can categorize a given graph as dense if it is k-core, or sparse otherwise. As another example, a γ-clique [1] is defined as a graph of size n that contains at least $\gamma\binom{n}{2}$ edges, where $0 < \gamma \leq 1$ is a parameter. Other notions include k-club [14], k-plex [17], k-clique [13], and DN-Graphs [21]. Here, let $\chi(S)$ be the indicator function for the induced subgraph $G[S]$ for some notion of a dense graph. That is, $\chi(S) = 1$ if $G[S]$ is dense in the sense of a dense graph, and $\chi(S) = 0$ otherwise. Accordingly, it is natural to define a density metric either of the form $f(S) = \chi(S)/h(|S|)$ or $f(S) = \chi(S) - h(|S|)$, where $h(x)$ is a function defined over the integers $\{2, 3, \ldots, |V|\}$. For example, the authors of [1] considered the problem of finding a γ-clique with maximum cardinality as a subgraph. This problem is equivalent to solving the densest subgraph problem using the density metric $f(S) = \chi(S)/h(|S|)$, where $\chi(S)$ is the indicator function for a γ-clique and $h(x) = 1/x$. We refer to these kinds of metrics that use an indicator function $\chi(S)$ for some notion of a dense graph as *threshold-based density metrics*.

We determined whether or not these density metrics satisfy the four axioms. Table 1 summarizes the results, where the checked results (from the fourth to the seventh columns) are for the parameter settings in the third column.

Table 1: Check List of Density Metrics in Relation to the Four Axioms

Density metric	$f(S)$	Parameter Setting	Concentration Axiom	Size Axiom	Clique Axiom	Connectivity Axiom	Overall
Average degree	$e[S]/\|S\|$		✓	✓	✓	✗	✗
Discounted average degree	$e[S]/\|S\|^\beta$	$1 < \beta \leq 2$	✓	✓	✓	✓	✓
Clique ratio	$e[S]/\binom{\|S\|}{2}$		✓	✓	✗	✓	✗
Quasi-clique	$e[S] - \alpha\binom{\|S\|}{2}$	$1/3 \leq \alpha < 1$	✓	✓	✓	✓	✓
Triangle density	$t[S]/\|S\|$		✗	✗	✓	✗	✗
Threshold-based metrics	$\chi(S)/h(\|S\|)$		✗	✗(at least one of these two)		-	✗
	$\chi(S) - h(\|S\|)$		✗	✗(at least one of these two)		-	✗

3.1 Discounted Average Degree

First, we show that the discounted average degree metric $f(S) = e[S]/|S|^\beta$ satisfies all four axioms as long as $1 < \beta \leq 2$. Since the average degree metric can be viewed as a variant of the discounted average degree metric with parameter $\beta = 1$, in this section, we will consider these two metrics together.

With respect to the concentration axiom, we have

$$f(S_1) = e[S_1]/|S_1|^\beta < e[S_2]/|S_2|^\beta = f(S_2),$$

where the inequalities hold due to assumptions $e[S_2] = m_2 > m_1 = e[S_1] > 0$ and $|S_1| = |S_2| = n$, and $\beta > 0$. This proves that both the average degree metric and discounted average degree metric satisfy the concentration axiom.

With respect to the size axiom, we have

$$f(S_1) = e[S_1]/|S_1|^\beta > e[S_2]/|S_2|^\beta = f(S_2),$$

where the inequalities hold due to assumptions $e[S_1] = e[S_2] = m$, $|S_1| = n_1 < n_2 = |S_2|$, and $\beta > 0$. This proves that both the average degree metric and discounted average degree metric satisfy the size axiom.

With respect to the clique axiom, we have

$$\begin{aligned} f(S_1) &= e[S_1]/|S_1|^\beta \\ &= n_1(n_1-1)/2n_1^\beta \\ &= n_1^{2-\beta}/2 - 1/2n_1^{\beta-1} \\ &< n_2^{2-\beta}/2 - 1/2n_2^{\beta-1} \\ &= f(S_2), \end{aligned}$$

where the inequalities hold due to assumptions $0 < n_1 < n_2$ and $1 \leq \beta \leq 2$. This proves that both the average degree metric and discounted average degree metric satisfy the clique axiom. Note that if β does not satisfy $1 \leq \beta \leq 2$, the clique axiom is violated.

With respect to the connectivity axiom, we have

$$\begin{aligned} \max\{f(S_1), f(S_2)\} &= \max\left\{\frac{e[S_1]}{|S_1|^\beta}, \frac{e[S_2]}{|S_2|^\beta}\right\} \\ &\geq \frac{|S_1|^\beta}{|S_1|^\beta + |S_2|^\beta}\frac{e[S_1]}{|S_1|^\beta} + \frac{|S_2|^\beta}{|S_1|^\beta + |S_2|^\beta}\frac{e[S_2]}{|S_2|^\beta} \\ &= \frac{e[S_1 \cup S_2]}{|S_1|^\beta + |S_2|^\beta} \\ &> \frac{e[S_1 \cup S_2]}{|S_1 \cup S_2|^\beta} \\ &= f(S_1 \cup S_2) \end{aligned}$$

where the first inequality uses the fact that the maximum over two values is at least as large as the weighted average of the two values and the last inequality holds when $\beta > 1$. This proves that the discounted average degree metric satisfies the connectivity axiom. However, the average degree metric does not satisfy this axiom, because a disconnected subgraph $S_1^* \cup S_2^*$ satisfies $f(S_1^* \cup S_2^*) = f(S_1^*) = f(S_2^*)$ for the graph shown in Fig. 1, where $S_1^* = \{v_2, v_5, v_6\}$ and $S_2^* = \{v_3, v_8, v_9\}$.

3.2 Clique Ratio

Using an analysis similar to that of the discounted average degree metric, we can show that the clique ratio metric $f(S) = e[S]/\binom{|S|}{2}$ satisfies the concentration, size, and connectivity axioms. For the clique axiom, we have $f(S_1) = 1 = f(S_2)$ for the two cliques S_1 and S_2, which means that it violates the clique axiom.

3.3 Quasi-Clique

Using an analysis similar to that of the discounted average degree metric, we can show that the quasi-clique metric $f(S) = e[S] - \alpha\binom{|S|}{2}$ satisfies the concentration and size axioms whenever $\alpha > 0$. For the clique axiom, since $f(S_1) = e[S_1] - \alpha\binom{|S_1|}{2} = (1-\alpha)\binom{n_1}{2}$, $f(S_2) = (1-\alpha)\binom{n_2}{2}$, and $n_1 < n_2$, we have $f(S_1) < f(S_2)$ as long as $\alpha < 1$. For the connectivity axiom, the authors of [19] already proved that this axiom is satisfied only when $\alpha \geq 1/3$. Hence, they concluded that the parameter α of the quasi-clique metric should be set to at least $1/3$.

Remark. Note that even a slight modification of the density metric may change the satisfiability of the four axioms. For example, a variant of the quasi-clique metric, $f(S) = e[S] - \alpha|S|^2$, does not satisfy all of the four axioms. It is easy to see that α of this metric has to satisfy $\alpha < 1/2$, since otherwise, the trivial solution (a clique of size 2) maximizes this density metric. However, although we will omit the proof, the metric does not satisfy the size and clique axioms when $\alpha < 1/2$. Even if we change the density metric to $f(S) = e[S] - \alpha|S|^\beta$ by using an additional parameter β, we can show that there are no valid parameter settings that satisfy all of the axioms. For example, when $\alpha = \beta = 1$, the metric $f(S) = e[S] - |S|$ does not satisfy the connectivity axiom.

3.4 Triangle Density

It is easy to verify that the triangle density metric satisfies the clique axiom. However, it is trivial to construct counterexamples for the concentration, size, and connectivity axioms.

3.5 Threshold-Based Density

Here, we consider a threshold-based density metric of the form $f(S) = \chi(S)/h(|S|)$. (We will omit the proof for the density metric of the form $f(S) = \chi(S) - h(|S|)$ since the argument is quite similar.) First, we show that this density metric cannot satisfy the concentration axiom. When two subgraphs S_1 and S_2 satisfy $\chi(S_1) = \chi(S_2) = 1$, we have $f(S_1) = 1/h(|S_1|) = 1/h(|S_2|) = f(S_2)$ by the assumption $|S_1| = |S_2|$. This means that this metric cannot satisfy the condition $f(S_1) < f(S_2)$ of the concentration axiom. Next, we show that it must violate at least one of the size or clique axioms. When both S_1 and S_2 induce cliques, we have $\chi(S_1) = \chi(S_2) = 1$, $f(S_1) = 1/h(|S_1|)$, and $f(S_2) = 1/h(|S_2|)$. If the penalty function $h(x)$ favors large graphs (for example $h(x) = 1/x$), we have $f(S_1) \leq f(S_2)$, which does not satisfy the condition $f(S_1) > f(S_2)$ of the size axiom. Otherwise (for example $h(x) = x$), we have $f(S_1) \geq f(S_2)$, which does not satisfy the condition $f(S_1) < f(S_2)$ of the clique axiom. Finally, we note that the indicator function $\chi(S)$ (and $h(|S|)$) determines whether this density metric satisfies the connectivity axiom, but we will not go into detail on this.

3.6 Remarks on the Connectivity Axiom

The connectivity axiom requires any density metric to satisfy the strict inequality $\max\{f(S_1), f(S_2)\} > f(S_1 \cup S_2)$, but it might be acceptable to weaken the condition so that $\max\{f(S_1), f(S_2)\} \geq f(S_1 \cup S_2)$. This is because, if a density metric satisfies this weakened connectivity axiom, an optimum subgraph $G[S^*]$ might be disconnected but any connected component of $G[S^*]$ is guaranteed to have the same density as $f(S^*)$. This means that we can obtain another optimum connected subgraph simply by using a postprocessing step that seeks any connected component from the (disconnected) optimum subgraph. The density metrics that satisfy this weakened connectivity axiom include the average degree metric and the triangle density metric. Note that we still must set $\alpha \geq 1/3$ for the quasi-clique metric because otherwise this density metric does not satisfy even the weakened connectivity axiom.

4 Exact Algorithms

Here, we show how to compute the optimum solution using a density metric of the form $f(S) = e[S]/h(|S|)$ or $f(S) = e[S] - h(|S|)$, where the function $h(x) > 0$ is a convex increasing function defined over the integers $\{2, 3, \ldots, |V|\}$,

to impose a penalty depending on the cardinality of S. (Recall that a function $h(x)$ defined over the integers is said to be *convex* if $h(x-1)+h(x+1) \geq 2h(x)$ holds for any x.) This approach can be used for many of the density metrics, including the discounted average degree and the quasi-clique, which satisfy all four of the axioms in the previous section.

4.1 Mixed Integer Quadratic Programming

First, we show how to obtain the optimum solution S that maximizes

$$f(S) = e[S] - \lambda h(|S|), \tag{1}$$

where λ is a fixed parameter. Our approach is to formulate the problem of maximizing the objective function (1) using Mixed Integer Quadratic Programming (MIQP).

To represent the first term $e[S]$ of the objective function (1) for MIQP, we use the Laplacian matrix L of graph $G = (V, E)$, which is formally defined as

$$L_{ij} = \begin{cases} -1 & \text{if } i \neq j \text{ and } (i,j) \in E \\ 0 & \text{if } i \neq j \text{ and } (i,j) \notin E \\ d_i & \text{if } i = j, \end{cases}$$

where d_i is the degree of vertex $i \in V$. Using this matrix, we can express $e[S]$ as

$$-\frac{1}{2}\sum_{i\in V}\sum_{j\in V} L_{ij}x_i x_j + \frac{1}{2}\sum_{i\in V} d_i x_i,$$

where x_i is a binary variable defined for each $i \in V$ such that $x_i = 1$ means $i \in S$ and $x_i = 0$ means $i \notin S$.

Next, we consider how to represent the second non-linear term $\lambda h(|S|)$ of the objective function (1). Since the function $h(x)$ is convex, we can represent the epigraph $y \geq h(x)$ using a set of linear inequalities

$$y \geq a_k x + b_k$$

with appropriate (a_k, b_k) pairs for $k = 3, \ldots, |V|$. Specifically, we set a_k and b_k so that the line $y = a_k x + b_k$ goes through two points $(x, y) = (k-1, h(k-1))$ and $(k, h(k))$. (For example, when $h(x) = x^\beta$, we set $a_k = k^\beta - (k-1)^\beta$ and $b_k = k(k-1)^\beta - k^\beta(k-1)$.)

In summary, we can formulate the problem of maximizing the objective function (1) as

$$\begin{aligned} \max \quad & -\frac{1}{2}\sum_{i\in V}\sum_{j\in V} L_{ij}x_i x_j + \frac{1}{2}\sum_{i\in V} d_i x_i - \lambda y \\ \text{subject to} \quad & y \geq \sum_{i\in V} a_k x_i + b_k && \forall k \\ & \sum_{i\in V} x_i \geq 2 \\ & x_i \in \{0,1\} && \forall i \in V \\ & y \geq 0. \end{aligned}$$

Here, each variable x_i takes the value 1 if vertex $i \in V$ is contained in set S, and the value 0 otherwise. The continuous variable y represents an upper bound of the second term of the objective function (1). If we relax the integral constraint $x_i \in \{0, 1\}$ into a fractional constraint $0 \leq x_i \leq 1$, the relaxed problem becomes a convex programming problem since L is positive semidefinite. Hence, this problem is often categorized as a *convex quadratic integer programming* problem, and there are many programs that can be used to solve it.

4.2 Fractional Programming

Next, we show how to maximize the density metric of the form $f(S) = e[S]/h(|S|)$, by using a technique from [9]. The pseudocode is shown as Algorithm 1. Initially, we find an arbitrary feasible solution S, and compute $\lambda = f(S) = e[S]/h(|S|)$. Then we try to find a better feasible solution S' such that $f(S') = e[S']/h(|S'|) > \lambda$. Instead of directly finding such S', we obtain $S' = \text{argmax}_S(e[S] - \lambda h(|S|))$ by using the MIQP formulation. If $e[S'] - \lambda h(|S'|) > 0$ (*)

holds, we have $f(S') = e[S']/h(|S'|) > \lambda = f(S)$. This means we have found a better solution S' relative to S, and so, we replace S with S' and update $\lambda = e[S]/h(|S|)$ with the new S. We repeat this procedure of finding another feasible solution S' such that $f(S') = e[S']/h(|S'|) > \lambda$ for the new λ. Otherwise (if (*) does not hold), we have $\max_{S'}(e[S'] - \lambda h(|S'|)) = 0$ (recall that $e[S] - \lambda h(|S|) = 0$ by the definition of λ) and this means that there is no S' such that $f(S') = e[S']/h(|S'|) > \lambda = f(S)$. Therefore the algorithm outputs S' and terminates.

Algorithm 1 Fractional Programming

Input: density metric of the form $f(S) = e[S]/h(|S|)$
Output: optimum solution S^* that maximizes $f(S)$

1: find a feasible solution S'
2: **repeat**
3: $S = S'$
4: $\lambda = e[S]/h(|S|)$
5: $S' = \mathrm{argmax}_S(e[S] - \lambda h(|S|))$.
6: **until** $e[S']/h(|S'|) = \lambda$
7: **return** S'

A key observation about this algorithm is that the value of λ increases in each iteration. Another is that, after we first obtain a feasible solution on line 5 of Algorithm 1, the size of S decreases in each iteration, because the penalty term $\lambda h(|S|)$ increases as λ increases when we solve $S' = \mathrm{argmax}_S(e[S] - \lambda h(|S|))$ while the first term $e[S]$ does not change. (We will omit a formal of these facts.)

The number of iterations of this algorithm is at most $(|E| + 1)(|V| - 1)$, because λ takes one of the possible $(|E| + 1)(|V| - 1)$ values for $f(S)$ and increases in each iteration. (Recall that the nominator and denominator of $f(S) = e[S]/h(|S|)$ take $|E|+1$ values (from $0, 1, \ldots, |E|$) and $|V|-1$ values (from $h(2), h(3), \ldots, h(|V|)$), respectively.) Since the number of iterations has a large impact on the total computation time in practice, we should discuss some ideas to reduce it. The first idea is to find a good initial solution S quickly so that the initial λ is close to the optimum objective value. In our implementation, we used the greedy algorithm [6] (the details are in Sec. 6) to obtain a good approximate initial solution. The second idea is to avoid an unneeded final iteration by changing the termination condition of the algorithm. While we find a better feasible solution in every iteration except for the last, the last iteration is executed only to confirm that there exists no other solution that is better than the current best one. Therefore we can skip the last iteration when we know that there is no other better solution from the fact that the size of S decreases in each iteration. For example, if the current best solution S is a clique, we can stop the iteration immediately when we use the discounted average degree metric, since we know that there is no S' such that $f(S') > f(S)$ as long as $|S'| \leq |S|$ and S is a clique.

5 Heuristic Algorithm

Since the exact algorithm presented in the previous section takes too long to obtain the optimum solution for large graphs, we developed a faster heuristic algorithm, which we refer to as the *accordion search* algorithm. The pseudocode is shown in Algorithm 2, and Fig. 3 illustrates its execution. As input, the algorithm receives an initial solution S, which can be any feasible solution, but preferably it should be small, typically consisting of a single vertex. (The black vertex in the top left graph in Fig. 3 corresponds to an initial solution.) Then it improves this initial solution in two phases. In the first phase, it expands S by repeatedly adding a vertex to the incumbent solution S. In each iteration of this phase, the vertex v to be added is chosen from $V \setminus S$ such that $f(S \cup \{v\})$ is maximized. This can be done by finding a vertex $v \in V \setminus S$ such that the number of edges between v and S is maximized for most of the density metrics. (The numbers in Fig. 3 show the order of the vertices added to S.) The algorithm continues this expand phase until S satisfies a certain condition (discussed later); then it moves on to the second phase. In the second phase, it repeatedly removes a vertex u from S until S becomes empty. In each iteration of this second phase, the vertex u is chosen from S so that $f(S \setminus \{u\})$ is maximized. This can be done by removing the vertex with the minimum degree in $G[S]$ for most of the density metrics. (Again, the numbers in Fig. 3 show the order of the vertices removed from S.) The algorithm outputs the best subgraph that appeared during the execution.

Intuitively, an ideal scenario for this algorithm is that it receives an initial solution S such that $S \subset S^*$ where S^* is an optimum solution, expands S into one large enough to hold $S^* \subset S$, and finally outputs a near-optimum solution

Algorithm 2 Accordion Search

Input: a density metric $f(S)$ and an initial solution S
Output: a vertex set $S' \subseteq V$

$S' \leftarrow S$
repeat // the first phase
  find vertex $v \in V \setminus S$ that maximizes $f(S \cup \{v\})$
  add v to S
  if $f(S) > f(S')$ **then** $S' \leftarrow S$
until S becomes large enough
repeat // the second phase
  find vertex $u \in S$ that maximizes $f(S \setminus \{u\})$
  remove u from S
  if $f(S) > f(S')$ **then** $S' \leftarrow S$
until S becomes empty
return S'

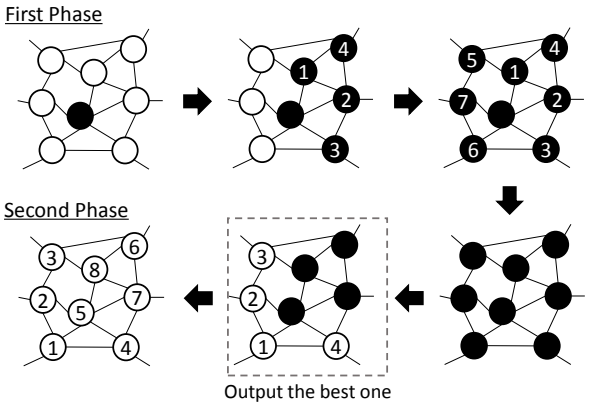

Figure 3: Illustration of Execution of Accordion Search

S' close to S^*. Therefore, the condition to terminate the first expand phase should be set so that $|S|$ is at least $|S^*|$. Since the exact $|S^*|$ is unknown and we know only $|S^*| \leq |V|$, we can terminate the first expand phase when S reaches V. However, this is not a good strategy in terms of the computation time, because we know $|S^*|$ is typically far smaller than $|V|$ and we can save time if we terminate the first phase sooner. In our implementation, we terminated the first expand phase when $f(S) < f(S')/5$ or $|S| > 10000$.

Besides the termination condition, the strategy to choose a good initial solution is also important for this algorithm, since the approximation quality of its output strongly depends on the initial solution given as input. In our implementation, we compute the *triangle ratio* $t(v)/d(v)$, which was originally introduced in [19], for each vertex $v \in V$ where $t(v)$ is the number of triangles that involve v and $d(v)$ is the degree of v. Then we choose the top-k triangle ratio vertices $v_1, v_2, \ldots, v_k$ and create a set of k initial solutions $S_1 = \{v_1\}$, $S_2 = \{v_2\}$, ..., and $S_k = \{v_k\}$. Finally, we apply our accordion search algorithm to each of the S_i $(i = 1, \ldots, k)$ and output the best solution from k runs. Intuitively, by using this multi-start strategy, we intend to increase the possibility that at least one of S_i is contained in S^*.

Table 2: Graph Characteristics

Graph Name	$\|V\|$	$\|E\|$	$\|E\|/\|V\|$
dolphins	62	159	2.56
polbooks	105	441	4.20
adjnoun	112	425	3.79
celegansneural	297	2345	7.90
bcspwr05	443	1033	2.33
celegans_metabolic	453	2025	4.47
email	1133	5451	4.81
bcspwr06	1454	3377	2.32
wb-cs-stanford	9914	27 427	2.77
p2p-Gnutella31	62 586	147 892	2.36
Wordnet3	82 670	120 399	1.46
internet	124 651	193 620	1.55
in-2004	1 382 908	13 591 473	9.83
as-Skitter	1 696 415	11 095 298	6.54
patents	3 774 768	14 970 766	3.97

6 Experiments

Here, we present the experimental results of our new density metric and the exact and approximate algorithms. All of the experiments were done on a workstation equipped with an Intel Xeon (E5540) with eight cores running at 2.53 GHz, 52 GB of RAM, and Red Hat Enterprise Linux Workstation 6.4. We wrote all of the programs in C++ and used the gcc 4.4.7 compiler with the `-O3` option. We used IBM ILOG CPLEX Version 12.5.1 as a solver for MIQP with the default parameter settings, including the number of threads (which was set to 16). All of the execution times exclude the times for reading the input files and are expressed in seconds. We obtained the graph data from the University of Florida Sparse Matrix Collection [7]. Table 2 summarizes the basic properties of the graphs used in the experiments, where the second and third columns show the numbers of the vertices and edges, respectively. We converted all of the graphs into simple undirected graphs by removing self-loops and redundant edges.

First, we measured the computation times of the exact algorithm for small real-world graphs. The results are summarized in Table 3, where the second and third columns respectively show the computation times for solving the densest subgraph problem using our Discounted Average Degree (DAD) metric with $\beta = 1.5$ and the Quasi-Clique (QC) metric with $\alpha = 1/3$, respectively. These results indicate that our exact algorithm can solve the densest subgraph problem for small graphs in a reasonable amount of times. The numbers in parenthesis in the second column show the numbers of iterations of Algorithm 1 and mean that the iterations are sufficiently small in number. Overall, the results of Table 3 suggest that our exact algorithm takes more time for graphs with higher average degrees ($|E|/|V|$). To confirm this, we conducted additional experiments on random graphs generated by an Erdös-Rényi model. We generated random graphs with a $G(n, p)$ model, wherein each graph contained $n = 50$ vertices and each edge was included with a probability p independently of every other edge. Table 4 shows the computation times of our exact algorithm for various p when using the DAD and QC metrics. Each computation time in this table is the geometric mean of the computation times for five randomly generated instances. The results show that the computation time increases as p increases as long as $p \leq 0.3$.

Next, we compared the optimum solutions of the densest subgraph problem for the graph *dolphins* using the DAD metric with β from $\{1.001, 1.2, 1.4, \ldots, 2.0\}$ and the QC metric with α from $\{1/3, 2/3, 0.999\}$. The results are summarized in Table 5, where the second column $|S|$ shows the size of the optimum subgraph, the third column δ shows the clique ratio ($= e[S]/\binom{|S|}{2}$), the fourth column D shows the diameter (the shortest path length between most distant vertices in $G[S]$), the fifth column τ shows the triangle ratio (the number of triangles in S divided by the number of triangles in a clique of size $|S|$), and the sixth column $\frac{\lambda(G[S])}{|S|-1}$ shows the edge-connectivity normalized to 1 (the edge-connectivity $\lambda(G[S])$ for graph $G[S]$ is defined as the minimum number of edges that must be deleted to disconnect $G[S]$ and it takes values between 0 and $|S| - 1$). Just for reference, the optimum subgraphs for the DAD metric with $\beta = 1.001$ and $\beta = 2.0$ and for the QC metric with $\alpha = 1/3$ and $\alpha = 0.999$ are shown in Figures 4 to 6, where the vertices in the optimum subgraphs are colored in red. These results show that the size of the optimum subgraph for the DAD metric decreases as β increases. The size of the optimum subgraph for the QC metric also

Table 3: Execution Times (in sec.) of the Exact Algorithm for Small Graphs

Graph Name	DAD ($\beta = 1.5$)	QC ($\alpha = 1/3$)
dolphins	0.61 (1)	0.46
polbooks	1.21 (2)	0.22
adjnoun	1.30 (2)	1.84
celegansneural	445.14 (2)	23.73
bcspwr05	8.23 (2)	1.37
celegans_metabolic	12.49 (2)	5.65
email	2743.37 (1)	1679.26
bcspwr06	101.71 (1)	13.97

Table 4: Execution Times (in sec.) of the Exact Algorithm for Random Graphs

n	p	DAD ($\beta = 1.5$)	QC ($\alpha = 1/3$)
50	0.1	0.32	0.38
	0.2	2.24	1.77
	0.3	13.57	11.65
	0.4	34.07	2.08
	0.5	58.67	0.53
	0.6	81.95	0.15
	0.7	127.45	0.49
	0.8	190.59	0.50
	0.9	315.88	0.55

decreases as α increases, but the optimum size for this metric with $\alpha = 1/3$ is smaller than that for the DAD metric with small parameters such as $\beta = 1.001$ and $\beta = 1.2$. This means we should use the DAD metric rather than the QC metric if we want to find a large dense graph.

To see the difference in the sizes of the densest subgraphs, we compared the sizes of the optimum subgraphs for different values of α and β of the QC and DAD metrics for other graphs. Table 6 summarizes the results. The second and third columns of the table show the sizes of the optimum subgraphs when we vary the parameters of the DAD and QC metrics between $1/3 \leq \alpha < 1$ and $1 < \beta \leq 2$. It is not surprising that the QC and DAD metrics are the same for the lower values of the ranges, because both metrics favor a clique over a near-clique if α and β are large. In contrast, the upper values of the ranges of the QC metric are smaller than those for the DAD metric for all of the graphs. This means that the DAD metric has an advantage over the QC metric in the sense that it gives us a broader range of control over the output size by changing the parameter. Meanwhile, remember that Tables 3 and 4 tell us that we should use the QC metric when the execution time needed to obtain the optimum solution with the exact algorithms is important.

Table 5: Characteristics of Optimum Subgraphs for the *dolphins* Graph

Density Metric	$\|S\|$	δ	D	τ	$\frac{\lambda(G[S])}{\|S\|-1}$
DAD ($\beta = 1.001$)	20	0.326	3	0.042	0.211
DAD ($\beta = 1.2$)	17	0.382	3	0.065	0.250
DAD ($\beta = 1.4$)	6	0.933	2	0.800	0.800
DAD ($\beta = 1.6$)	6	0.933	2	0.800	0.800
DAD ($\beta = 1.8$)	6	0.933	2	0.800	0.800
DAD ($\beta = 2.0$)	5	1.000	1	1.000	1.000
QC ($\alpha = 1/3$)	9	0.639	3	0.262	0.375
QC ($\alpha = 2/3$)	6	0.933	2	0.800	0.800
QC ($\alpha = 0.999$)	5	1.000	1	1.000	1.000

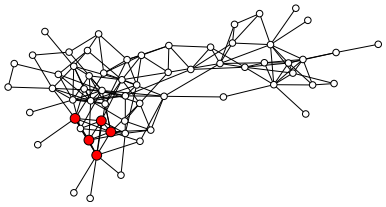

Figure 4: Optimum subgraph with $\alpha = 0.999$ and $\beta = 2.0$

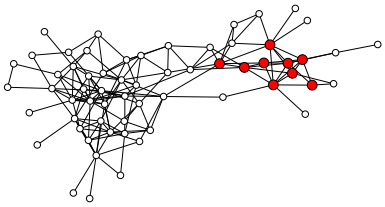

Figure 5: Optimum subgraph with $\alpha = 1/3$

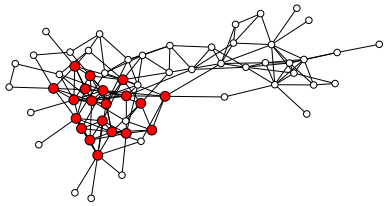

Figure 6: Optimum subgraph with $\beta = 1.001$

Table 6: Sizes of Optimum Subgraphs

Graph	DAD ($1 < \beta \leq 2$)	QC ($1/3 \leq \alpha < 1$)
dolphins	5–**20**	5–**9**
polbooks	6–**24**	6–**17**
adjnoun	5–**48**	5–**16**
celegansneural	8–**136**	8–**25**
bcspwr05	3–**62**	3–**5**
celegans_metabolic	9–**45**	9–**28**
bcspwr06	4–**23**	4–**6**

Next, we evaluated the approximation performance of our new heuristic algorithm against the best two current algorithms: a greedy algorithm and a local search algorithm.

- **Greedy algorithm** presented by Charikar [6]. This algorithm was originally designed for the densest subgraph problem using the average degree metric, but we extended it so that it could use other density metrics. The extended algorithm iteratively removes vertices with the minimum degree from the graph G until the graph becomes empty and outputs the best subgraph that appeared during the execution. This extended algorithm can be viewed as a special case of our accordion search algorithm when we use the (discounted) average degree metric, because our algorithm is equivalent to this greedy algorithm when the whole set of vertices V is given as input. If V is given as the initial solution, the first expand phase of our algorithm is skipped, and the second phase is equivalent to this greedy algorithm.
- **Local search algorithm** presented by Tsourakakis et al. [19]. This algorithm receives an initial solution S, and updates S by adding a vertex v to S or by removing a vertex v from S until $f(S)$ cannot be improved by adding or removing a single vertex. In the original paper [19], this local search is supposed to start with an initial solution $\{v^*\} \cup N(v^*)$, where v^* is the vertex with the highest triangle ratio (see Sec. 5 for the definition of the triangle ratio) and $N(v^*)$ is the set of the vertices adjacent to v^* in G. In our experiments, we extended this single-start local search into a multi-start local search algorithm. Specifically, we first extract the top-k triangle ratio vertices v_1, v_2, …, v_k. Then we apply the local search to each of $\{v_1\} \cup N(v_1)$, $\{v_2\} \cup N(v_2)$, …, $\{v_k\} \cup N(v_k)$ and output the best subgraph from k runs.

In our experiments, we implemented our accordion search algorithm and these two heuristic algorithms as single-thread programs, although it is easy to parallelize our accordion search algorithm and the local search algorithm with k initial solutions by using k threads. For the computation of the triangle ratio, we used the approximate triangle counting algorithm presented in [15] for large graphs.

First, we compared the solutions obtained by the three algorithms. Table 7 shows the results for the graph *p2p-Gnutella31* using the DAD metric with β from $\{1.1, 1.5, 1.9\}$. The third column shows the number of initial solutions used for the local search algorithm and our accordion search algorithm, the fourth column shows the size of the output subgraph, and the fifth column shows the objective value of the obtained solution. Comparing the results for the local search algorithm when we use $\beta = 1.1$, we see that the solution obtained by the single-start local search algorithm is worse than those obtained by the multi-start local search algorithm. We also see that the solution obtained by the single-start accordion search algorithm is worse than those obtained by the multi-start accordion search algorithm, when we use $\beta = 1.5$. These results prove that our multi-start strategy is effective at improving the quality of the solutions. Apparently, the objective value of the solution obtained by a multi-start algorithm can be improved by

Table 7: Results of Heuristic Algorithms for Graph *p2p-Gnutella31*

β	Algorithm	#init	$\|S\|$	$f(S)$	Time (sec.)
1.1	Local Search	1	6	1.3933	0.047
	Local Search	10	22	2.1022	0.122
	Local Search	100	22	2.1022	0.853
	Greedy		23	2.6692	0.035
	Accordion (new)	1	23	2.6692	0.099
	Accordion (new)	10	23	2.6692	0.572
	Accordion (new)	100	23	2.6692	5.369
1.5	Local Search	1	4	0.7500	0.042
	Local Search	10	4	0.7500	0.104
	Local Search	100	4	0.7500	0.723
	Greedy		14	0.8400	0.031
	Accordion (new)	1	11	0.7949	0.044
	Accordion (new)	10	14	0.8400	0.129
	Accordion (new)	100	14	0.8400	1.052
1.9	Local Search	1	4	0.4308	0.044
	Local Search	10	4	0.4308	0.115
	Local Search	100	4	0.4308	0.835
	Greedy		10	0.3147	0.032
	Accordion (new)	1	4	0.4308	0.044
	Accordion (new)	10	4	0.4308	0.117
	Accordion (new)	100	4	0.4308	0.860

increasing the number of initial solutions, but we should also notice, upon examining the results for the local search with $\beta = 1.1$, even if we use as many as 100 initial solutions, the quality of the solutions with the local search remains the same as that of the solution with 10 initial solutions and worse than those obtained by the greedy and accordion algorithms. This tells us that increasing the number of initial solutions does not necessarily lead to the optimum (or even a near-optimum) solution. We should also notice that, whereas the greedy algorithm outputs the best results when we use $\beta = 1.1$ and $\beta = 1.5$, this algorithm is worse than the local search algorithm and the accordion search algorithm when we use $\beta = 1.9$. As we have seen, at least for the graph p2p-Gnutella31 with β from $\{1.1, 1.5, 1.9\}$, the accordion search algorithm performs the best in terms of the quality of the solution. We conducted further experiments on other large graphs. In these experiments, the numbers of initial solutions for the local search algorithm and our accordion search algorithm are set to 25, because the quality of the solutions of the accordion search algorithm are the same even if we use 100 initial solutions for most of the graphs. The results are summarized in Table 8. An objective value that has been underlined is smaller than 0.9 times the maximum objective value obtained by the three heuristic algorithms, and hence, an underlined value means that the solution is far from optimum. The comparison tells us that, while the local search and greedy algorithm fail to produce near-best solutions for some graphs, all of the solutions obtained by our accordion search algorithm are the best or nearly the best for the graphs we tested. To confirm that our accordion search algorithm outputs optimum or near-optimum solutions, we conducted additional experiments on the seven small graphs (dolphins, polbooks, adjnoun, celegansneural, bcspwr05, celegans_metabolic, and bcspwr06) by using the DAD metric with parameter β from $\{1.1, 1.3, 1.5, 1.7, 1.9\}$. Out of 35 ($= 7 \times 5$) combinations of the graph and the parameter β, we confirmed that the 33 solutions obtained by the accordion search algorithm matched the optimum solutions obtained by the exact algorithm.

Finally, we should note that the results shown in Tables 7 and 8 compare the solutions in terms of their quality, *not* the computation times of the three algorithms. The execution times in the tables are just for reference, because the implementations of the three algorithms are not optimized to reduce the computation time. We should be aware that we can easily reduce the execution times of the local search algorithm and the accordion search algorithm with multiple initial solutions by using multi-threading.

Table 8: Results of Heuristic Algorithms for Large Graphs (execution times are in sec.)

		Local Search (#init=25)			Greedy Search			Our Accordion Search (#init=25)		
β	Graph Name	$\|S\|$	$f(S)$	Time	$\|S\|$	$f(S)$	Time	$\|S\|$	$f(S)$	Time
1.1	wb-cs-stanford	44	9.5581	0.026	44	9.5581	0.004	44	9.5581	0.107
	p2p-Gnutella31	22	2.1022	0.243	23	2.6692	0.035	23	2.6692	1.396
	Wordnet3	42	2.6542	0.231	76	2.6794	0.036	67	2.6956	0.821
	internet	24	4.7000	0.381	47	4.1695	0.064	24	4.7000	0.742
	in-2004	1920	156.2330	9.372	492	132.0730	1.442	2021	156.2630	10.346
	as-Skitter	420	48.6402	13.107	420	48.6402	2.012	420	48.6402	23.714
	patents	30	3.5110	26.534	49	8.5185	6.412	55	8.5982	35.029
1.3	wb-cs-stanford	35	4.5530	0.027	35	4.5530	0.004	35	4.5530	0.061
	p2p-Gnutella31	22	1.1329	0.234	19	1.4578	0.033	19	1.4578	0.899
	Wordnet3	16	1.3330	0.230	15	1.3609	0.035	18	1.3772	0.639
	internet	21	2.5026	0.390	23	2.2234	0.064	21	2.5026	0.440
	in-2004	492	38.2316	9.442	492	38.2316	1.452	492	38.2316	10.320
	as-Skitter	286	15.1056	13.122	288	15.1059	2.082	288	15.1059	22.956
	patents	12	1.7794	27.766	49	3.9113	6.410	55	3.8577	29.515
1.5	wb-cs-stanford	28	2.2610	0.025	32	2.2484	0.004	28	2.2610	0.046
	p2p-Gnutella31	4	0.7500	0.207	14	0.8400	0.031	14	0.8400	0.279
	Wordnet3	6	0.8165	0.195	10	0.8222	0.032	11	0.8223	0.284
	internet	19	1.3644	0.357	23	1.1876	0.061	19	1.3644	0.363
	in-2004	492	11.0670	8.911	492	11.0670	1.393	492	11.0670	9.750
	as-Skitter	210	5.0421	12.315	210	5.0421	2.057	210	5.0421	19.916
	patents	8	0.9281	26.453	49	1.7959	6.187	54	1.7313	27.457
1.7	wb-cs-stanford	23	1.1961	0.026	23	1.1961	0.004	23	1.1961	0.039
	p2p-Gnutella31	4	0.5684	0.241	10	0.4988	0.033	4	0.5684	0.251
	Wordnet3	5	0.5834	0.224	4	0.5684	0.035	5	0.5834	0.247
	internet	8	0.7872	0.385	23	0.6344	0.066	10	0.7981	0.394
	in-2004	492	3.2036	9.422	492	3.2036	1.440	492	3.2036	10.167
	as-Skitter	112	1.8384	12.894	133	1.8046	2.139	88	1.8118	17.666
	patents	5	0.6483	27.923	48	0.8249	6.360	13	0.8430	28.173
1.9	wb-cs-stanford	21	0.6426	0.026	21	0.6426	0.004	21	0.6426	0.033
	p2p-Gnutella31	4	0.4308	0.234	10	0.3147	0.032	4	0.4308	0.237
	Wordnet3	4	0.4308	0.215	4	0.4308	0.034	4	0.4308	0.226
	internet	7	0.5206	0.365	22	0.3406	0.062	7	0.5206	0.369
	in-2004	492	0.9273	9.060	492	0.9273	1.401	492	0.9273	9.961
	as-Skitter	75	0.7471	12.331	73	0.7453	2.081	74	0.7535	13.847
	patents	5	0.4698	27.120	8	0.4809	6.210	9	0.5075	27.513

7 Parameter Settings for Density Metrics

A remaining question regarding the densest subgraph problem is how to choose the parameters for a given density metric (such as β for the discounted average degree metric). Since the best parameters depend on the applications and the parameters intrinsically represent preferences of the user even if the application is determined, there is no single recommended parameter settings that work for all of the applications and their users. We think that the parameters should be determined experimentally. That is, we first solve the densest subgraph problem with an arbitrary parameter setting and repeatedly change the parameters until we get a desired subgraph.

However, for those who want to avoid numerous trials, we can use a standard technique from *learning to rank* to determine the best parameters. First, we collect a set of subgraphs $D = \{S_1, S_2, \ldots, S_n\}$ to be ordered. Then we specify preferences as a set of ordered pairs $O = \{(S_{i_k} \succ S_{j_k}) \mid k = 1, \ldots, m\}$ where each pair $S_{i_k} \succ S_{j_k}$ indicates that $f(S_{i_k})$ should be larger than $f(S_{j_k})$. Finally we formulate a problem to find a set of parameters that conform to O. For example, if we are interested in the discounted average degree metric $f(S) = e[S]/|S|^{\beta}$, we can formulate the

problem as

$$\begin{aligned} \min \quad & \sum_{k=1}^{m} w_k \\ \text{subject to} \quad & \log \frac{f(S_{j_k})}{f(S_{i_k})} = \log \frac{e[S_{j_k}]}{e[S_{i_k}]} \frac{|S_{i_k}|^\beta}{|S_{j_k}|^\beta} \leq w_k \quad \forall k \\ & w_k \geq 0 \qquad\qquad\qquad\qquad\quad \forall k. \end{aligned}$$

Here, each w_k represents a penalty for violating the order $S_{i_k} \succ S_{j_k}$, and $w_k = 0$ means that the order is satisfied. We can use the optimum solution β^* of this formulation for the parameter of the discounted average degree metric.

8 NP-Hardness of the Discounted Average Degree Metric

Finally, we show that it is NP-hard to obtain an optimum solution for the densest subgraph problem by using the discounted average degree metric with the parameter setting $\beta = 2$. To show this, we will use two well-known theorems.

Theorem 8.1 *[8] One can construct a graph $G = (V, E)$ such that it is NP-hard to distinguish between these two cases.*

(i) G contains a clique of size at least $|V|/3$ or

(ii) G does not contain a clique of size $|V|/9$.

This theorem tells us that, even if we know that the size of the largest clique in G is not between the two cases (i) and (ii), it is hard to tell if the largest clique is large (Case (i)) or small (Case (ii)). Note that this theorem states that it is hard even to *decide* which case G belongs to, since we are not asked to find a clique of size at least $|V|/3$ for Case (i) as evidence.

We will also use another famous theorem from graph theory.

Theorem 8.2 *[20] (Turán's Theorem.) If $G[S^*]$ does not contain a clique of size $r + 1$, then*

$$e[S^*] \leq \frac{|S^*|^2}{2}\left(1 - \frac{1}{r}\right).$$

Using these theorems, we will show the NP-hardness of the problem.

Theorem 8.3 *Given a graph $G = (V, E)$, it is NP-hard to find the subgraph $S \subseteq V$ that maximizes $f(S) = e[S]/|S|^2$.*

Proof. To prove this theorem, it is sufficient to prove this claim: a graph $G = (V, E)$ contains a clique of size $|V|/3$ if and only if there exists a subset $S^* \subseteq V$ such that $f(S^*) \geq 1/2 - 3/2|V|$. If this claim is true, then we can decide whether G contains a clique of size at least $|V|/3$ by checking whether $f(S^*) \geq 1/2 - 3/2|V|$, where $S^* \subseteq V$ is a vertex set that maximizes the density metric $f(S) = e[S]/|S|^2$.

Here is the proof of this claim. If G contains a clique C of size $|V|/3$, then we have

$$f(C) = \frac{e[C]}{|C|^2} = \frac{(|V|/3)(|V|/3 - 1)/2}{|V|^2/9} = \frac{1}{2} - \frac{3}{2|V|},$$

which means that $f(S^*) \geq 1/2 - 3/2|V|$.

Next we show the opposite direction of the claim. If we have a set S^* such that $f(S^*) \geq 1/2 - 3/2|V|$, then we have

$$e[S^*] = f(S^*)|S^*|^2 \geq \frac{|S^*|^2}{2}\left(1 - \frac{3}{|V|}\right). \tag{2}$$

As a corollary of Turán's Theorem (Theorem 8.2), we know that, if

$$e[S^*] > \frac{|S^*|^2}{2}\left(1 - \frac{1}{r}\right), \tag{3}$$

then $G[S^*]$ contains a clique of size $r + 1$. Since we know that Inequality (3) holds whenever $r < |V|/3$ from Inequality (2), we can conclude that $G[S^*]$ contains a clique of size $|V|/3$. □

9 Conclusions

We have presented four axioms for choosing an appropriate density metric for the densest subgraph problem, which allows us to avoid counterintuitive results due to the use of an inappropriate metric. In addition, we have proposed a new density metric, the discounted average degree, which satisfies all of the axioms and has a broader range of control than the quasi-clique metric. As for the algorithms, we have presented exact and approximate algorithms that can be used with typical density metrics, including the quasi-clique metric and our discounted average metric. The exact algorithm is useful for evaluating various density metrics and for verifying the approximation performance of heuristic algorithms. Our approximate algorithm outperforms the current best algorithms in terms of approximation quality for the graphs we tested.

References

[1] J. Abello, M. G. C. Resende, and S. Sudarsky. Massive quasi-clique detection. In *LATIN 2002*, pages 598–612, 2002.

[2] A. Altman and M. Tennenholtz. Ranking systems: the PageRank axioms. In *EC 2005*, pages 1–8, 2005.

[3] A. Angel, N. Koudas, N. Sarkas, and D. Srivastava. Dense subgraph maintenance under streaming edge weight updates for realtime story identification. *PVLDB*, 5(6):574–585, 2012.

[4] P. Boldi and S. Vigna. Axioms for centrality. *Internet Mathematics*, 10(3–4):222–262, 2014.

[5] C. G. Chakrabarti and I. Chakrabarty. Shannon entropy: axiomatic characterization and application. *International Journal of Mathematics and Mathematical Sciences*, 2005(17):2847–2854, 2005.

[6] M. Charikar. Greedy approximation algorithms for finding dense components in a graph. In *APPROX 2000*, pages 84–95, 2000.

[7] T. A. Davis and Y. Hu. The university of Florida sparse matrix collection. *ACM Transactions on Mathematical Software*, 38(1):Article No. 1, 2011. http://www.cise.ufl.edu/research/sparse/matrices.

[8] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.

[9] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.

[10] A. V. Goldberg. Finding a maximum density subgraph. Technical report, University of California, 1984.

[11] H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. *BIOINFORMATICS*, 21:213–221, 2005.

[12] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal. A survey of algorithms for dense subgraph discovery. *Advances in Database Systems*, 40:303–336, 2010.

[13] R. D. Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15(2):169–190, 1950.

[14] R. J. Mokken. Cliques, clubs and clans. *Quality and Quantity*, 13(2):161–173, 1979.

[15] R. Pagh and C. E. Tsourakakis. Colorful triangle counting and a MapReduce implementation. *Information Processing Letters*, 112(7):277–281, 2012.

[16] S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983.

[17] S. B. Seidman and B. L. Foster. A graph-theoretic generalization of the clique concept. *The Journal of Mathematical Sociology*, 6(1):139–154, 1978.

[18] C. E. Tsourakakis. A novel approach to finding near-cliques: The triangle-densest subgraph problem. Technical report, ICERM, Brown University, 2014.

[19] C. E. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. A. Tsiarli. Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees. In *KDD 2013*, pages 104–112, 2013.

[20] P. Turán. Egy gráfelméleti szélsöértékfeladatról (in Hungarian). *Mat. Fiz. Lapok*, 48:436–452, 1941.

[21] N. Wang, J. Zhang, K.-L. Tan, and A. K. H. Tung. On triangulation-based dense neighborhood graph discovery. *PVLDB*, 4(2):58–68, 2010.