This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3444608

Mining Triangle-Dense Subgraphs of a Fixed Size: Hardness, Lovász extension and Applications

Aritra Konar, and Nicholas D. Sidiropoulos, *Fellow, IEEE*

Abstract—We introduce the triangle-densest-k-subgraph problem (TDkS) for undirected graphs: given a size parameter k, compute a subset of k vertices that maximizes the number of induced triangles. The problem corresponds to the simplest generalization of the edge-based densest-k-subgraph problem (DkS) to the case of higher-order network motifs. We prove that TDkS is NP-hard and is not amenable to efficient approximation, in the worst-case. By judiciously exploiting the structure of the problem, we propose a relaxation algorithm for the purpose of obtaining high-quality, sub-optimal solutions. Our approach utilizes the fact that the cost function of TDkS is submodular to construct a convex relaxation for the problem based on the Lovász extension for submodular functions. We demonstrate that our approaches attain state-of-the-art performance on real-world graphs and can offer substantially improved exploration of the optimal density-size curve compared to sophisticated approximation baselines for DkS. We use document summarization to showcase why TDkS is a useful generalization of DkS.

Index Terms—Dense subgraph discovery, triangle-motifs, intractability and approximation, submodularity, convex optimization.

1 INTRODUCTION

The task of extracting dense subgraphs from a given graph has diverse applications in graph mining ranging from fraud detection [1], [2], chemical informatics [3], computational biology [4] and knowledge discovery [5]–[7]. Owing to its practical relevance, the problem has received extensive attention (see [8]–[10] and references therein) - we briefly highlight some prominent formulations.

Given an undirected graph, the classic densest-subgraph (DS) problem [11] aims to detect the subgraph with the maximum average induced degree. The problem is known to be polynomial-time solvable, and admits a simple linear-time $1/2$ approximation via a greedy algorithm [12]. These ideas have also been extended to directed graphs (see [13] and references therein). However, real-world examples are known [14] where the greedy algorithm returns the trivial solution corresponding to the graph itself as the densest subgraph. This undesirable behavior can be attributed to the fact that the approach does not allow explicit specification of the desired subgraph size. Adding a simple size constraint to the DS problem results in the densest-k-subgraph (DkS) problem [15], which, for a specified node-size k, aims to find the subgraph with the maximum number of induced edges. Unfortunately, the constraint also renders DkS NP–hard. Moreover, the problem is notorious for being very difficult to approximate, in the worst-case sense [16]–[18]. Notwithstanding such pessimistic results, polynomial-time algorithms which work well in practice for DkS are known – these include low (constant) rank matrix approximation techniques [19] and a recent work [20], which uses tools from submodular optimization to construct a convex relaxation for DkS. Another size constrained variant, Dam$k$S, [21] maximizes average degree subject to the size of the subgraph being *at most* k, and is NP–hard. Meanwhile, the Dal$k$S problem [21] maximizes average degree subject to the subgraph size being *at least* k. Dal$k$S is also NP–hard [22], with greedy and linear programming approximation algorithms known.

A salient feature of the aforementioned formulations is that they quantify subgraph density in terms of induced edges, which represent pair-wise relationships between vertices. However, real-world graphs are often rich in higher-order motifs, which signify stronger associations among vertices compared to pair-wise relationships alone [23]. This suggests that leveraging higher-order motif structure for dense subgraph discovery can detect subgraphs which are more clique-like compared to those obtained via the edge-based formulations. For example, prior work [24] has introduced the ℓ-clique densest subgraph problem to extract the subgraph with the largest average number of induced ℓ-cliques. This is a generalization of the DS problem (the latter corresponds to choosing $\ell = 2$) which remains polynomial-time solvable and also admits effective approximation via a greedy algorithm [24]. More importantly, applying this formulation with $\ell = 3$ (triangles, the simplest example of a higher-order motif) to real-world graphs yields subgraphs of higher *edge density* compared to using the DS with $\ell = 2$. However, like its edge-based counterpart, the ℓ-clique densest subgraph problem formulation does not provide a means of explicitly controlling the desired subgraph size. We argue that this is a restrictive feature, since it does not allow the end-user the flexibility in picking a desired solution. By varying the size explicitly, one can obtain small subsets of vertices which are tightly knit, to larger subsets which exhibit smaller density, and everything in-between.

For this purpose, in this paper, we introduce the triangle-

- *A. Konar is with the Dept. of Electrical Engineering (ESAT), KU Leuven, Belgium. E-mail: aritra.konar@kuleuven.be*
- *Nicholas D. Sidiropoulos is with the Dept. of Electrical and Computer Engineering, University of Virginia, USA. E-mail: nikos@virginia.edu*

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3444608

densest-k-subgraph problem (TDkS). Given an undirected graph $\mathcal{G}$ on n vertices and a desired subgraph size k, we aim to compute the subgraph with the maximum number of induced triangles over all possible $\binom{n}{k}$ subgraphs. Clearly, TDkS is the simplest higher-order generalization of its edge-based counterpart DkS. To the best of our knowledge, however, this is the first time that the problem has been studied. Can adopting such a formulation enable us to discover denser subgraphs compared to DkS, and thereby do a better job at exploring the optimal density-size curve on real-world graphs? Can TDkS extract more meaningful subsets than DkS in real-world applications? These are the main questions considered in our paper. Given this context, our contributions can be summarized as follows.

• **Hardness:** We prove that TDkS is NP-hard in the worst-case. Additionally, we show that it is difficult to obtain a favorable approximation of the optimal objective value of TDkS in polynomial time.

• **Submodular relaxation and algorithm:** Not withstanding such pessimistic worst-case results, we focus on developing an approximation algorithm which can work well on real-world instances. We show and leverage the fact that the discrete cost function of TDkS is endowed with a specific type of combinatorial structure - namely, it is a submodular function. As such functions possess a unique, continuous, convex extension (i.e., the Lovász extension), we devise a convex relaxation for TDkS that minimizes the Lovász extension over the convex hull of the cardinality constraints. Additionally, a key technical contribution of our paper is to show that for TDkS, the Lovász extension admits an analytical functional form, which is difficult to determine for general submodular functions. We exploit this structure to develop a scalable Mirror Descent algorithm for solving the problem, which, combined with a simple rounding procedure, can be employed for extracting candidate triangle-dense subgraphs.

• **Experiments:** Our experiments reveal that the proposed approach is very effective in mining triangle-dense subgraphs on real-world datasets. Interestingly, it can also extract subgraphs of higher edge density than state-of-the-art DkS baselines, which is a bonus. Our experiments further indicate that when TDkS is used for unsupervised document summarization it yields more meaningful and interpretable summaries than DkS does.

We point out that, at a high level, our use of the Lovász relaxation for TDkS is in the spirit of [20] which introduced the Lovász relaxation for DkS. That being said, there are also important differences (apart from the fact that the two problems are distinct), the one key being that computing an analytical functional form for the Lovász extension of TDkS is substantially more challenging compared to the classical edge-based case. Additionally, the form that the Lovász extension of TDkS takes is more complicated than that for DkS, which necessitates an entirely different algorithmic approach. Finally, to put our contributions into broader context, several recent works [24]–[27] have considered generalizing classical edge-based graph mining tasks to account for higher-order network motifs. Our present work seeks to contribute to this thread of research by developing new tools for tackling a challenging problem in this area.

• **Summary of Differences:** A conference version of this work has previously appeared in [28]. Relative to that, the present journal version adds a new section on problem motivation to highlight the conceptual appeal of our formulation, additional experiments including a comprehensive and insightful case study on document summarization, and fully fleshed out technical proofs.

2 PRIMER ON SUBMODULARITY

We provide a brief overview of basic concepts regarding submodular functions [29]–[31]. For a set of n objects $\mathcal{V} = \{1, \cdots, n\}$, a *set function* $F : 2^{|\mathcal{V}|} \rightarrow \mathbb{R}$ assigns a real value to any subset $\mathcal{S} \subseteq \mathcal{V}$. A set function F is said to be *submodular* if and only if $F(\mathcal{A} \cup \mathcal{B}) + F(\mathcal{A} \cap \mathcal{B}) \leq F(\mathcal{A}) + F(\mathcal{B})$ for all subsets $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$. For the special case where $n = 2$ and $\mathcal{V} = \{a, b\}$, the above condition simplifies to $F(\emptyset) + F(\mathcal{V}) \leq F(\{a\}) + F(\{b\})$. A notable feature of submodular functions is that they possesses a continuous, *convex* extension known as the Lovász extension, which extends their domain from $2^{|\mathcal{V}|}$ to the unit interval $[0, 1]^n$ (recall $n = |\mathcal{V}|$). Formally, the Lovász extension $f_L : [0, 1]^n \rightarrow \mathbb{R}$ of a given submodular function F is

$$f_L(\mathbf{x}) := \max_{\mathbf{g} \in \mathcal{B}_F} \mathbf{g}^T \mathbf{x}, \tag{1}$$

where the set $\mathcal{B}_F$ is the *base polytope* associated with F and is defined as

$$\mathcal{B}_F := \{\mathbf{g} \in \mathbb{R}^n : \mathbf{g}^T \mathbf{1}_{\mathcal{V}} = F(\mathcal{V});\ \mathbf{g}^T \mathbf{1}_{\mathcal{S}} \leq F(\mathcal{S}), \forall\ \mathcal{S} \subseteq \mathcal{V}\}. \tag{2}$$

It can be seen that the Lovász extension is the support function of the base polytope $\mathcal{B}_F$, and is thus a convex function. In fact, f_L is convex if and only if F is submodular [29]. Furthermore, when evaluated at a binary vector $\mathbf{x} \in \{0, 1\}^n$, the Lovász extension equals the value of the submodular function F.

3 PROBLEM STATEMENT

Consider an undirected graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ on n vertices. Given a size parameter $k \in \{4, \cdots, n\}$ and a subset of k vertices $\mathcal{S} \subseteq \mathcal{V}$, let $\rho_2(\mathcal{S}, k)$ denote the edge density of the subgraph $\mathcal{G}_{\mathcal{S}}$ induced by $\mathcal{S}$. This quantity equals the ratio of the sum of the induced edges and the maximum possible number of induced edges $\binom{k}{2}$. In an analogous fashion, let $\rho_3(\mathcal{S}, k)$ denote the triangle density of $\mathcal{S}$. In this paper, we consider the problem of extracting the subgraph of size k that exhibits the maximum sum of induced triangles. Let $\mathcal{X}_k := \{\mathbf{x} \in \{0, 1\}^n : \mathbf{1}^T \mathbf{x} = k\}$ be the set of all binary vectors with k non-zero entries. Formally, the triangle-densest-k-subgraph (TDkS) problem can be expressed as

$$\max_{\mathbf{x} \in \mathcal{X}_k} \left\{ f(\mathbf{x}) := \sum_{\{u,v,w\} \in \Delta} w_t x_u x_v x_w \right\}, \tag{3}$$

where Δ denotes the set of triangles in the graph (each counted once), and w_t is a positive weight associated with triangle $t := \{u, v, w\} \in \Delta$ [1].

Let $\mathbf{x}^*$ denote an optimal solution of (3) and $f(\mathbf{x}^*)$ represent the optimal sum of induced triangles. If we denote the optimal vertex subset as $\mathcal{S}^* = \{i \in \mathcal{V} : x_i^* = 1\}$, then

1. If $\mathcal{G}$ is unweighted, each triangle $t \in \Delta$ has weight $w_t = 1$.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3444608

the maximum triangle density is $\rho_3(\mathcal{S}^*, k) := f(\mathbf{x}^*)/\binom{k}{3}$. Note that by varying the size parameter k, TDkS outputs a spectrum of dense subgraphs. We designate this spectrum as the triangle density versus size curve - each point on this curve corresponds to a pair $(k, \rho_3(\mathcal{S}^*, k))$.

3.1 Motivation

It is evident that TDkS is a higher-order extension of DkS, i.e., it quantifies subgraph density with respect to triangles, as opposed to edges. Since the triangle motif is a higher-order clique compared to an edge, it is then natural to consider what additional benefits the TDkS formulation (3) can offer relative to DkS.

To this end, consider a subgraph $\mathcal{S} \subseteq \mathcal{V}$ on k vertices, with edge and triangle densities $\rho_2(\mathcal{S}, k)$ and $\rho_3(\mathcal{S}, k)$ respectively. It is known that these two quantities obey the following relationship.

Fact 1. ***The Kruskal-Katona Theorem [32], [33]:*** *For any subgraph, it always holds that*

$$\rho_2(\mathcal{S}, k) \geq \rho_3(\mathcal{S}, k)^{\frac{2}{3}}. \tag{4}$$

Hence, maximizing triangle density for a fixed subgraph size serves as a surrogate for maximizing edge density as well. The above fact formalizes the simple notion that maximizing the number of induced triangles in a subgraph also has the effect of increasing the number of induced edges.

Additionally, for a subgraph $\mathcal{S} \subseteq \mathcal{V}$, let $\tau(\mathcal{S})$ denotes its transitivity (i.e., global clustering coefficient). Recall that it can be expressed as

$$\tau(\mathcal{S}) = 3 \cdot \frac{\text{No. of triangles induced by } \mathcal{S}}{\text{No. of paths of length 2 in } \mathcal{S}}. \tag{5}$$

The value of subgraph transitivity always lies in the unit interval, and can be interpreted as the probability of sampling (uniformly at random) a path of length 2 from $\mathcal{G}_{\mathcal{S}}$, and having its endpoints connected to connected to form a triangle. Simply put, it determines how globally "triangle-like" the subgraph is. Hence, transitivity can serve as an alternate means of quantifying density with regard to triangle motifs. However, it turns out that transitivity obeys the following relationship with triangle density.

Lemma 1. *Given a subgraph $\mathcal{S}$ on $k := |\mathcal{S}|$ vertices, its transitivity $\tau(\mathcal{S})$ is at least its triangle density $\rho_3(\mathcal{S}, k)$, with equality if and only if $\mathcal{S}$ is a k-clique.*

Proof. The denominator of $\tau(\mathcal{S})$ can be expressed as $\sum_{u \in \mathcal{S}} \binom{d_u}{2}$, where d_u is the induced degree of vertex $u \in \mathcal{S}$. It then holds that

$$\frac{1}{3} \sum_{u \in \mathcal{S}} \binom{d_u}{2} \leq \frac{k}{3} \binom{d_{\max}}{2} \leq \frac{k}{3} \binom{k-1}{2} = \binom{k}{3}, \tag{6}$$

where $d_{\max} \leq k-1$ is the largest induced degree in $\mathcal{S}$. Consequently, we have

$$\tau(\mathcal{S}) \geq \frac{\text{No. of triangles induced by } \mathcal{S}}{\binom{k}{3}} = \rho_3(\mathcal{S}, k). \tag{7}$$

The above inequality is satisfied with equality if and only if all inequalities in (6) hold with equality, which is true if and only if the subgraph induced by $\mathcal{S}$ is a k-clique. □

The result is intuitive, since increasing the number of induced triangles in a subgraph cannot decrease its transitivity. Hence, for a fixed subgraph size, maximizing triangle density additionally serves as a surrogate for maximizing transitivity.

Thus far, we have demonstrated that employing triangle motifs for maximizing subgraph density via TDkS offers the twin advantages of additionally improving the edge density and transitivity. At this point, it is instructive to compare the following two problems - TDkS and its closest counterpart in the literature, the triangle-densest-subgraph (TDS) formulation [24], [34]. The TDS problem can be expressed as

$$\max_{\mathbf{z} \in \{0,1\}^n} \left\{ g(\mathbf{z}) := \frac{f(\mathbf{z})}{\mathbf{1}^T \mathbf{z}} \right\}. \tag{8}$$

Recall that $f(.)$ denotes the weighted sum of triangles induced by a vertex subset with indicator vector $\mathbf{z}$. The objective function of TDS assigns a greater reward to those subgraphs for which the average (induced) triangle degree of the vertices comprising the subgraph is large. In contrast to TDkS, the problem does not explicitly constrain the size of the desired subgraph. Interestingly, it turns out that we can establish a link between the solution of the two problems. Let $\mathbf{z}^*$ denote an optimal solution of (8) and $\mathcal{Z}^* = \{i \in \mathcal{V} : z_i^* = 1\}$ denote the optimal vertex subset. Then, we have the following claim.

Lemma 2. *For the choice of the size parameter $k = |\mathcal{Z}^*|$, $\mathcal{Z}^*$ is also an optimal solution for TDkS.*

Proof. We proceed via contradiction. Suppose that $\mathcal{Z}^*$ is not optimal for TDkS with $k = |\mathcal{Z}^*|$. Then, there exists a subgraph $\bar{\mathcal{Z}} \neq \mathcal{Z}^*$ on k vertices such that the weighted sum of triangles it induces $f(\bar{\mathbf{z}}) > f(\mathbf{z}^*)$. Since both subgraphs are of size k, this further implies that $g(\bar{\mathbf{z}}) > g(\mathbf{z}^*)$, which contradicts the optimality of $\mathcal{Z}^*$ for TDS. □

Thus, the solution of TDS corresponds to *a* point on the triangle-density versus size curve. The caveat of TDS, however, is that one cannot control the subgraph size explicitly. Hence, given a graph, one cannot predict *apriori* how large or small the size of the solution will be; i.e., pinpoint the exact location on the triangle-density versus size curve where the solution of TDS will land.

In order to empirically assess the quality of the solution of TDS, we tested the method of [34] [2] on several publicly available graph datasets [3]. The results are summarized in Table 1, from which it can be seen that in general, the solution computed by TDS is large in size with low density (see Table 1). These observations motivate the need to consider the additional flexibility afforded by TDkS in explicitly specifying the size parameter k, as TDS may not yield a satisfactory solution.

3.2 Hardness

Problem (3) corresponds to maximizing a discrete third-order polynomial subject to a cardinality constraint. This suggests that it is no easier to solve compared to the discrete

2. The code is available at https://github.com/tsourolampis/Scalable-Large-Near-Clique-Detection.

3. For details about the datasets, see Section VI.

Table 2: Statistics of the triangle-densest-subgraph computed using the method of [34] on representative datasets.

Graph	Size	Edge density	Triangle density
PPI-HUMAN	361	0.42	0.14
FACEBOOK-B	198	0.36	0.08
CAIDA	75	0.55	0.20
WEB-STANFORD	684	0.17	0.02
WEB-GOOGLE	66	0.85	0.64

quadratic maximization form associated with its classic edge-based counterpart, which is known to be NP–hard [15], and also very difficult to approximate. We make these notions concrete by proving the following pair of negative results regarding TDkS.

Theorem 1. *TDkS is NP–hard.*

Proof. Consider the decision version of the maximum clique problem, which is known to be NP–complete [35]. For a unweighted, undirected graph $\mathcal{G}$, the decision variant of the maximum clique problem asks whether $\mathcal{G}$ contains a complete subgraph on $\alpha \geq 3$ vertices. For such an instance, let $f_\alpha(\mathbf{x}^*)$ denote the optimal value of problem (3) with $k = \alpha$. Note that $f_\alpha(\mathbf{x}^*)$ cannot exceed $\binom{\alpha}{3}$, with equality attained if and only if $\mathcal{G}$ contains a clique on α vertices. Hence, solving problem (1) is at least as hard as solving an arbitrary decision instance of the maximum clique problem. □

In light of the above result, it is unlikely that the problem admits an efficient solution in polynomial time. Consequently, we focus on developing effective approximation algorithms for problem (3) that run in polynomial time. However, we first show that TDkS is fundamentally not amenable to favorable approximation in the worst-case sense; in fact it is more difficult to approximate compared to DkS.

More precisely, given an instance of DkS, let $\mathcal{M}^*$ denote the optimal solution and $\rho_2(\mathcal{M}^*, k)$ denote the optimal edge-density. Regarding the hardness of approximation of DkS, the following result is known [18].

Fact 2. *Assuming that the Exponential Time Hypothesis (ETH) is valid, there is no polynomial-time algorithm that can approximate the optimal value of DkS better than a multiplicative factor $\alpha(n) := n^{1/(\log\log n)^c}$, where $c > 0$ is a universal constant.*

Note that the quantity $\alpha(n) > 1$. Hence, given an arbitrary instance of DkS, there is no polynomial-time algorithm which can output a size-k subgraph whose edge-density is guaranteed to be no worse than a fraction $1/(\alpha(n))^{1-\epsilon}$ of the optimal edge-density $\rho_2(\mathcal{M}^*, k)$, for any $\epsilon > 0$. In other words, if $\mathcal{M}$ denotes the output of *any* polynomial-time approximation algorithm applied on a fixed instance of DkS and $\rho_2(\mathcal{M}, k)$ is the achieved edge density, it must hold that

$$\rho_2(\mathcal{M}^*, k) \geq \rho_2(\mathcal{M}, k) \geq O\left(\frac{1}{\alpha(n)}\right)\rho_2(\mathcal{M}^*, k). \tag{9}$$

We now demonstrate that the above hardness result for DkS can be utilized to derive an analogous hardness of approximation result for TDkS as well.

Theorem 2. *Assuming ETH is true, there is no polynomial-time algorithm that can approximate the optimal value of TDkS better than a multiplicative factor $\beta(n) := (\alpha(n))^{3/2}$.*

Proof. See Appendix B in the supplement. □

The above result implies that TDkS is more difficult to approximate compared to DkS, which is already known to be a challenging problem. Roughly speaking, Theorem 2 asserts that even the best possible polynomial-time approximation algorithm for TDkS must exhibit an approximation gap that grows as a sub-polynomial in the size of the problem input n, which is a very pessimistic result.

That being said, the results of Theorem 1 and 2 are based on viewing the problem from the perspective of the worst-case scenario, which may not always arise in practice. With this in mind, we propose a convex relaxation for TDkS with the aim of obtaining high-quality, sub-optimal solutions on real-world instances.

4 THE LOVÁSZ RELAXATION

In order to explain our approach, we first reformulate TDkS in combinatorial form as follows. Let $\mathcal{C} := \{\mathcal{S} \subset \mathcal{V} : |\mathcal{S}| = k\}$ denote the collection of subsets of vertices of size k. Note that there is a one-to-one correspondence between the elements of $\mathcal{X}_k$ and $\mathcal{C}$; every vector $\mathbf{x} \in \mathcal{X}_k$ is precisely the indicator function of a subset of vertices $\mathcal{S} \in \mathcal{C}$, i.e., given a vector $\mathbf{x} \in \mathcal{X}_k$ and a set $\mathcal{S} \in \mathcal{C}$, we have the equivalence

$$x_u = \begin{cases} 1 & \Leftrightarrow u \in \mathcal{S} \\ 0 & \Leftrightarrow u \notin \mathcal{S}. \end{cases} \tag{10}$$

This observation allows us to equivalently express problem (3) in minimization form as

$$\min_{\mathcal{S} \in \mathcal{C}} \left\{ F(\mathcal{S}) := \sum_{(u,v,w) \in \Delta} F_{uvw}(\mathcal{S}) \right\}, \tag{11}$$

where for each triangle $(u, v, w) \in \Delta$, we have defined the function

$$F_{uvw}(\mathcal{S}) := F(\mathcal{S} \cap \{u, v, w\}) = \begin{cases} -w_t, & \text{if } (u, v, w) \in \mathcal{S}, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Hence, the cost function $F(\mathcal{S})$ linearly decomposes over the set of triangles of $\mathcal{G}$, with each component function $F_{uvw}(\mathcal{S})$ contributing to the overall cost if and only if all three vertices constituting a triangle are included in the subgraph induced by $\mathcal{S} \in \mathcal{C}$. Our starting point is the following observation regarding the cost function $F(\mathcal{S})$.

Theorem 3. *$F(S)$ is a submodular function.*

Proof. See Appendix C in the supplement. □

Note that Theorem 3 does not change the fact that problem (3) is difficult to solve in the worst-case. However, it does allow us to adopt the following relaxation strategy. Let $\mathcal{P}_k := \{\mathbf{x} \in [0, 1]^n ; \mathbf{1}^T\mathbf{x} = k\}$ denote the convex hull of the combinatorial sum-to-k constraints. The key idea underpinning our approach is the following. Since the cost function of

(11) is submodular, we can replace it by its Lovász extension to obtain the following equivalent problem

$$\begin{array}{ll} \min & f_L(\mathbf{x}) \\ \text{s.to} & \mathbf{x} \in \{0,1\}^n \cap \mathcal{P}_k. \end{array} \tag{13}$$

Note that the equivalence stems from the fact that the Lovász extension equals the value of $F(.)$ at all binary $\{0,1\}^n$ vectors. Upon dropping the discrete constraints, we obtain the relaxed problem

$$\min_{\mathbf{x} \in \mathcal{P}_k} \; f_L(\mathbf{x}) \tag{14}$$

which corresponds to minimizing the Lovász extension of F over the convex hull of the combinatorial set $\mathcal{C}$. Our rationale for employing the Lovász extension as a convex surrogate of F stems from the fact that it corresponds to the convex closure of F on the domain $[0,1]^n$. In other words, in a certain sense, the Lovász extension is the tightest convex under-estimator of F.

It is evident that problem (14) is convex, and hence can be optimally solved in polynomial-time to obtain a lower bound on the optimal value of (11). However, from an algorithmic standpoint, a major issue in solving the above problem is that the Lovász extension of a submodular function does not admit an analytical functional form in general. This can be attributed to the fact that the base polytope $\mathcal{B}_F$ is characterized by (potentially) an exponential number of inequalities in the problem dimension n. In a seminal paper, Edmonds [36] established that a greedy algorithm based on sorting and querying F on n specific subsets suffices to compute a subgradient of the Lovász extension at any point $\mathbf{x} \in [0,1]^n$ without requiring explicit specification of the base polytope $\mathcal{B}_F$. While this result can be utilized within a projected subgradient framework for solving (14), for our present problem, we elect not to do so. This is due to the fact that the greedy algorithm is generic, i.e., it is not tailored to exploit the form of the submodular cost function of (11), which, in addition to its incremental nature, can result in a heavy computational footprint on large graphs.

We now demonstrate that it is possible to circumvent the aforementioned challenges related to solving (14) efficiently, and the main reason is that the Lovász extension for TDkS *does* admit an analytical form. In order to formally establish the result, we exploit the fact that F is linearly decomposable over the triangle-set Δ, which in turn implies that its base polytope can be expressed as the Minkowski sum of the base polytopes of the constituent functions F_{uvw} [37, Theorem 44.6], i.e., we have

$$\mathcal{B}_F = \sum_{\{u,v,w\} \in \Delta} \mathcal{B}_{F_{uvw}}, \tag{15}$$

where $\mathcal{B}_{F_{uvw}}$ is the base polytope associated with the component F_{uvw}, and we have overloaded notation to represent Minkowski sum [4] using the standard addition operator. Our next result shows that each such "sub"-polytope $\mathcal{B}_{F_{uvw}}$ admits a simple characterization.

Lemma 3. *The base polytope of F_{uvw} is given by*

$$\mathcal{B}_{F_{uvw}} = -w_t \mathit{conv}(\mathbf{e}_u, \mathbf{e}_v, \mathbf{e}_w). \tag{16}$$

Proof. See Appendix E in the supplement. □

Hence, the base polytope of F_{uvw} is the probability simplex in the space spanned by the coordinates indexed via (u,v,w) reflected about the origin and scaled by the weight w_t. Next, we exploit this result to derive an analytical form for the Lovász extension of F.

Theorem 4. *The Lovász extension of F is given by*

$$f_L(\mathbf{x}) = -\sum_{\{u,v,w\} \in \Delta} w_t \min\{x_u, x_v, x_w\}$$

Proof. See Appendix F in the supplement. □

The above result allows us to explicitly express problem (14) as

$$\min_{\substack{\mathbf{x} \in [0,1]^n, \\ \sum_{u=1}^n x_u = k}} \sum_{\{u,v,w\} \in \Delta} w_t \max\{-x_u, -x_v, -x_w\}, \tag{17}$$

which we designate as the Lovász relaxation. On inspecting the problem, however, it offers little in terms of an intuitive explanation as to why it can serve as a useful approximation for TDkS. To this end, our next result shows that the Lovász extension can be cast in an alternate form, which provides additional insight regarding (17).

Theorem 5. *The Lovász extension of F can be expressed as*

$$f_L(\mathbf{x}) = -\mathbf{t}^T\mathbf{x} + \sum_{\{u,v,w\} \in \Delta} w_t \cdot \phi_t(x_u, x_v, x_w) \tag{18}$$

where $\phi_t(x_u, x_v, x_w) := \max\{x_u + x_v - 2x_w, x_v + x_w - 2x_u, x_u + x_w - 2x_v\}$.

Proof. See Appendix G in the supplement. □

In the above expression, $\mathbf{t}$ is the $n \times 1$ vector whose i^{th} entry denotes the number of triangles that vertex i participates in. Using the above derived form, we now provide an intuitive explanation for the Lovász relaxation. Given any subset of vertices $\mathcal{S} \subseteq \mathcal{V}$, define the triangle "volume" $\text{vol}_\Delta(\mathcal{S}) := \sum_{v \in \mathcal{S}} t_v$ of $\mathcal{S}$ to be the sum of the weighted triangle counts of the vertices that constitute $\mathcal{S}$. Using a double counting argument, the triangle volume of any subset $\mathcal{S} \subseteq \mathcal{V}$ can be equivalently expressed as

$$\text{vol}_\Delta(\mathcal{S}) = t_1(\mathcal{S}) + 2t_2(\mathcal{S}) + 3t_3(\mathcal{S}), \tag{19}$$

where $t_1(\mathcal{S}), t_2(\mathcal{S})$ and $t_3(\mathcal{S})$ denote the weighted sum of triangles with one, two and three endpoints in $\mathcal{S}$, respectively. The above identity can be re-written as

$$t_3(\mathcal{S}) = (\text{vol}_\Delta(\mathcal{S}) - [2t_2(\mathcal{S}) + t_1(\mathcal{S})])/3, \forall\, \mathcal{S} \subseteq \mathcal{V}. \tag{20}$$

Note that the term on the left hand side corresponds to the objective function of TDkS. Hence, among subgraphs of a given size, those containing a large number of induced triangles must exhibit a large triangle volume (the first term on the right hand side) while simultaneously having few triangles being cut as a result of crossing the boundary of $\mathcal{S}$ (measured by the sum of the two terms subtracted from the volume). To be precise, for any given subset, a severed triangle with two endpoints $\{u,v\} \in \mathcal{S}$ affects the triangle counts (t_u, t_v) of both respective vertices (and hence the -2 factor), whereas a cut triangle with a single endpoint $u \in \mathcal{S}$ affects the triangle count t_u of only that vertex (and

4. The Minkowski sum of two sets $\mathcal{A}$ and $\mathcal{B}$ is given by $\mathcal{A} + \mathcal{B} = \{\mathbf{a} + \mathbf{b} \mid \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}$.

hence the -1 factor). The above equation asserts that for subgraphs with high triangle density, these losses stemming from severed triangles should be small compared to the triangle volume.

In order to establish the link with the form of the Lovász extension established in Theorem 5, we re-write (20) as

$$-t_3(\mathcal{S}) = -(\mathrm{vol}_\Delta(\mathcal{S}) + [2t_2(\mathcal{S}) + t_1(\mathcal{S})])/3. \quad (21)$$

Since we have already established that $-t_3(\mathcal{S})$ is a submodular function and $\mathrm{vol}_\Delta(\mathcal{S})$ is a modular (and thus submodular) function, the remainder on the right hand side must also be submodular, as submodularity is preserved under addition. Furthermore, as the Lovász extension of the sum of submodular functions equals the sum of the Lovász extensions of the component functions, inspecting the result of Theorem 5 reveals that it corresponds to the sum of the Lovász extensions of the terms on the right hand side of (21). Hence, the extension preserves the first term, corresponding to the triangle volume, whereas it uses a convex surrogate for the second term to approximate the losses in the volume stemming from severed triangles. In particular, when solving the Lovász relaxation, each vertex is assigned a soft score that indicates how likely it is to belong to the triangle-densest-k subgraph. The formulation then assigns the highest emphasis on those vertices which have large triangle counts, but also exhibit small variation in scores across triangles.

5 Algorithm: Mirror Descent

In this section, we describe our algorithm for efficiently solving the Lovász relaxation (17), which is a convex problem. Since the Lovász extension is non-differentiable, this suggests employing a Euclidean projected subgradient algorithm for solving (17). The algorithm starts from an initial feasible point $\mathbf{x}^0 \in \mathcal{P}_k$ and then proceeds in the following iterative fashion

$$\mathbf{x}^{r+1} = \arg\min_{\mathbf{x}\in\mathcal{P}_k} \left\{ (\mathbf{g}^r)^T\mathbf{x} + \frac{1}{\beta^r}\|\mathbf{x}-\mathbf{x}^r\|_2^2 \right\}, \forall\, r \in \mathbb{N} \quad (22)$$

where $\mathbf{g}^r \in \partial f_L(\mathbf{x}^r)$ denotes a subgradient of the Lovász extension $f_L(\mathbf{x})$ at the current iterate $\mathbf{x} = \mathbf{x}^r$ and $\beta^r > 0$ is the learning rate. A standard result in convex optimization states that *if* the subgradients of f_L are bounded in the Euclidean sense; i.e., there exists a constant $G > 0$ such that

$$\|\mathbf{g}\|_2 \leq G, \forall\, \mathbf{g} \in \partial f_L(\mathbf{x}), \forall\, \mathbf{x} \in \mathcal{P}_k, \quad (23)$$

then using the learning rate schedule $\beta_r = O(1/(\sqrt{r}))$ is sufficient to guarantee convergence to the optimal cost of (17) at a sublinear-rate of $O(G/\sqrt{r})$ [38, Theorem 3.2]. From this result, one can hope that the iteration complexity of the Euclidean subgradient algorithm is independent of the problem dimension n, which is a desirable trait for scaling up to large problem instances. However, the above claim is true provided that the Lipschitz constant G of the Lovász extension is independent of n. Unfortunately, this is not the case for our problem, as we now demonstrate.

Since f_L is linearly separable over the set of triangles in the graph, a standard result in convex analysis [39] asserts that a subgradient of the Lovász extension $\mathbf{g} \in \partial f_L(\mathbf{x})$ at a feasible point $\mathbf{x} \in \mathcal{P}_k$ can be expressed as

$$\mathbf{g} = \sum_{\{u,v,w\}\in\Delta} \mathbf{g}_{uvw}, \quad (24)$$

where $\mathbf{g}_{uvw} \in \partial f_{uvw}(\mathbf{x})$ denotes a subgradient of f_{uvw} at the point $\mathbf{x} \in \mathcal{P}_k$, and we have overloaded notation to represent set addition using the standard addition operator. The subdifferential set of each component function f_{uvw} is characterized by the equation

$$\partial f_{uvw}(\mathbf{x}) \in \arg\max_{\mathbf{y}\in\mathcal{B}_{F_{uvw}}} \mathbf{y}^T\mathbf{x}. \quad (25)$$

From the form of the base polytope $\mathcal{B}_{F_{uvw}}$ given by Lemma 3, it follows that any subgradient $\mathbf{g}_{uvw} \in \partial f_{uvw}(\mathbf{x})$ obeys

$$\|\mathbf{g}_{uvw}\|_\infty \leq w_t, \forall\, \{u,v,w\} \in \Delta. \quad (26)$$

Combining this observation with equation (24), we obtain that a subgradient of the Lovász extension $\mathbf{g} \in \partial f_L(\mathbf{x})$ is bounded in the ℓ_∞ sense as

$$\|\mathbf{g}\|_\infty \leq \sum_{\{u,v,w\}\in\Delta} \|\mathbf{g}_{uvw}\|_\infty \leq \sum_{\{u,v,w\}\in\Delta} w_t. \quad (27)$$

Since the sum of all triangle weights is a constant and $\|\mathbf{g}\|_2 \leq \sqrt{n}\|\mathbf{g}\|_\infty$, this implies that the Lipschitz constant G of f_L as defined in (23) in terms of ℓ_2-distances is $O(\sqrt{n})$, which is dependent on the dimension. Consequently, the Euclidean subgradient method (22) applied to solve (17) attains a dimension-*dependent* convergence rate of $O(\sqrt{\frac{n}{r}})$, which has undesirable implications for large-scale instances.

Thus, the non-Euclidean geometry of the problem renders the standard subgradient method (which measures distances in the ℓ_2-sense) a poor fit. In order to correct for this "mismatch" in geometry, we propose to employ the Mirror Descent algorithm (MDA) [40], which can be viewed as a generalization of the subgradient algorithm to non-Euclidean spaces. To be specific, MDA is an iterative first-order algorithm that starts from a point $\mathbf{x}^0 \in \mathcal{P}_k$ and performs the following updates

$$\mathbf{x}^{r+1} = \arg\min_{\mathbf{x}\in\mathcal{P}_k} \left\{ (\mathbf{g}^r)^T\mathbf{x} + \frac{1}{\beta^r}D(\mathbf{x},\mathbf{x}^r) \right\}, \forall\, r \in \mathbb{N} \quad (28)$$

where $D(.,.)$ is an appropriate "proximity"-measuring function. For example, on choosing $D(\mathbf{x},\mathbf{x}^r) = \|\mathbf{x}-\mathbf{x}^r\|_2^2$, we obtain the standard subgradient algorithm. This proximal term can be viewed as the Bregman divergence associated with the function $\|\mathbf{x}\|_2^2$, which is strongly convex w.r.t. the ℓ_2 norm.

Since the subgradients of the Lovász extension have constant size when measured using the ℓ_∞ norm, this motivates measuring distances using the ℓ_1 norm (which is the dual norm of the ℓ_∞ norm). This observation also suggests the choice of $D(.,.)$ in MDA to be the un-normalized Kullback-Leibler (KL) divergence between the points $\mathbf{x}$ and $\mathbf{x}^r$, which is defined as $D_{\mathrm{KL}}(\mathbf{x},\mathbf{x}^r) = \sum_{i=1}^n x_i\left(\log\frac{x_i}{x_i^r} - 1\right) + x_i^r$. Such a choice is based on the fact that the KL divergence is the Bregman divergence associated with the negative entropy function, which is strongly convex w.r.t. ℓ_1 norm on the feasible set $\mathcal{P}_k$. On performing the MDA udpates (28) using KL divergence with the learning rate schedule

$\beta_r = O(\sqrt{\log n/r})$, invoking a standard result in convex optimization [38, Theorem 4.2] guarantees a convergence rate of $O(G_\infty\sqrt{\log n/r})$, where G_∞ denotes the Lipschitz constant of f_L w.r.t. the ℓ_1 norm. From (27), since this quantity is a constant, we obtain a convergence rate that exhibits a significantly improved dependence on the problem dimension n compared to that of the standard subgradient algorithm. Hence, fixing the geometry mismatch by employing the ℓ_1 norm to measure distances in MDA pays substantial dividends in this case.

With the above choice of KL divergence, the MDA updates (28) can be equivalently expressed as

$$\mathbf{x}^{r+1} = \arg\min_{\mathbf{x}\in\mathcal{P}_k}\Big\{D_{\mathrm{KL}}(\mathbf{x}, \mathbf{x}^r \circledast \exp(-\beta^r\mathbf{g}^r))\Big\}, \tag{29}$$

where the $\circledast$ operator denotes element-wise multiplication. The update reveals that the algorithm utilizes a subgradient of the Lovász extension f_L to perform a multiplicative update on the present iterate $\mathbf{x}^r$ followed by computing the KL projection of the result onto $\mathcal{P}_k$ in order to ensure iterate feasibility. Hence, the potentially intensive tasks that have to be performed at each step are: (a) computing a subgradient of f_L, and (b) computing the KL projection onto the feasible set $\mathcal{P}_k$. As it turns out, these operations can be computed efficiently - with task (a) requiring $O(|\Delta|)$ time and task (b) being solvable via bisection search.

(A) Computing a subgradient: In order to compute a subgradient of f_L at a given point $\mathbf{x}\in\mathcal{P}_k$, it suffices to compute a subgradient of each component function f_{uvw}, and then sum the results up. From equation (25), a subgradient $\mathbf{g}_{uvw}\in\partial f_{uvw}(\mathbf{x})$ is given by

$$\mathbf{g}_{uvw} = -w_t\mathbf{e}_{s^*}, s^* \in \arg\min_{s\in\{u,v,w\}}\{x_s\}. \tag{30}$$

Then, the full subgradient $\mathbf{g} \in \partial f_L(\mathbf{x})$ is $\mathbf{g} = \sum_{\{u,v,w\}\in\Delta}\mathbf{g}_{uvw}$. We implement this procedure as follows: we visit each component function f_{uvw} and extract the index of the sub-vector $[x_u, x_v, x_w]^T$ that attains the minimum (with ties broken arbitrarily). This operation incurs $O(|\Delta|)$ time, and can be trivially parallelized. For a vertex $u\in\mathcal{V}$, let Δ_u denote the sub-collection of component functions f_{uvw} (and corresponding triangles) and is the index that attains the minimum in (30). Since each subgradient $\mathbf{g}_{uvw}$ corresponds to updating a single coordinate of the full subgradient $\mathbf{g}$, its entries are given by $g_u = -\sum_{t\in\Delta_u} w_t, \forall\ u\in\mathcal{V}$. In the case where the graph is unweighted, i.e., $w_t = 1, \forall\ t\in\Delta$, the update can be further simplified to $g_u = -|\Delta_u|, \forall\ u\in\mathcal{V}$. Implementing this update requires counting the number of times x_u attains the minimum across all the component functions that it participates in (which correspond to the triangles that vertex $u\in\mathcal{V}$ belongs to).

(B) Computing KL projections: Let $\mathbf{y}^r := \mathbf{x}^r \circledast \exp(-\beta^r\mathbf{g}^r)$ denote the vector obtained by performing the multiplicative subgradient update. The MDA update (29) can then be expressed as

$$\mathbf{x}^{r+1} = \arg\min_{\mathbf{x}\in\mathcal{P}_k}\Big\{D_{\mathrm{KL}}(\mathbf{x}, \mathbf{y}^r)\Big\}. \tag{31}$$

Our next result provides an explicit characterization of the optimal solution of the above problem.

Algorithm 1 MIRROR DESCENT

Input: Triangle list Δ, triangle weights $\{w_t\}_{t\in\Delta}$, subgraph size k, bisection tolerance $\epsilon > 0$.
Initialize: $\mathbf{x}^1 = (k/n)\mathbf{1}, r = 1$.

```
1: while Convergence criterion is not met do
2:     Obtain g^r ∈ ∂f_L(x^r).
3:     Update step-size β^r = c/√r.
4:     y^r := x^r ⊛ exp(−β^r g^r).
5:     x^{r+1} = BISECTION(y^r, k, ε).
6:     Update r = r + 1.
7: end while
8: return x_L = (1/r) Σ_{i=1}^r x^i
```

Lemma 4. *The solution $\mathbf{x}^{r+1}$ satisfies the conditions*

$$x_i^{r+1} = \min\Big\{1, \exp(-\nu^*)y_i^r\Big\}, \forall\ i\in[n], \sum_{i=1}^n x_i^{r+1} = k,$$

where $\nu^*\in\mathbb{R}$ is the optimal dual variable associated with the sum-to-k constraint.

Proof. See Appendix H in the supplement. □

The above result can be exploited to solve (31) via a simple procedure. Define the positive variable $\alpha^* := \exp(-\nu^*)$. Note that in order to solve for $\mathbf{x}^{r+1}$, it suffices to solve for α^*. This in turn, can be accomplished by finding the root of the non-linear, continuous equation

$$\phi(\alpha) := \left[\sum_{i=1}^n \min\{1, \alpha y_i^r\}\right] - k. \tag{32}$$

To this end, note that $\phi(\alpha)$ is a continuous function that is monotone non-decreasing in α, which suggests a simple bisection search procedure. The lower and upper limits of the initial bisection interval can be set to be $\alpha_l := 1/\min_{i\in[n]}\{y_i^r\}$ and $\alpha_u := 1/(n\max_{i\in[n]}\{y_i^r\})$ respectively, for which the value of the lower interval is $\phi(\alpha_l) \le 1-k < 0$ and that of the upper interval is $\phi(\alpha_u) = n-k > 0$. For a prescribed exit tolerance $\epsilon > 0$, the algorithm requires $O(\log[\phi(\alpha_l)-\phi(\alpha_u)/\epsilon])$ iterations to exit. Since each iteration of the bisection algorithm incurs $O(n)$ complexity, the overall complexity of bisection is $O(n\cdot\log(n/\epsilon))$. Putting everything together, the final complexity of executing r iterations of MDA is $O((|\Delta| + n\cdot\log(n/\epsilon))\cdot\sqrt{\log n/r})$.

A full description of MDA is provided in Algorithm 1. Since the computed solution $\mathbf{x}_L$ is not guaranteed to be integral in general, we perform a simple post-processing rounding step in order to obtain a binary indicator vector corresponding to a candidate subgraph. This is accomplished by simply projecting $\mathbf{x}_L$ onto the discrete sum-to-k constraints, which is equivalent to identifying the support of the k-largest entries in $\mathbf{x}_L$, and can be performed in $O(n\log k)$ time using heaps.

6 EXPERIMENTS

In this section, we test the effectiveness of our proposed method in exploring the triangle density-size trade-off across a collection of real-world graphs. Our results indicate that contrary to the worst-case scenario, real-world instances of TDkS can be far from adversarial, with the Lovász relaxation being effective at identifying high-quality, sub-optimal solutions.

TABLE 1: Summary of graph statistics: the number of vertices (n), the number of edges (m), the number of triangles ($|\Delta|$), and network type.

Graph	n	m	$\|\Delta\|$	Network Type
PPI-HUMAN	21,557	342K	2.39M	Biological
FACEBOOK-B	63,731	817K	3.51M	Social
CAIDA	192K	609K	455K	Router
WEB-STANFORD	281K	2.31M	11.33M	Web graph
WEB-GOOGLE	875K	5.10M	13.39M	Web graph
WIKI-TOPCATS	1.8M	28.51M	52.11M	Hyperlinks

6.1 Baselines

To the best of our knowledge, we are unaware of any pre-existing algorithms for the TDkS problem. Hence, we employ two state-of-the-art baselines for the (edge) densest-k-subgraph DkS problem, and test their efficacy at discovering triangle-dense subgraphs. These methods are described in brief below.

Lovász Relaxation for DkS [20]: The same approach considered herein, but applied to the edge-density based formulation, i.e., minimizing the Lovász extension for induced edges over the convex hull of the sum-to-k constraints. Utilizes a variant of the Alternating Direction Method of Multipliers (ADMM) [41] to solve the relaxed problem. As the solution is not guaranteed to be integral, a rounding post-processing step is used to obtain the candidate subgraph.

Low-rank Binary Matrix Principal Component [19]: Employs a low-rank decomposition of the graph adjacency matrix $\mathbf{A}$, followed by solving the DkS problem with the low-rank approximation in place of $\mathbf{A}$. For the rank-1 approximation scenario, the resulting problem admits a simple solution in $O(n)$ time, whereas for constant ranks (i.e., $r = O(1)$), the problem can be surprisingly solved in polynomial-time $O(n^{r+1})$. Furthermore, the resulting solution can be utilized to construct an instance-specific upper bound on the optimal edge density for a given subgraph size, which, while not attainable in general, can serve as a useful performance benchmark. In practice, the algorithm is run using ranks $r \leq 5$, owing to its high complexity. In our experiments, we ran the algorithm with rank-1 approximation for all our datasets to generate a candidate subgraph and the edge-density upper bound, as even the rank-2 case proved too expensive to compute.

Triangle density upper bound: We demonstrate that the edge density upper bound for DkS obtained via the above approach can also be converted into an upper bound on the optimal triangle density for TDkS via the Kruskal-Katona theorem, which asserts that for any unweighted, undirected (sub)graph $\mathcal{G}$, its edge density $\rho_2(\mathcal{G})$ and triangle density $\rho_3(\mathcal{G})$ must obey the relationship $\rho_3(\mathcal{G}) \leq (\rho_2(\mathcal{G}))^{3/2}$. Maximizing both sides of the above inequality w.r.t. all subgraphs of a fixed size k then yields the following relationship between the optimal triangle density of TDkS $\rho_3^*(\mathcal{G}, k)$ and the optimal edge density of DkS $\rho_2^*(\mathcal{G}, k)$: we must have $\rho_3^*(\mathcal{G}, k) \leq (\rho_2^*(\mathcal{G}, k))^{3/2}$. Hence, an upper bound on $\rho_2^*(\mathcal{G}, k)$ translates into an upper bound on $\rho_3^*(\mathcal{G}, k)$ as well. However, such a bound is not attainable in general for every choice of k as it is more loose compared to the bound on $\rho_2^*(\mathcal{G}, k)$. In spite of this, we observed that on real-world graphs the Lovász relaxation for TDkS can attain this upper bound, or capture a significant fraction of it.

Since the first two baselines do not aim to directly detect triangle dense subgraphs, for fair comparison, we also compare the efficacy of our proposed methods for TDkS at detecting edge-dense subgraphs against the above baselines.

6.2 Datasets, pre-processing and implementation

We used a collection of graph datasets (summarized in Table 1) from standard repositories [42], [43] to test the performance of all methods. Each dataset is unweighted, and pre-processed by symmetrizing any directed arcs, removing self-loops, and extracting the largest connected component.

For TDkS, we used the well-known NODEITERATOR++ algorithm [44, Algorithm 2] to obtain a list of triangles in the graph, which incurs a run-time complexity of $O(m^{3/2})$. A beneficial byproduct of this step is that we can eliminate all vertices which do not participate in forming triangles from belonging to a triangle-dense graph. Since this step does not change the number of triangles in the graph, it does not affect the input to TDkS. This, in turn, reduces the problem dimension n and results in a substantial improvement in the practical run-time of the Mirror Descent Algorithm. Note that this step, however, does change the edge set of the graph, and hence, the input to DkS. Consequently, the edge-based baselines outlined in the previous section are applied on the full graph.

All our experiments were performed in Matlab on a Windows workstation equipped with 16GB RAM and an Intel i7 processor. The Matlab code for the low-rank principal component approximation approach [19] was obtained via personal communication with the respective authors. Regarding our Mirror Descent algorithm for solving the Lovász relaxation for TDkS, we employed a diminishing step-size schedule $(c/t_{\max})/\sqrt{\log n/r}$, where $t_{\max}$ denotes the largest triangle "degree" in the graph, and $c > 0$ is a constant that was empirically chosen for each dataset. The maximum number of iterations applied was no more than 500 across all datasets.

6.3 Results and Discussion

The outcomes of our experiments on the considered datasets are depicted in Figure 1. Our main findings are:

• With regard to subgraph triangle density (left column in Figure 1), solving the Lovász relaxation for TDkS via Mirror Descent followed by rounding consistently yields the best results across all considered graphs. In fact, for small subgraph sizes ($\leq$ 100), it is the only method that attains, or comes close to attaining the upper bound on the optimal triangle density. Our results demonstrate that although TDkS is NP–hard and difficult to approximate in the worst-case, the Lovász relaxation can still prove to be an effective tool for detecting triangle-dense subgraphs in real-world graphs.

• Although the Mirror Descent algorithm aims to detect subgraphs with high triangle density, it turns in a commendable performance in terms of edge density as well (right column in Figure 1). In fact, for subgraph sizes ≤ 200, it outperforms the dedicated edge-based formulations, often by a significant margin and comes closest to attaining the edge density upper bound. This can be viewed as a consequence of

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3444608

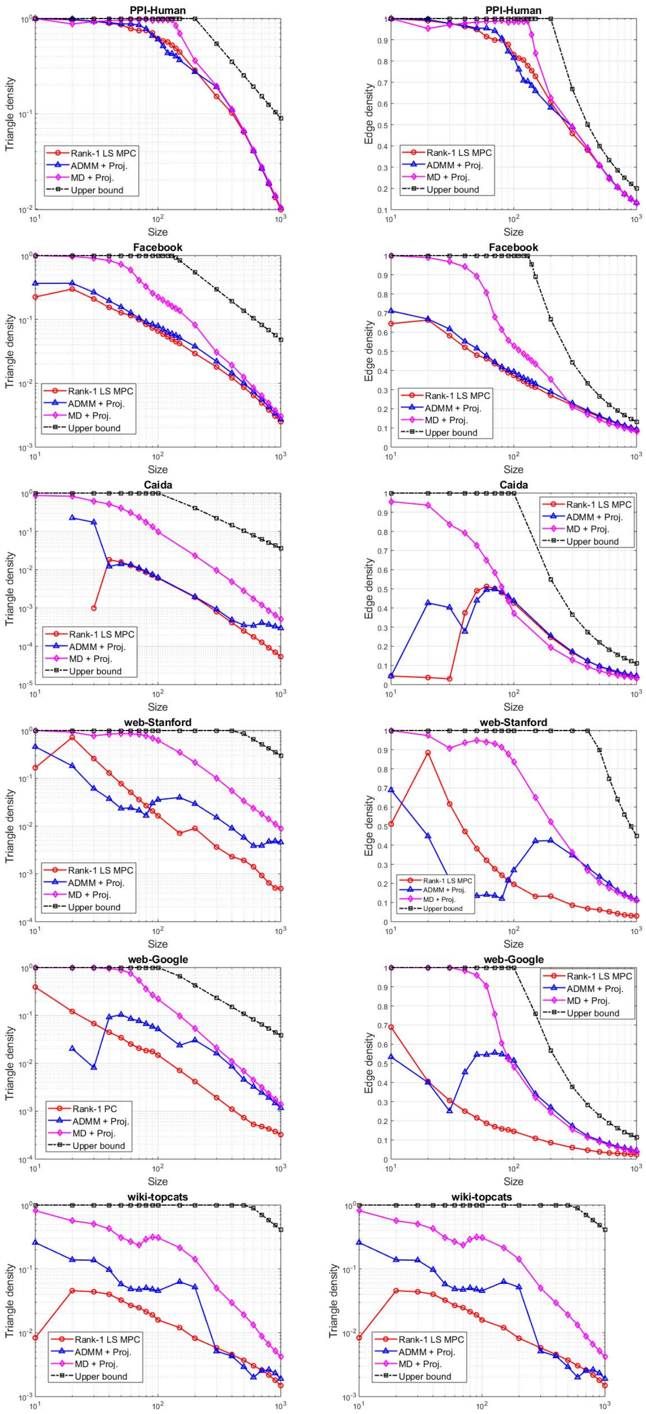

Fig. 1: **Left column:** Triangle density (on a log-scale) vs size. **Right column:** Edge density vs size. Red (Rank-1 least-squares matrix principal component) and blue (ADMM) curves are methods for DkS while the magenta curve (MD) is for TDkS. The black curve in the right column depicts the edge density upper bound for DkS. This bound combined with the Kruskal-Katona theorem also yields an upper bound on TDkS (black curve in the left column).

the Kruskal-Katona theorem which formalizes the following intuitive notion: if a subgraph has high triangle density, then it must possess high edge density as well. Looking at Figure 1 (right column) confirms this observation.

• For large subgraph sizes, the edge density obtained by Mirror Descent / TDkS is often second (although by a small margin) to that obtained by applying the Lovász relaxation for DkS. Empirically, we note that this occurs (i.e., the blue curve "overtakes" the magenta curve) when the edge density falls below the 50% threshold. A possible explanation is as follows. Turán's theorem [45] implies that a graph can exhibit an edge density at most 0.5 without harboring any triangles. In other words, below this threshold, there do exist graphs with edge density up to 50% while containing very few triangles. Consequently, in the regime where the densest subgraph of a given size has edge density upper bounded by 0.5, employing a density measure based on edges may prove to be more beneficial as opposed to using triangles, if one cares more about edge density.

• In terms of timing, the approach of [19] is the fastest as it simply requires computing the principal component of the adjacency matrix. In contrast, the Lovász relaxation schemes for both DkS and TDkS (blue and magenta curves respectively) have to solved using iterative methods and thus they consume more time. The complete results are provided in Appendix I.

7 Case Study – Single Document Keyword Extraction

Overview: We consider an application of TDkS to the real-world problem of unsupervised single document keyword extraction. For this purpose, we adopt the popular Graph-of-Words (GoW) model of [46]. In this model, a given text document is represented as an undirected graph, where vertices correspond to unique words occurring in the document and an edge connects a pair of vertices if the words they represent co-occur within a window of pre-specified length L which is slid from the start to the end of the document, spanning across sentences. For example, the choice $L = 2$ corresponds to connecting pairs of words appearing in bi-grams, whereas $L = 3$ connects triplets of words forming tri-grams via triangles. In this context, extracting a dense subgraph from a GoW-representation of a text document corresponds to determining a highly cohesive subset of keywords, which can form an informative summary of a document's content.

Prior Work: Such an approach has been adopted previously in [5] to extract summaries of trending news stories from streams of Twitter data, based on a GoW model with window length $L = 2$. Later work [6] considered GoW representations of general window length $L \geq 2$, and proposed the following two-step approach. First, a k-core decomposition [47] is performed on the graph followed by a k-truss [5] decomposition [48] refinement step in order to extract informative keywords. We point out that such density based approaches to keyword extraction, which quantify importance of keywords based on how frequently they

5. A k-truss is a subgraph where every edge participates in at least $k-2$ triangles.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3444608

co-occur with other groups of words (i.e., cohesiveness), are different from random-walk based approaches such as TextRank [46], which assign importance to words based on eigen-vector centrality. It is known that the former approach based on density generally outperforms the latter on real-world data [6], [49].

Limitations: That being said, the majority of density based approaches to keyword extraction still suffer from two main limitations: (i) they do not feature a natural means of explicitly controlling the size of the extracted summary, and (ii), few methods can effectively deal with higher-order word co-occurrences (corresponding to GoW models with $L \geq 3$). This is important as they model tighter notions of cohesiveness amongst words in contrast to pair-wise co-occurrences. Hence, using higher-order word co-occurrences for keyword-extraction has the potential to yield more cohesive text summaries compared to using their pair-wise counterpart. An exception to the second limitation is the method of [6], which employs k-truss decomposition as a proxy for detecting a triangle-dense subgraph. However, the authors of [6] noted that their approach was effective in extracting informative summaries from real text data only for GoW models constructed from higher-order n-grams (with window length L at least $4, 5$). This observation was attributed to the fact that the number of triangles in the constructed graph increases with L, which in turn facilitates the detection of a dense k-truss. However, this method cannot effectively extract cohesive summaries from tri-grams.

We argue that TDkS is well positioned to address these shortcomings. Since our problem allows pre-specification of the desired subgraph size, given a GoW representation of a document, it enables direct control of the size of the extracted summary. Additionally, the objective function of TDkS is directly geared towards maximizing the weighted sum of induced triangles in the subgraph, which we intuitively expect will allow our formulation to extract meaningful summaries from simple GoW representations constructed using tri-grams alone (with window length $L = 3$).

Experiments: In order to provide empirical corroboration of our hypothesis, we applied TDkS to the problem of generating high-level descriptions of recently released Hollywood movies based on publicly available text reviews submitted by movie critics. We used the popular review aggregator website https://www.metacritic.com/ to obtain reviews for the following movies, which are briefly described below.

- **John Wick 3: Parabellum** [6] - an action movie in the John Wick franchise directed by Chad Stahelski and starring Keanu Reeves as the titular character.
- **Baby Driver** [7] - a heist movie directed by Edgar Wright.
- **Arrival** [8] - a science fiction movie that doubles as a drama; directed by Denis Villeneuve and starring Amy Adams in the lead role.
- **Hereditary** [9] - a horror movie centered around the evil that befalls a family; written and directed by Ari Aster (in his directorial debut) and starring Toni Collette in the lead role.
- **Mad Max: Fury Road** [10] - an action movie set in a wasteland that serves as a soft reboot of the Mad Max franchise; directed by George Miller.
- **Joker** [11] - an origin story about a popular comic book villain, The Joker; directed by Todd Phillips and starring Joaquin Phoenix in the lead role.

For each movie, we collected all the available reviews submitted by movie critics into a single text document. On average, a movie had 45 reviews, each of which represents a summary/opinion of a critic. We performed a simple pre-processing step on each text document where we filtered out short, commonly ocurring words using the list of stopwords provided in Python's Natural Language Processing Toolkit (NLTK) [12]. The resulting text document comprised, on average, 740 unique words used to describe a movie by multiple critics. Then, from each document, we enumerated all tri-grams and bi-grams in order to construct two different GoW representations. In the first model, triplets of vertices are connected by a triangle if their corresponding words co-occur together in a tri-gram; each triangle has a weight denoting the co-occurrence probability of the word triplet in the text document. We apply the Mirror Descent algorithm developed for the Lovász relaxation of TDkS in order to detect a triangle-dense subgraph of a pre-specified size k. Note that the vertices comprising such a subgraph correspond to words which frequently co-occur together in sub-sentences of length 3. As a baseline, we construct a second GoW model, where a pair of vertices are connected by an edge if their respective words co-occur in a bi-gram, with each edge being weighted by the pair-wise co-occurrence probability of the word pair. We then apply the ADMM algorithm developed in [20] for the Lovász relaxation of DkS to extract a dense subgraph of size-k based on pair-wise cohesiveness. For a fixed summary size k, we expect the summary generated using TDkS to be more cohesive and informative as compared to DkS, since the former approach exploits higher-order co-ocurrences in text.

Results and Discussion: The outcomes of our experiments are depicted in Tables 4 and 5, which display the 30-word summaries generated for each movie using TDkS and DkS respectively. At a high level, it is evident that the movie summaries generated from the tri-gram based GoW model using TDkS are substantially more cohesive and descriptive compared to that obtained from the bi-gram GoW model using DkS. A minor drawback, however, is that the depth of coverage obtained from the tri-gram model sometimes comes at the expense of breadth. For example, from Table 5 it can be seen that the summary of each movie contains the name of the director, but this is true for 4/6 movies in Table 4. The similarities and differences in the two kinds of summaries for each movie are elucidated below.

- **John Wick 3: Parabellum** - both summaries extract the name of the movie and the fact that it belongs

6. https://www.metacritic.com/movie/john-wick-chapter-3---parabellum
7. https://www.metacritic.com/movie/baby-driver
8. https://www.metacritic.com/movie/arrival
9. https://www.metacritic.com/movie/hereditary
10. https://www.metacritic.com/movie/mad-max-fury-road
11. https://www.metacritic.com/movie/joker
12. https://www.nltk.org/

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3444608

in a franchise. However, the TDkS summary additionally identifies that the movie is the third in the franchise. While the DkS summary contains the dircetor, it does not contain the full name of the movie's main star Keanu Reeves, who is present in the TDkS summary. The DkS summary reveals the movie's content as being violent and also mentions the choreography, although few additional details are present. In contrast, the TDkS summary reveals that the critics found the level of violence excessive, but enjoyed the choreography of the fight scenes.

- **Baby Driver** - the twin summaries contain the movie's name and that of the director. The TDkS summary reveals that the critics enjoyed the story and pacing of the movie (**tale, perfect, enjoy, fun, fast, furious, inventive**), and also the fact that the movie plot simultaneously combines elements of action, romance and musicals (**heist, romance, love, musicals**). The soundtrack of the movie is also mentioned (**jukebox**). In contrast, the DkS summary is sparser in details regarding the plot, and beyond the action elements does not pick up on the other aspects of the movie's plot. The soundtrack is also mentioned (**music, beat**), and the fact the movie was released in the summer.
- **Arrival** - common to both summaries are the names of the movie, director and the lead actress. However, the TDkS summary identifies the genre of the film as being both science fiction *and* a drama – the DkS summary does not pick up the latter. This serves as key context for description of the movie's content which was praised by the critics for the weight of its intellectual themes, as well as the moving story. In contrast, the DkS summary, while picking up on some of the emotional themes in the story, misses out this key intellectual aspect.
- **Hereditary** - both summaries highlight the movie's name, the genre being horror, and the fact that the first-time director was also the writer. The TDkS summary additionally reveals that the critics praised the direction (**excels, smugly, promising**) and found the movie impressive, and even designated it as a masterpiece. They also commented on the fact that the movie is more in the vein of old school horror movies - we note that these details are missing from the DkS summary. However, the latter summary does mention the performance of the lead actress, Toni Collette.
- **Mad Max: Fury Road** - while having the movie's and director's name in common with the DkS summary, the triangle-based summary contains substantially more information about the content - the fact the reviewers praised the style and energy of the action sequences (**delivers, effective, loco**), and concurred that the movie marked a successful return to the Mad Max universe, and even surpassed previous installments.
- **Joker** - the TDkS summary reveals that the movie is a re-imagining of a comic book character (the DkS summary adds the detail that it is a DC comic), and is anchored by a powerful performance from the actor Joaquin Phoenix, which is described as being Oscar worthy (Joaquin Phoenix indeed won the Oscar for Best Male Actor for his portrayal of the Joker). The critics also described the movie as being depressing, but at the same time also offered praise (**engrossing, masterful, iconic, well, made, great, fine**). Praise was not uniform, however - downsides are mentioned, but more specific details are not captured. By comparison, the DkS summary is spartan in its descriptive quality - the main addition being the director's name.

TABLE 2: Top-30 words which most frequently co-occur in trigrams obtained from movie review text data; detected using the Lovasz relaxation for TDkS (with $k = 30$). Key terms are highlighted in bold.

John Wick 3	Baby Driver	Arrival	Hereditary	Mad Max: Fury Road	Joker
movies	**movie**	**movie**	**movie**	**films**	**movie**
john	**movies**	**film**	**film**	**mad**	**joker**
wick	**baby**	**arrival**	**hereditary**	**max**	**comic**
3	**driver**	makes	also	**fury**	**book**
parabellum	almost	**villeneuve**	**director**	**road**	**character**
still	feels	take	**writer**	**made**	**reimagined**
starring	**edgar**	almost	**ari**	**director**	movies
keanu	**wright**	**science**	**aster**	**george**	one
reeves	literalizes	**fiction**	**first**	**miller**	**anchored**
may	like	**sci**	**time**	also	**performance**
another	always	**fi**	**feature**	**action**	**joaquin**
one	watching	**drama**	**debut**	**sequences**	**phoenix**
chapter	**tale**	used	us	**scenes**	**oscar**
series	**perfect**	**adams**	buy	**delivers**	**worthy**
third	**enjoy**	head	gives	**effective**	allows
anything	**fun**	like	**excels**	**loco**	would
action	**fast**	**moving**	**smugly**	**style**	study
fight	**furious**	**thoughtful**	**promising**	**orgy**	**depressing**
scenes	**inventive**	**great**	**impressively**	stands	**engrossing**
gratuitously	nothing	**intimate**	**masterpiece**	**return**	**masterful**
violent	much	**poignant**	**horror**	**revived**	**iconic**
choreographed	going	**intellectual**	**agitating**	**universe**	**well**
gorgeously	**musical**	**intelligent**	**unnerving**	**wasteland**	**made**
grander	**romance**	**beguiling**	**old**	**surpassed**	**great**
baring	**heist**	**leaves**	**school**	**previous**	**fine**
fluid	**jukebox**	**questions**	makes	interested	almost
story	**love**	truly	making	indebted	created
closer	plays	best	**family**	**stone**	**downside**
draw	sorts	something	takes	**cold**	**killer**
parabolically	mashed	conundrum	much	**warrior**	work

TABLE 3: Top-30 words which most frequently co-occur in bigrams obtained from movie review text data; detected using the Lovasz relaxation for DkS (with $k = 30$). Key terms are highlighted in bold.

John Wick 3	Baby Driver	Arrival	Hereditary	Mad Max: Fury Road	Joker
film	**film**	**film**	**film**	**movie**	**film**
movies	**movie**	**movie**	**movie**	**film**	**movie**
movie	movies	**arrival**	**hereditary**	**mad**	**joker**
john	**baby**	much	see	**max**	**comic**
wick	**driver**	less	**director**	**fury**	**book**
3	**director**	**denis**	**writer**	**road**	**character**
parabellum	**edgar**	**villeneuve**	**ari**	movies	**dc**
though	**wright**	like	**aster**	films	much
get	something	**science**	**first**	**george**	work
first	also	**fiction**	**time**	**miller**	**performance**
like	much	**sci**	mind	never	**joaquin**
director	**story**	**fi**	**debut**	much	**phoenix**
chad	**fun**	one	**feature**	things	**arthur**
stahelski	nothing	**adams**	every	us	**villain**
reeves	even	**performance**	gets	**mayhem**	**todd**
chapter	**heist**	**moving**	**toni**	**action**	**phillips**
series	**action**	**emotional**	**collette**	**chase**	study
franchise	**crime**	**human**	**performance**	makes	one
action	work	**great**	takes	also	bad
still	us	**story**	even	get	social
violence	one	best	long	work	last
violent	high	life	genre	feels	well
good	filmmaking	truly	**horror**	first	point
far	**love**	made	**family**	even	also
point	part	open	**supernatural**	muscle	go
scenes	like	time	**atmosphere**	new	enough
two	**beat**	ideas	**dread**	like	anything
choreographed	**music**	first	much	post	specific
set	car	makes	new	story	like
new	**summer**	**questions**	one	one	movies

8 CONCLUSIONS

We considered the triangle-densest-k-subgraph problem (TDkS) which aims to compute the size k subgraph with the largest number of induced triangles. Unfortunately, not only is the problem NP–hard, but it is also difficult to approximate in polynomial-time, in the worst-case sense. With the aim of computing high-quality, sub-optimal solutions on real-world instances, we exploited the fact that the cost function of TDkS is submodular to construct a convex relaxation of the problem based on the Lovász extension of submodular functions. As we derived an analytical functional form for the extension, this enabled us to devise a Mirror Descent algorithm for efficiently solving the problem at scale. Our results on real-world graphs showcased that our approach can effectively exploit triangle motifs to attain state-of-the-art performance, and can provide a more effective means of exploring the density-size trade-off compared to baselines that only use edges for density maximization. Additionally, we utilized the problem of document summarization to showcase that TDkS can generate more informative word summaries compared to DkS.

9 Acknowledgements

Supported by the KU Leuven Special Research Fund (BOF/STG-22-040), the National Science Foundation, USA (IIS-1908070) and the Army Research Office, USA (ARO-W911NF1910407).

References

[1] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "Fraudar: Bounding graph fraud in the face of camouflage," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 895–904, ACM, 2016.

[2] S. Zhang, D. Zhou, M. Y. Yildirim, S. Alcorn, J. He, H. Davulcu, and H. Tong, "Hidden: hierarchical dense subgraph detection with application to financial fraud detection," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 570–578, SIAM, 2017.

[3] Y. Podolyan and G. Karypis, "Common pharmacophore identification using frequent clique detection algorithm," *Journal of chemical information and modeling*, vol. 49, no. 1, pp. 13–21, 2009.

[4] B. Saha, A. Hoch, S. Khuller, L. Raschid, and X.-N. Zhang, "Dense subgraphs with restrictions and applications to gene annotation graphs," in *Annual International Conference on Research in Computational Molecular Biology*, pp. 456–472, Springer, 2010.

[5] A. Angel, N. Sarkas, N. Koudas, and D. Srivastava, "Dense subgraph maintenance under streaming edge weight updates for real-time story identification," in *Proceedings of the 38th International Conference on Very Large Data Bases*, pp. 574–585, VLDB Endowment, 2012.

[6] A. Tixier, F. Malliaros, and M. Vazirgiannis, "A graph degeneracy-based approach to keyword extraction," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1860–1870, 2016.

[7] Z. Ertem, A. Veremyev, and S. Butenko, "Detecting large cohesive subgroups with high clustering coefficients in social networks," *Social Networks*, vol. 46, pp. 1–10, 2016.

[8] J. Cadena, F. Chen, and A. Vullikanti, "Graph anomaly detection based on steiner connectivity and density," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 829–845, 2018.

[9] Y. Xu, C. Ma, Y. Fang, and Z. Bao, "Efficient and effective algorithms for generalized densest subgraph discovery," *Proceedings of the ACM on Management of Data*, vol. 1, no. 2, pp. 1–27, 2023.

[10] T. Lanciano, A. Miyauchi, A. Fazzone, and F. Bonchi, "A survey on the densest subgraph problem and its variants," *ACM Computing Surveys*, vol. 56, no. 8, pp. 1–40, 2024.

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

[12] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pp. 84–95, Springer, 2000.

[13] C. Ma, Y. Fang, R. Cheng, L. V. Lakshmanan, W. Zhang, and X. Lin, "On directed densest subgraph discovery," *ACM Transactions on Database Systems (TODS)*, vol. 46, no. 4, pp. 1–45, 2021.

[14] 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*, pp. 104–112, 2013.

[15] U. Feige, D. Peleg, and G. Kortsarz, "The dense k-subgraph problem," *Algorithmica*, vol. 29, no. 3, pp. 410–421, 2001.

[16] S. Khot, "Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique," *SIAM Journal on Computing*, vol. 36, no. 4, pp. 1025–1071, 2006.

[17] A. Bhaskara, M. Charikar, V. Guruswami, A. Vijayaraghavan, and Y. Zhou, "Polynomial integrality gaps for strong sdp relaxations of densest k-subgraph," in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pp. 388–405, SIAM, 2012.

[18] P. Manurangsi, "Almost-polynomial ratio eth-hardness of approximating densest k-subgraph," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 954–961, 2017.

[19] D. Papailiopoulos, I. Mitliagkas, A. Dimakis, and C. Caramanis, "Finding dense subgraphs via low-rank bilinear optimization," in *International Conference on Machine Learning*, pp. 1890–1898, 2014.

[20] A. Konar and N. D. Sidiropoulos, "Exploring the subgraph density-size trade-off via the lovaśz extension," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 743–751, 2021.

[21] R. Andersen and K. Chellapilla, "Finding dense subgraphs with size bounds," in *International workshop on algorithms and models for the web-graph*, pp. 25–37, Springer, 2009.

[22] S. Khuller and B. Saha, "On finding dense subgraphs," in *International colloquium on automata, languages, and programming*, pp. 597–608, Springer, 2009.

[23] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, p. 440, 1998.

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

[25] C. E. Tsourakakis, J. Pachocki, and M. Mitzenmacher, "Scalable motif-aware graph clustering," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1451–1460, 2017.

[26] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[27] Y. Zhang and S. Parthasarathy, "Extracting analyzing and visualizing triangle k-core motifs within networks," in *2012 IEEE 28th international conference on data engineering*, pp. 1049–1060, IEEE, 2012.

[28] A. Konar and N. D. Sidiropoulos, "The triangle-densest-k-subgraph problem: Hardness, lovász extension, and application to document summarization," in *Proceedings of AAAI*, 2022.

[29] L. Lovász, "Submodular functions and convexity," in *Mathematical programming the state of the art*, pp. 235–257, Springer, 1983.

[30] F. Bach *et al.*, "Learning with submodular functions: A convex optimization perspective," *Foundations and Trends® in Machine Learning*, vol. 6, no. 2-3, pp. 145–373, 2013.

[31] S. Fujishige, *Submodular functions and optimization*. Elsevier, 2005.

[32] J. B. Kruskal, "The number of simplices in a complex," *Mathematical optimization techniques*, vol. 10, pp. 251–278, 1963.

[33] G. O. Katona, "A simple proof of the erdös-chao ko-rado theorem," *Journal of Combinatorial Theory, Series B*, vol. 13, no. 2, pp. 183–184, 1972.

[34] 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*, pp. 815–824, 2015.

[35] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*, pp. 85–103, Springer, 1972.

[36] J. Edmonds, "Submodular functions, matroids, and certain polyhedra," *Edited by G. Goos, J. Hartmanis, and J. van Leeuwen*, vol. 11, 1970.

[37] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*, vol. 24. Springer Science & Business Media, 2003.

[38] S. Bubeck, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[39] R. T. Rockafellar, *Convex analysis*. No. 28, Princeton university press, 1970.

[40] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, vol. 31, no. 3, pp. 167–175, 2003.

[41] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, 2013.

[42] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection." http://snap.stanford.edu/data, 2014.

[43] J. Kunegis, "KONECT – The Koblenz Network Collection," in *Proc. Int. Conf. on World Wide Web Companion*, pp. 1343–1350, 2013.

[44] S. Suri and S. Vassilvitskii, "Counting triangles and the curse of the last reducer," in *Proceedings of the 20th international conference on World wide web*, pp. 607–614, 2011.

[45] P. Turán, "On an external problem in graph theory," *Mat. Fiz. Lapok*, vol. 48, pp. 436–452, 1941.

[46] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, 2004.

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

[48] J. Cohen, "Trusses: Cohesive subgraphs for social network analysis," *National security agency technical report*, vol. 16, no. 3.1, 2008.

[49] F. Rousseau and M. Vazirgiannis, "Main core retention on graph-of-words for single-document keyword extraction," in *European Conference on Information Retrieval*, pp. 382–393, Springer, 2015.

Aritra Konar received the BTech degree in electronics and communications engineering from the West Bengal University of Technology, West Bengal, India, in 2011, and the MS and PhD degrees in electrical engineering from the University of Minnesota, Minneapolis, Minnesota, in 2014 and 2017, respectively. He is currently an assistant professor in the Department of Electrical Engineering at KU Leuven, Belgium. His research interests include network science, graph mining, nonlinear optimization, and signal processing.

Nicholas D. Sidiropoulos (Fellow, IEEE) received the Diploma in electrical engineering from the Aristotle University of Thessaloniki, Greece, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 1988, 1990, and 1992, respectively. He is the Louis T. Rader Professor at the University of Virginia. He has previously served on the faculty of the University of Minnesota, and the Technical University of Crete, Greece. His research interests are in signal processing, communications, optimization, and tensor decomposition with applications in machine learning and communications. He received the NSF/CAREER award in 1998, the IEEE Signal Processing Society (SPS) Best Paper Award in 2001, 2007, 2011, and 2023, and the IEEE SPS Donald G. Fink Overview Paper Award in 2023. He served as IEEE SPS Distinguished Lecturer (2008–2009), as Vice President - Membership of IEEE SPS (2017–2019), and as chair of the SPS IEEE Fellow Evaluation Committee (2020-2021). He received the IEEE Signal Processing Society Meritorious Service Award (2010), the Distinguished Alumni Award of the Department of ECE, University of Maryland (2013), the EURASIP Technical Achievement Award (2022), and the IEEE SPS Claude Shannon - Harry Nyquist Technical Achievement Award (2023). He is a Fellow of EURASIP (2014).

Supplement

Aritra Konar, and Nicholas D. Sidiropoulos, *Fellow, IEEE*

Appendix A: Proof of Theorem 2

Our starting point is the Kruskal-Katona Theorem [1], [2], which states that for any graph $\mathcal{G}$ of size k, its edge-density $\rho_2(\mathcal{G},k)$ and triangle-density $\rho_3(\mathcal{G},k)$ must obey the relationship

$$\rho_2(\mathcal{G},k) \geq \rho_3(\mathcal{G},k)^{2/3}. \tag{1}$$

As an immediate consequence of this result, we obtain the following relationship between the optimal values of TDkS and DkS, for a fixed input.

$$\rho_2(\mathcal{M}^*,k) \geq \rho_3(\mathcal{S}^*,k)^{2/3}, \tag{2}$$

where the size-k vertex subsets $\mathcal{M}^*$ and $\mathcal{S}^*$ represent optimal solutions of DkS and TDkS respectively. Combining the above result with the statement of Fact 2, we obtain the following relationship

$$\rho_2(\mathcal{M}^*,k) \geq \rho_2(\mathcal{M},k) \geq O\left(\frac{1}{\alpha(n)}\right)\rho_3(\mathcal{S}^*,k)^{2/3}, \tag{3}$$

which must hold for any joint instance of TDkS and DkS, and for any polynomial-time approximation algorithm for DkS, which produces a subgraph $\mathcal{M}$ of size k as output.

We are now ready to establish the proof of our main result, which relies on an argument based on contradiction. Assume that there exists a polynomial-time approximation algorithm $\mathcal{A}$ for TDkS which outputs a size-k subgraph $\mathcal{S}$ whose triangle density $\rho_3(\mathcal{S},k)$ is guaranteed to be no worse than a fraction $O(1/(\alpha(n))^{3/2-\epsilon})$ of the optimal triangle density $\rho_3(\mathcal{S}^*,k)$, for some $\epsilon > 0$. In other words, the triangle density achieved by the output of the algorithm $\mathcal{A}$ obeys the relationship

$$\rho_3(\mathcal{S}^*,k) \geq \rho_3(\mathcal{S},k) \geq \frac{C_1}{(\alpha(n))^{3/2-\epsilon}} \cdot \rho_3(\mathcal{S}^*,k) \tag{4}$$

for every instance of TDkS (here $C_1 > 0$ is a universal constant).

Let $\rho_2(\mathcal{S},k)$ denote the edge density of the subgraph induced by $\mathcal{S}$. Applying the Kruskal-Katona theorem and the second inequality in (4), we obtain the following lower bound on $\rho_2(\mathcal{S},k)$.

$$\rho_2(\mathcal{S},k) \geq \rho_3(\mathcal{S},k)^{2/3} \geq \frac{C_2}{(\alpha(n))^{1-\delta}} \cdot \rho_3(\mathcal{S}^*,k)^{2/3}, \tag{5}$$

where $C_2 := C_1^{2/3} > 0$ and $\delta := \frac{2\epsilon}{3} > 0$. Furthermore, we point out that $\rho_2(\mathcal{S},k)$ cannot exceed the optimal edge density $\rho_2(\mathcal{M}^*,k)$ for the same problem instance. Combining this fact with (5), we obtain

$$\rho_2(\mathcal{M}^*,k) \geq \rho_2(\mathcal{S},k) \geq \frac{C_2}{(\alpha(n))^{1-\delta}} \cdot \rho_3(\mathcal{S}^*,k)^{2/3}, \tag{6}$$

which is a clear contradiction of (3).

Appendix B: Proof of Theorem 3

Recall that $F(\mathcal{S})$ can be linearly decomposed over the set of triangles Δ as follows

$$F(\mathcal{S}) = \sum_{(u,v,w)\in\Delta} F_{uvw}(\mathcal{S}). \tag{7}$$

Since submodularity is preserved under addition and restriction, in order to establish the desired result, it suffices to show that each component function $F_{uvw}(\mathcal{S})$ is submodular. To this end, note that by construction, the domain of F_{uvw} is the power-set of the reduced ground set comprising vertices $\{u,v,w\}$. In order to demonstrate that F_{uvw} is submodular, we invoke the following fact about submodular functions - a function is submodular if and only if all its projections onto 2 variables are submodular [3]. Considering the discrete polynomial form of F_{uvw}

$$f_{uvw}(x_u,x_v,x_w) = -w_t x_u x_v x_w,$$

the above test for submodularity is tantamount to fixing any one of the variables and testing whether the function is submodular in the remaining pair of free variables, for all such possible configurations. Since the function is symmetric with respect to (w.r.t.) its arguments, it follows without loss of generality that we can fix x_u and check whether the projection of f_{uvw} onto (x_v,x_w) is submodular. This in turn simplifies to testing whether $g_t(x_v,x_w) := f(1,x_v,x_w) = -w_t x_v x_w$ is submodular (note that the case $x_u = 0$ is trivial). Recalling the definition of submodularity for functions of two variables, we note that the condition

$$g_t(1,0) + g_t(0,1) \geq g_t(0,0) + g_t(1,1)$$

is always satisfied since w_t is non-negative. Hence, each component function $F_{uvw}(\mathcal{S})$ is submodular, from which the claim follows.

Remark: We point out that when the graph is unweighted and each triangle has unit weight, the above theorem recovers the result of [4, Theorem 3] as a special case. Furthermore, as we now explain, the above line of reasoning can be extended to establish a considerably more general result regarding submodularity of the weighted sum of induced higher-order cliques in a subgraph. Consider an undirected graph on n vertices represented as $\mathcal{G} = (\mathcal{V},\mathcal{C}_k,w)$, where $\mathcal{V}$ denotes the vertex set, $\mathcal{C}_k$ denotes the set of all k-cliques in the graph (with $k \geq 3$), and each clique $c \in \mathcal{C}_k$ comprising k vertices $\{i_1,\cdots,i_k\} \subset \mathcal{V}$ is associated with a non-negative weight w_c. Given a subset of vertices $\mathcal{S} \subset \mathcal{V}$, define the function

$$h(\mathbf{x}) := \sum_{(i_1, i_2, \cdots, i_k) \in \mathcal{C}_k} w_c \prod_{j=1}^{k} x_{i_j}, \tag{8}$$

where $\mathbf{x}$ is the binary indicator vector of $\mathcal{S}$. Note that for a given subset of vertices $\mathcal{S} \subset \mathcal{V}$, h returns the weighted sum of k-cliques induced by $\mathcal{S}$. Then, we have the following result.

Corollary 1. *$-h$ is a submodular function.*

Proof. See Appendix C. □

Appendix C: Proof of Corollary 1

It suffices to show that each component function

$$h_c(x_{i_1}, \cdots, x_{i_k}) := -w_c \prod_{j=1}^{k} x_{i_j} \tag{9}$$

corresponding to a given clique $c \in \mathcal{C}_k$ is submodular. To do so, we employ the same sequence of observations as in the previous result. Namely, that each function is defined on a reduced ground set comprising vertices $\{i_1, \cdots, i_k\}$ and is symmetric w.r.t. to its arguments. Hence, w.l.o.g., we can fix the first $k-2$ variables $(x_{i_1}, \cdots, x_{i_{k-2}})$ to one and test whether the projection of the function

$$h_c(\underbrace{1, \cdots, 1}_{k-2}, x_{i_{k-1}}, x_{i_k}) = -w_c x_{i_{k-1}} x_{i_k} \tag{10}$$

onto the pair of variables $(x_{i_{k-1}}, x_{i_k})$ is submodular. Since w_c is non-negative, as shown previously, this is indeed the case.

Appendix D: Proof of Lemma 3

From the definitions of F_{uvw} and the base polytope of a submodular function, we obtain the following characterization of $\mathcal{B}_{F_{uvw}}$.

$$\begin{aligned} \mathcal{B}_{F_{uvw}} = \Big\{ & \mathbf{y} \in \mathbb{R}^n : y_u \leq 0, y_v \leq 0, y_w \leq 0, \\ & y_u + y_v \leq 0, y_v + y_w \leq 0, y_u + y_w \leq 0, \\ & y_u + y_v + y_w \leq -w_t, y_u + y_v + y_w = -w_t \Big\}. \end{aligned} \tag{11}$$

Observe that the first three inequalities imply the next three, whereas the final inequality is made redundant by the lone equality. The result then follows after eliminating all redundant inequalities.

Appendix E: Proof of Theorem 4

Let f_{uvw} denote the Lovász extension of F_{uvw}. Invoking the result of Lemma 3, we obtain

$$f_{uvw}(\mathbf{x}) = \max_{\mathbf{y} \in \mathcal{B}_{F_{uvw}}} \mathbf{y}^T \mathbf{x} = -w_t \min\{x_u, x_v, x_w\}. \tag{12}$$

The result then follows since $f_L = \sum_{(u,v,w) \in \Delta} f_{uvw}$.

Appendix F: Proof of Theorem 5

Our starting point is the result of Lemma 3, which states that the base polytope $\mathcal{B}_{F_{uvw}}$ is the probability simplex appropriately scaled and reflected about the origin. While every vector lying in this set can be expressed as the convex combination of the three extreme points of the scaled simplex, it possesses only two degrees of freedom (due to the summation constraint). Hence, it follows that $\mathcal{B}_{F_{uvw}}$ can be equivalently expressed as

$$\begin{aligned} \mathcal{B}_{F_{uvw}} &= \Big\{ \alpha_t \geq 0, \beta_t \geq 0, \\ &\quad \alpha_t + \beta_t \leq 1 : -w_t(\alpha_t \mathbf{e}_u + \beta_t \mathbf{e}_v + (1 - \alpha_t - \beta_t)\mathbf{e}_w) \Big\}, \\ &= \Big\{ \alpha_t \geq 0, \beta_t \geq 0, \\ &\quad \alpha_t + \beta_t \leq 1 : -w_t(\alpha_t(\mathbf{e}_u - \mathbf{e}_w) + \beta_t(\mathbf{e}_v - \mathbf{e}_w) + \mathbf{e}_w) \Big\}. \end{aligned} \tag{13}$$

Define the variables $\gamma_t := 1 - 3\alpha_t$ and $\delta_t := 1 - 3\beta_t$. By construction, these variables obey the relationships

$$\gamma_t \leq 1, \delta_t \leq 1, \gamma_t + \delta_t \geq -1. \tag{14}$$

On performing a change of variables, we can express $\mathcal{B}_{F_{uvw}}$ as

$$\begin{aligned} \mathcal{B}_{F_{uvw}} &= -w_t \left[\frac{(1-\gamma_t)(\mathbf{e}_u - \mathbf{e}_w) + (1-\delta_t)(\mathbf{e}_v - \mathbf{e}_w)}{3} + \mathbf{e}_w \right], \\ &= -w_t \left[\frac{(\mathbf{e}_u + \mathbf{e}_v + \mathbf{e}_w)}{3} - \frac{\gamma_t(\mathbf{e}_u - \mathbf{e}_w) + \delta_t(\mathbf{e}_v - \mathbf{e}_w)}{3} \right]. \end{aligned} \tag{15}$$

Since the polytope $\mathcal{B}_F$ is the Minkowski sum of the polytopes $\mathcal{B}_{F_{uvw}}$, we obtain the following expression

$$\begin{aligned} \mathcal{B}_F &= -\sum_{(u,v,w) \in \Delta} w_t(\mathbf{e}_u + \mathbf{e}_v + \mathbf{e}_w) \\ &\quad + \sum_{(u,v,w) \in \Delta} w_t \Big\{ \gamma_t(\mathbf{e}_u - \mathbf{e}_w) + \delta_t(\mathbf{e}_v - \mathbf{e}_w) \Big\} \\ &= -\mathbf{t} + \sum_{(u,v,w) \in \Delta} w_t \Big\{ \gamma_t(\mathbf{e}_u - \mathbf{e}_w) + \delta_t(\mathbf{e}_v - \mathbf{e}_w) \Big\}, \end{aligned} \tag{16}$$

where for every triangle $t \in (u, v, w) \in \Delta$, the variables $\{\gamma_t, \delta_t\}$ obey (14). The second equality stems from the fact that the contribution of each vertex $u \in \mathcal{V}$ to the first summand is precisely t_u. From the definition of the Lovász extension, we obtain

$$\begin{aligned} f_L(\mathbf{x}) &= -\mathbf{t}^T \mathbf{x} \\ &\quad + \sum_{(u,v,w) \in \Delta} w_t \max_{\substack{\gamma_t \leq 1, \delta_t \leq 1, \\ \gamma_t + \delta_t \geq -1}} \Big\{ \gamma_t(\mathbf{e}_u - \mathbf{e}_w) + \delta_t(\mathbf{e}_v - \mathbf{e}_w) \Big\}. \end{aligned} \tag{17}$$

We focus on each maximization subproblem within the summand, which is a linear program in the variables $\{\gamma_t, \delta_t\}$. We now make the following observation regarding the constraint set defined by (14).

$$\text{conv}((1,1), (-2,1), (1,-2)) \Leftrightarrow \begin{cases} \gamma_t \leq 1, \delta_t \leq 1, \\ \gamma_t + \delta_t \geq -1. \end{cases} \tag{18}$$

With the convex-hull representation of the inequality constraints in hand, and exploiting the fact that the optimal solution of a linear program is always attained at an extreme point of its feasible set, we conclude that the solution of each subproblem is given by

$$\max\{x_u + x_v - 2x_w, x_v + x_w - 2x_u, x_u + x_w - 2x_v\},$$

which concludes the proof.

Appendix G: Proof of Lemma 4

Define the function

$$h(\mathbf{x}) := \begin{cases} D_{\mathrm{KL}}(\mathbf{x}, \mathbf{y}^r), & \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{19}$$

Then, the MDA update at each step (i.e., problem (31)) can be equivalently written as

$$\mathbf{x}^{r+1} = \arg \min_{\mathbf{1}^T\mathbf{x}=k} h(\mathbf{x}). \tag{20}$$

Let $\nu \in \mathbb{R}$ denote the dual variable associated with the equality constraint. Then, the Lagrangian of the above problem is

$$L(\mathbf{x}, \nu) := \begin{cases} \sum_{i=1}^n x_i(\log(x_i/y_i^r) + \nu - 1), & \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{21}$$

Let $(\mathbf{x}^*, \nu^*)$ denote a pair of primal-dual optimal solutions for (20). Applying the Karush-Kuhn-Tucker (KKT) conditions (which are necessary and sufficient for optimality in this case) yields the conditions

$$\mathbf{x}^* = \arg \min_{\mathbf{0}\leq\mathbf{x}\leq\mathbf{1}} L(\mathbf{x}, \nu^*), \ \mathbf{1}^T\mathbf{x}^* = k. \tag{22a}$$

Since the Lagrangian is linearly separable in $\mathbf{x}$, the first condition can be further simplified as

$$x_i^* = \arg \min_{0\leq x_i\leq 1} \left\{ x_i(\log(x_i/y_i^r) + \nu^* - 1) \right\}, \forall\, i \in [n]. \tag{23}$$

Each sub-problem admits an analytical solution of the form

$$x_i^* = \min\{1, \exp(-\nu^*)y_i^r\}, \forall\, i \in [n], \tag{24}$$

which completes the proof.

Appendix H. Transitivity results

Figure 1 depicts the transitivity of the detected subgraphs as a function of the size on representative datasets. From Lemma 2, we know that for a fixed subgraph size, maximizing triangle density is a surrogate for maximizing transitivity as well. Since we developed a dedicated algorithm for TDkS, we expect it to perform better with regard to transitivity compared to the edge-based baselines, which is indeed the case.

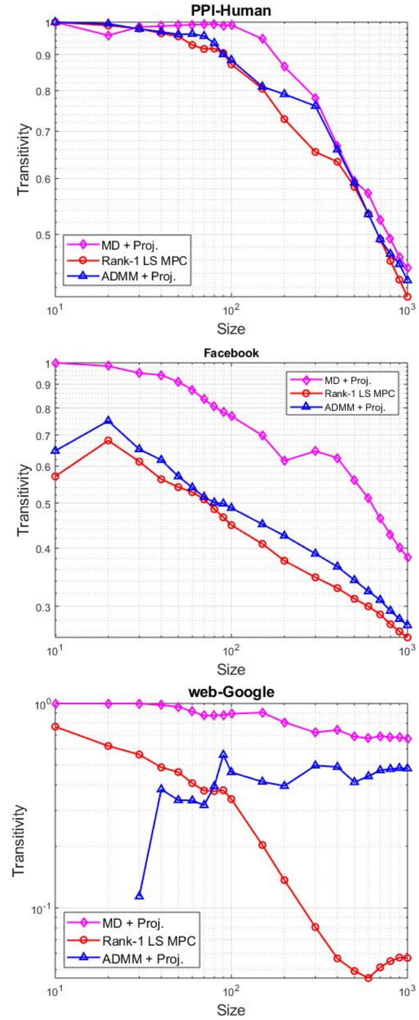

Fig. 1: Subgraph transitivity versus size.

Appendix I. Timing results

The timing results of the methods (in seconds) across different datasets are presented in Figure 2. It can be seen that the rank-1 principal component approximation approach of [5] is consistently the fastest, as it simply requires computing the top-eigenvector of the graph adjacency matrix once followed by extracting the support of the top-k entries to determine the candidate subgraph of size k. In contrast, the Lovász relaxation schemes for both DkS and TDkS (blue and magenta curves respectively) require more time in general as they entail solving convex optimization problems via iterative methods. For the larger datasets (bottom row of Figure 2), we remark that the Mirror Descent algorithm for TDkS is faster compared to the ADMM algorithm described in [6] for DkS.

Appendix J. Mirror Descent Parameters

We run Algorithm 1 with the learning-rate $\beta^r = (c/t_{\max})\sqrt{\log n/r}$, where the choice of constant c and the number of total iterations is listed in Table 1 for each dataset.

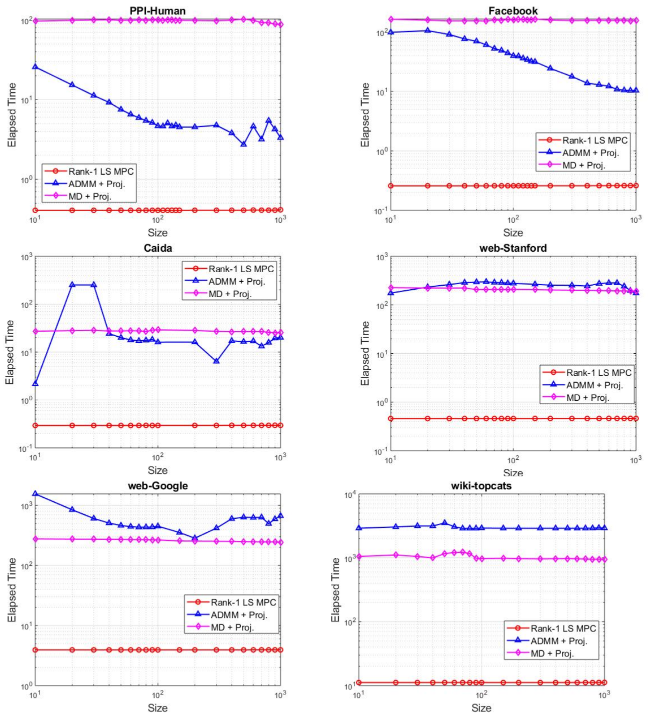

Fig. 2: Run-time of the methods across datasets. Red and blue curves are methods for DkS while the magenta curve is the MD algorithm for TDkS.

TABLE 1: Step-size constants (c) and number of iterations (N)

Graph	c	N
PPI-HUMAN	40	500
FACEBOOK-B	100	500
CAIDA	5	400
WEB-STANFORD	1000	200
WEB-GOOGLE	100	200
WIKI-TOPCATS	1000	200

REFERENCES

[1] J. B. Kruskal, "The number of simplices in a complex," *Mathematical optimization techniques*, vol. 10, pp. 251–278, 1963.

[2] G. O. Katona, "A simple proof of the erdös-chao ko-rado theorem," *Journal of Combinatorial Theory, Series B*, vol. 13, no. 2, pp. 183–184, 1972.

[3] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," in *European conference on computer vision*, pp. 65–81, Springer, 2002.

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

[5] D. Papailiopoulos, I. Mitliagkas, A. Dimakis, and C. Caramanis, "Finding dense subgraphs via low-rank bilinear optimization," in *International Conference on Machine Learning*, pp. 1890–1898, 2014.

[6] A. Konar and N. D. Sidiropoulos, "Exploring the subgraph density-size trade-off via the lovaśz extension," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 743–751, 2021.