On Finding Socially Tenuous Groups for Online Social Networks

Chih-Ya Shen
Department of Computer Science
National Tsing Hua Univ., Taiwan
chihya@cs.nthu.edu.tw

Liang-Hao Huang
Institute of Information Science
Academia Sinica, Taiwan
r90221003@gmail.com

De-Nian Yang
Institute of Information Science
Academia Sinica, Taiwan
dnyang@iis.sinica.edu.tw

Hong-Han Shuai
Department of Electrical and
Computer Engineering
National Chiao Tung Univ., Taiwan
hhshuai@nctu.edu.tw

Wang-Chien Lee
Department of Computer Science
and Engineering
The Pennsylvania State Univ., USA
wlee@cse.psu.edu

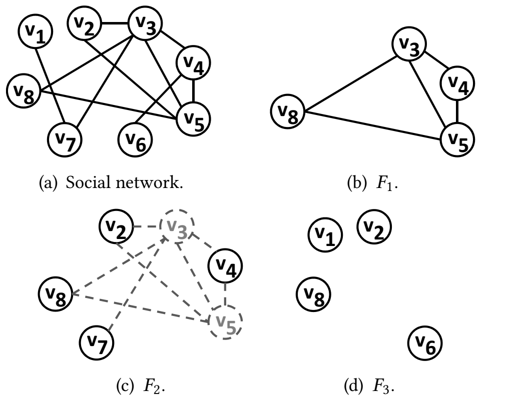
Ming-Syan Chen
Department of Electrical Engineering
National Taiwan Univ., Taiwan
mschen@ntu.edu.tw

ABSTRACT

Existing research on finding social groups mostly focuses on *dense* subgraphs in social networks. However, finding *socially tenuous groups* also has many important applications. In this paper, we introduce the notion of *k-triangles* to measure the tenuity of a group. We then formulate a new research problem, *Minimum k-Triangle Disconnected Group (MkTG)*, to find a socially tenuous group from online social networks. We prove that MkTG is NP-Hard and inapproximable within any ratio in arbitrary graphs but polynomial-time tractable in *threshold graphs*. Two algorithms, namely *TERA* and *TERA-ADV*, are designed to exploit graph-theoretical approaches for solving MkTG on general graphs effectively and efficiently. Experimental results on seven real datasets manifest that the proposed algorithms outperform existing approaches in both efficiency and solution quality.

1 INTRODUCTION

With the popularity and wide accessibility of online social networks (OSNs), e.g., Facebook, LiveJournal, LinkedIn, research on finding various *social groups* for community detection [14] and activity coordination [15, 17] has drawn increasing attention. Existing research works mostly focus on extracting *dense* groups of socially connected individuals from online social networks. However, *socially tenuous groups* (STGs), i.e., subgraphs with few social interactions and weak relationships among members, have not received much research attention[1]. We argue that STGs have many real needs, e.g., psychoeducational group formation and reviewer selection, and thus deserve more research effort.



(a) Social network. (b) F_1.

(c) F_2. (d) F_3.

Figure 1: Motivating example.

Psychoeducational group formation. For group therapy for substance abuse treatment, an important task is to form psychoeducational or cognitive-behavioral groups [9]. In addition to selecting individuals with similar disorder symptoms and behaviors, one essential criterion for group formation is to assign patients who do not know each other (and sometimes not even multi-hop friends) to form an STG [9]. Forming such an STG is critical for engaging the group members to share their feeling without hesitation. Moreover, it's less likely for members of such an STG to form *subgroups* which may act against other members during therapy sessions.

Consider a scenario where a clinical psychologist would like to select four patients from candidates in Figure 1(a) (their social relationships are illustrated) to form a psychoeducational therapy group. Note that $F_1 = \{v_3, v_4, v_5, v_8\}$ (see Figure 1(b)) may not be a good choice because there are many edges (social relationships) among them. While $F_2 = \{v_2, v_4, v_7, v_8\}$ (see Figure 1(c)) does not have direct social relationships, each pair of them forms a friend-of-friend relationship (through v_3 or v_5), which may lead to subgroups (due to common friends) or make them hesitate to share their private experiences (which may be leaked to common friends). As shown, $F_3 = \{v_1, v_2, v_6, v_8\}$ (see Figure 1(d)) is the best choice because the patients induce no direct or friend-of-friend relationships, minimizing the chance for private information shared in the therapy group to spread out.

Reviewer selection. STG also finds its applications in paper reviews. Conference program chairs need to assign experts to review

[1] Reducing only the number of edges in the group is not sufficient for real applications (explained later).

KDD'17, August 13–17, 2017, Halifax, NS, Canada.
© 2017 ACM. 978-1-4503-4887-4/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3097983.3097995

papers. Besides matching the expertise of reviewers with the topics of submissions, it is crucial to avoid assigning reviewers socially close to each other and the authors of a paper in order to ensure unbiased assessments. While most review systems have utilized co-authorship, affiliations, and countries to avoid conflict of interests, current systems do not carefully consider social tenuity among the authors and reviewers[2]. STG can help!

To find socially tenuous subgraphs (STGs)[3], the tenuity of an STG needs to be properly modelled. Thus, we introduce the notion of *k-triangles* as the basis for measuring tenuity of STGs. A k-triangle in a social network exists when three individuals are located within k hops from each other. In the following, we first formally define the k-triangle and then discuss its properties and advantages.

Definition 1.1. A k-triangle is a triplet of vertices $\{u, v, w\}$, such that $d_G(u, v) \leq k$, $d_G(u, w) \leq k$, and $d_G(v, w) \leq k$, where $d_G(x, y)$ is the shortest path distance (in hops) between two vertices x, y on G.

It is worth clarifying that, for a k-triangle $\{u, v, w\}$ in a subgraph F, the shortest path distance between each pair of vertices is computed on the overall graph G instead of F, because the social relationships of the selected members go beyond F. For example, consider $F = \{v_1, v_3, v_6\}$ in Figure 1(a). The shortest path distance on F from v_1 to v_3 is infinite, but it's 2 on G.

Triangles serve well as a basic unit to measure various density relationships among the neighborhoods of vertices in a network, e.g., clustering coefficient, transitivity ratio, and k-trusses [8]. Countering the idea, the fewer k-triangles in a group, the more tenuous is the group. In fact, a $(k-1)$-triangle is a k-triangle. If a group has no k-triangle, it does not have any $(k-1)$-triangle, $(k-2)$-triangle,..., and 1-triangle. In fact, if a group does not have any k-triangle, it *does not* include any subgraph H with $|H| \geq 3$ in which $d_G(u, v) \leq k$ for $u, v \in H$[4]. Thus, the count of k-triangles serves very well for measuring the *group tenuity*.

Note that k-triangles have great advantages in measuring the group tenuity. First, k-triangles capture social relationships up to k hops. Two individuals with more common friends within k hops have more k-triangles. Consider Figure 1(c) as an example. $F_2 = \{v_2, v_4, v_7, v_8\}$ has no triangles (no 1-triangles) among them, but there exist four 2-triangles, i.e., each pair of vertices has a common friend. In contrast, in Figure 1(d), $F_3 = \{v_1, v_2, v_6, v_8\}$, which has no 2-triangle, is a better tenuous group than F_2. Second, k-triangles serve well as a basic block for many other graph structures, such as paths, trees, stars, or even cliques. Consider the example in Figure 1. Path $\{v_1, v_7, v_3\}$ is a 2-triangle, while $\{v_3, v_4, v_7, v_8\}$ (a tree rooted at v_4 as well as a star centered at v_3) contains four 2-triangles. Moreover, each clique of size p contains exactly C_3^p triangles. If the number of k-triangles is minimized in a subgraph, the aforementioned graph structures (which imply a certain degree of denseness) are also effectively minimized.

In this paper, we formulate a new research problem, namely *Minimum k-Triangle Disconnected Group (MkTG)*, which finds an STG by optimizing the group tenuity – *minimizing the number of k-Triangles normalized by the group size.* Given a social network $G = (V, E)$, MkTG finds a group F from G with the minimum number of k-triangles for each vertex subject to the following constraints. 1) F contains no fewer than n individuals (size constraint). 2) There is no edge in F (no-pair constraint). The size constraint can be specified based on practical need, e.g., finding at least three reviewers for paper review. The no-pair constraint guarantees that F does not contain any ego friends (directly connected friends), which is also important for forming psychoeducational groups and finding paper reviewers. Please note that F may still contain a lot of k-triangles ($k \geq 2$) even when F contains no directly connected friends. For example, in Figure 1(c), $F_2 = \{v_2, v_4, v_7, v_8\}$ contains no edge. However, there are four 2-triangles in F_2.

The MkTG problem is nontrivial due to the entangled group size constraint, no-pair constraint and social tenuity objective function. We prove that the MkTG problem is NP-Hard and inapproximable within any ratio. After the hardness analysis, we take steps to solve the MkTG problem systematically. We propose an efficient and effective algorithm, namely *Triangle and Edge Reduction Algorithm (TERA)*, for solving the MkTG problem on general graphs. Moreover, we devise advanced processing strategies for TERA, namely *TERA with Advanced Processing Strategies (TERA-ADV)*, which incorporate graph-theoretical strategies, i.e., *Simplicial Pruning* and *Vicinal Partition and Elimination*, to significantly avoid examining redundant vertices in the graph. Then, we consider the MkTG problem in *threshold graphs* [31]. We pay special attention on threshold graphs because the structural properties (e.g., degree distribution, largest component size, edge density, and local clustering coefficient) of many popular online social networks are similar to those of threshold graphs [32]. We propose a polynomial-time algorithm to find the optimal solution based on the notion of *vicinal pre-order*.

The contributions are summarized as follows.

- We identify a new problem of finding tenuous groups in online social networks and introduce a novel notion of k-triangle for measuring the tenuity of groups. Accordingly, we formulate the Minimum k-Triangle Disconnected Group (MkTG) problem, and prove it NP-Hard and inapproximable within any ratio.
- For MkTG in general graphs, we devise two algorithms, namely *Triangle and Edge Reduction Algorithm (TERA)* and *TERA with Advanced Processing Strategies (TERA-ADV)*. The latter employs *Simplicial Pruning* and *Vicinal Partition and Elimination* based on graph theory to find solutions efficiently and effectively.
- We study the MkTG problem in a special class of graphs, i.e., *threshold graphs*, which have graph properties very similar to many well-known online social networks. We show that our proposed algorithms can obtain the optimal solution in polynomial time according to the notion of vicinal pre-order.

[2] In fact, some people argue that double-blind review may not really be blind because the research community is small. Sometimes it is not difficult to guess the author identities during the paper bidding phase.

[3] In this paper, STG refers to both *socially tenuous group* and socially tenuous subgraph, which are used interchangeably under the context of this paper.

[4] For the case of $|H| = 2$, we can add a virtual vertex q linking through artificial paths to every vertex in the group. By controlling the hop number of the artificial paths, eliminating k-triangles in the original group can be transformed to eliminating the subgraphs H with $|H| = 2$.

- We perform extensive experiments on real datasets to evaluate the proposed algorithms and different baselines. Experimental results show that our algorithms outperform the baselines in both solution quality and efficiency.

The paper is organized as follows. Section 2 formulates the MkTG problem. Section 3 introduces the works relevant to this paper. Sections 4 and 5 present the algorithms for MkTG on general graphs and threshold graphs, respectively. Section 6 presents the experimental results, and Section 7 concludes this paper.

2 PROBLEM FORMULATION AND HARDNESS

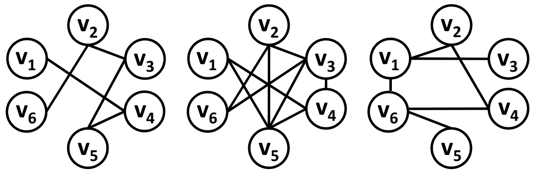
Given a social network $G = (V, E)$, let $|F|$ denote the number of vertices in F, and $\Delta_k(F)$ denote the number of k-triangles in F. The MkTG problem is formulated as follows.

Problem: Minimum k-Triangle Disconnected Group (MkTG).
Given: A social network $G = (V, E)$, size constraint n, and tenuity parameter k.
Find: A subgraph $F \subseteq G$ where $|F| \geq n$ (size constraint) and $\nexists u, v \in F$ with edge $(u, v) \in E$ (no-pair constraint), such that $\frac{\Delta_k(F)}{|F|}$ is minimized.

When the group size $|F|$ (or the size constraint n) increases, it becomes harder to minimize the number of k-triangles, and it is more inclined for F to violate the no-pair constraint. Therefore, the objective function of the group includes a normalization term[5] to encourage exploring groups with different sizes, instead of always trying the smallest group (i.e., $|F| = n$). Intuitively, the tenuity objective above aims to *minimize the average number of k-triangles for each member in the group*. Therefore, one of the challenges for MkTG lies in achieving a good balance between the group size $|F|$ (or the size constraint n) and the number of k-triangles in F. On the other hand, the tenuity parameter k also has crucial impact on the number of k-triangles in F. As k increases, it becomes more challenging to find a subgraph with a small number of k-triangles because the number of k-hop friends increases for each vertex.

One approach for MkTG is to first construct the k-hop graph (detailed later), G_k of G, then construct the complement graph $\widehat{G}_k$ of G_k, and employ existing algorithms to extract dense subgraphs from $\widehat{G}_k$. Specifically, given input graph $G = (V, E)$ and parameter k, the k-hop graph $G_k = (V, E_k)$ retains the vertex set V and augments the edge set E into E_k. An edge $(u, v) \in E_k$ exists if and only if u and v are within k hops on G, i.e., $d_G(u, v) \leq k$. By transforming G into G_k, we ensure that a k-triangle $\{u, v, w\}$ exists in G if and only if $\{u, v, w\}$ is a 1-triangle in G_k. However, finding dense subgraphs on $\widehat{G}_k$ cannot obtain the good solutions for MkTG due to the interplay of k-triangles and the no-pair constraint.

A counterexample of the above approach is shown in Figure 2. Given G in Figure 2(a) and an MkTG with $k = 2$, $n = 3$, Figure 2(b) is the 2-hop graph, i.e., G_2, and Figure 2(c) is the complement graph of G_2, i.e., $\widehat{G}_2$. One optimal solution of this MkTG instance on G is $\{v_1, v_2, v_5\}$ with no 2-triangles which satisfies the no-pair constraint. In contrast, if we employ the algorithm to find the subgraph F_d with maximum density (i.e., to maximize $\frac{|E(F_d)|}{|V(F_d)|}$) on $\widehat{G}_2$ in Figure 2(c), we have $F_d = \{v_1, v_2, v_3, v_4, v_6\}$ with $\frac{|E(F_d)|}{|V(F_d)|} = 1$. However, there exists a 2-triangle ($\{v_2, v_3, v_6\}$) in F_d on the original graph G, and F_d does not satisfy the no-pair constraint. This example manifests that MkTG is very challenging and straightforward approaches cannot solve it properly.



(a) G. (b) G_2. (c) $\widehat{G}_2$.

Figure 2: Counterexample to complementary graph approaches.

The proposed MkTG problem is *NP-Hard* and *inapproximable within any ratio*, which can be proved with a gap-introducing reduction from the Maximum Independent Set (MIS) problem (we present the hardness analysis in [10]). Therefore, it is impossible to design an approximation algorithm with any finite ratio for MkTG in an arbitrary graph. However, after carefully analyzing the problem, we observe that it is still possible to obtain the optimal solution to the MkTG problem in polynomial time for an important graph class, i.e., *threshold graphs* [31]. We are particularly interested in MkTG on threshold graphs due to their correspondence to many real-life social networks. For example, as reported in a recent study [32], the structural properties of the intergroup networks on online social networks (e.g., LiveJournal, Flickr, Youtube), including degree distribution, largest component size, edge density, and local clustering coefficient, match well with the structure of threshold graphs. Therefore, we also analyze the MkTG problem on threshold graphs.

3 RELATED WORKS

Extracting dense subgraphs or communities is an important research topic with many social applications. Various social cohesive measures have been proposed to find dense subgraphs, e.g., diameter [8], density [11], clique and its variations [12]. Moreover, community detection methods have been actively studied to extract densely connected subgraphs from social networks [13, 14], while research on organizing social groups based on tightness among existing friends and other crucial factors [15–17] has also been studied. GSGQ [15] and MRGQ [16] extract socially dense groups with spatial constraints, while user preference is also examined [17]. Although the above research covers various applications, they focus on extracting *dense* subgraphs from online social networks. In contrast, this paper explores a new problem of finding subgraphs with the minimum number of k-triangles. Therefore, the algorithms in prior works cannot be applied to the MkTG problem.

A recent line of research focuses on graph sparsification, simplification, sparse spanners, and sampling for massive networks [18–22]. These algorithms aim to find concise and representative subgraphs with the essential graph properties preserved so that

[5] The normalization term could be $(|F|)^t$ with $t \geq 1$. Without loss of generality, we consider the case where = 1.

the results are still informative for network analysis. For example, DEDS [18] processes the original graph into multiple smaller networks to improve the efficiency of link prediction, whereas the structure of a network is simplified for clustering [19]. In contrast, MkTG does not extract a subgraph with the graph properties preserved. It aims to find a subset of mutually remote vertices with the minimum number of k-triangles.

Some theoretical works analyze triangle-free graphs [23–26]. The number of independent sets in triangle-free graphs is studied in [23], while the number of pentagons in triangle-free graphs is derived in [24, 25]. Nevertheless, it is worth noting that, triangle-free graphs cannot ensure mutual tenuity. Even if a subgraph F contains no triangles, members of F may still be socially close to each other, e.g., friends-of-friends. Most importantly, the above works focus on analyzing the properties of triangle-free graphs, but apparently online social networks are not triangle-free. Some theoretical works [27–29] also analyze the properties of other sparse graphs, e.g., chordal graphs, interval graphs, and perfect graphs [27, 29]. Nevertheless, the above research does not aim to extract a subgraph from a social network.

4 MkTG ON GENERAL GRAPHS

In this section, we propose two new algorithms, namely *Triangle and Edge Reduction Algorithm (TERA)* and *TERA with Advanced Processing Strategies (TERA-ADV)*, for finding good solutions to the MkTG problem on general graphs efficiently. While it is inapproximable within any ratio as shown in Section 2, we prove later in Section 5 that the proposed TERA can find the optimal solution for MkTG on threshold graphs, which have similar properties with many online social networks, in polynomial time.

To solve MkTG, several crucial factors need to be carefully examined. The first factor is the tenuity objective and its interplay with the no-pair constraint, i.e., there must exist no edge in F. To minimize the number of k-triangles, one greedy approach is to iteratively select the vertices involved in few triangles. Nevertheless, these vertices may share common incident edges and thus are not able to ensure the no-pair constraint. The second factor is the trade-off between the minimum group size n and the number of k-triangles. The objective function aims to minimize the *average* number of k-triangles in F, i.e., to minimize $\frac{\Delta_k(F)}{|F|}$. As k and n increase, it is more likely to have k-triangles in F. Therefore, how to strike a balance between the group size and the number of k-triangles is crucial to minimizing the objective value.

To address the above factors, three ideas are considered in our algorithm design: 1) To include *isolated* vertices because isolated vertices ensure both the no-pair constraint and the minimization of the number of k-triangles. However, relying solely on the isolated vertices is not practical because the number of isolated vertices is usually small, especially in online social networks. 2) To identify the vertices appearing in many k-triangles. If these vertices are identified and removed from the resulting group F, the number of k-triangles can be significantly reduced. Note that a vertex with a great degree is not necessarily involved in many k-triangles in F since not all its neighboring vertices are always selected in F. 3) To generate multiple candidate groups of different sizes in order to extract the one with the best balance between the group size and the number of k-triangles.

In the following, we first present the basic TERA in Section 4.1 and then enhance it with advanced pre-processing and pruning techniques in Section 4.2. The proposed advanced techniques can pre-process the social networks offline to support arbitrary parameters k and n in MkTG issued by a user online.

4.1 Triangle and Edge Reduction Algorithm

In TERA, we first assign each vertex v with a weight $w(v)$, where $w(v)$ is the number of k-triangles v is involved in. Note that this step can be done efficiently offline by transforming G into the k-hop graph G_k (as mentioned in Section 2)[6] and then assigning the number of triangles each vertex v is involved in G_k as $w(v)$. Then, given the runtime parameters k and n, TERA iteratively removes a vertex v_i (and its incident edges) with the largest vertex weight from G. More specifically, let H_{i+1} denote the graph after removing vertex v_i from H_i in iteration i. Initially, H_1 is set as G. At each iteration afterwards, H_{i+1} represents the graph $H_i - \{v_i\}$.

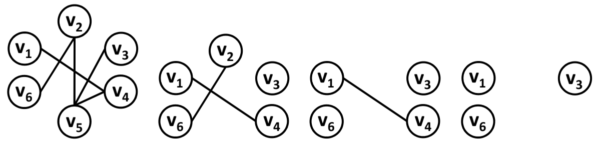
Algorithm 1 Triangle and Edge Reduction (TERA)

```
Input: $G = (V, E)$, $n$, $k$
$H_1 \leftarrow G$, $i \leftarrow 1$, $\mathbb{U} \leftarrow \varnothing$
while $|H_i| > n$ do
    identify $v_i \in H_i$ as the vertex with the maximum $w(v_i)$ (break ties by selecting the vertex with larger degree in $H_i$)
    $H_{i+1} \leftarrow H_i - \{v_i\}$
    if $H_{i+1}$ satisfies no-pair constraint then
        $\mathbb{U} \leftarrow \mathbb{U} \cup \{H_{i+1}\}$
    end if
    $i \leftarrow i + 1$
end while
$H^* \leftarrow \arg\min_{H_j \in \mathbb{U}} \frac{\Delta_k(H_j)}{|H_j|}$
if $H^* = \emptyset$ then
    $H^* \leftarrow \arg\min_{\forall j} \frac{\Delta_k(H_j)}{|H_j|}$
end if
output $H^*$
```

The intuition is that removing the vertices that are involved in a large number of k-triangles may likely reduce the number of k-triangles and keep isolated vertices in the remaining graph. Thus, v_i selected in the i-th iteration is the vertex which has the largest $w(v)$ in H_i, i.e., the remaining vertex that incurs the maximum number of k-triangles. Note that the vertex with a larger degree is more likely to violate the no-pair constraint. Thus, we prioritize the selection of the vertex with a larger degree on the induced subgraph of H_i on G if there are multiple vertices involved in the same number of k-triangles.

Accordingly, H_{i+1} is generated by removing v_i and its corresponding edges from H_i. Afterwards, H_{i+1} is processed in the next iteration $i+1$. The above procedure ends when $|H_i| \leq n$. Finally, we extract the graph $H^* \in \{H_1, H_2, ...\}$ with the minimum objective value that satisfies the no-pair constraint as the output solution. It

[6] For most networks, their k-hop graphs become complete graphs for $k \geq 6$. Therefore, we only need to consider the k-hop graphs for $2 \leq k \leq 5$. To reduce space consumption, we store only one copy of vertex set in the k-hop graph. Each edge e in the k-hop graph is marked with an integer k_e indicating e appears when $k \geq k_e$.



(a) $G = H_1$. (b) H_2. (c) H_3. (d) H_4.

Figure 3: Running example of TERA with $k = 2$ and $n = 3$.

is worth noting that as proved in the hardness analysis, deciding if MkTG has any feasible solution following the no-pair constraint is NP-Complete. Therefore, TERA and any other algorithm may not be able to find a feasible solution when an MkTG instance does not contain any feasible solution (otherwise, $P = NP$ holds)[7].

Example 4.1. Figure 3 is an example of TERA with $k = 2$ and $n = 3$. TERA starts with $H_1 = G$ (Figure 3(a)). Since v_5 is involved in the maximum number of 2-triangles, v_5 is removed from H_1 and produces H_2 (Figure 3(b)). Then, v_2 is removed from H_2 to create H_3 (Figure 3(c)). Finally, we remove v_4 and $H_4 = \{v_1, v_3, v_6\}$. The objective value of H_4 is minimum among all H_i, i.e., 0, and H_4 satisfies the no-pair constraint. Therefore, H_4 is returned by TERA.

Time Complexity Analysis of TERA. Given the input parameters k and n, TERA removes v_i and its incident edges in each iteration i, which requires $O(\delta_G)$ time, where δ_G is the maximum degree in G. Computing the number of k-triangles reduced by removing v_i takes $O(\delta_G^2)$ time. Since there are at most $O(|V|)$ iterations, the overall time complexity is $O(\delta_G^2|V|)$.

4.2 TERA with Advanced Strategies

Through the analysis and evaluation of TERA, we observe that it is not necessary to examine the whole vertex set in G, because many vertices will never satisfy the no-pair constraint. Moreover, many vertices are redundant and can be removed from G because these vertices can always be replaced to reduce the objective value. Therefore, we propose an advanced version of TERA, namely *TERA-ADV*, by exploring the above observations. TERA-ADV includes two main ideas: 1) we propose a pre-processing strategy, namely *Simplicial Pruning (SP)*, that significantly reduces the size of vertex set before TERA starts. 2) We partition the vertex set into several components based on graph theory and devise a strategy, namely *Vicinal Partition and Elimination (VPE)*, to eliminate redundant examinations in TERA. Conventional pruning strategies are usually performed at runtime. In contrast, Simplicial Pruning and Vicinal Partition and Elimination can be performed offline for arbitrary k and n in any problem instance before any query arrives. By removing a significant number of redundant vertices, these strategies reduce on-line computation cost and storage cost of the graph significantly. Let G_{SP} and G_v be the graphs after SP and VPE, respectively.

Simplicial Pruning (SP). Given a graph $G = (V, E)$ and a vertex $x \in V$, let $N(x)$ denote the 1-hop neighbors of x, and let $N[x]$ denote the *closed neighbors* of x, i.e., $N[x] = N(x) \cup \{x\}$. For example, $N[v_1] = \{v_1, v_3, v_9\}$ in Figure 4(a). A *simplicial vertex* $s \in V$ is a vertex where $N(s)$ forms a clique. For example, v_1, v_2, v_{10} are simplicial vertices. Simplicial vertices have nice properties and thus play important roles in the proposed Simplicial Pruning.

For the first nice property of simplicial vertices, given a feasible solution H (i.e., satisfying the no-pair constraint) of the MkTG problem and a simplicial vertex s, at most one vertex is overlapped by H and $N[s]$.

Lemma 4.2. *Given a feasible solution H and a simplicial vertex s, $|H \cap N[s]| \leq 1$ holds.*

Proof. Since s is a simplicial vertex and $N(s)$ forms a clique, $(x, y) \in E$ for any $x, y \in N[s]$. If more than one vertices in $N[s]$ are included in H, H must have at least one edge, and H will not be a feasible solution. □

Therefore, for a simplicial vertex s, we can select a vertex in $N[s]$ and trim others because at most one vertex in $N[s]$ would appear in any feasible solution to ensure the no-pair constraint. However, the identity of the chosen vertex is still not clear. Therefore, the second nice property of simplicial vertices states that, we can always choose the simplicial vertex s itself (and trim all other vertices in $N[s]$), which must satisfy the no-pair constraint and generates the minimal objective value.

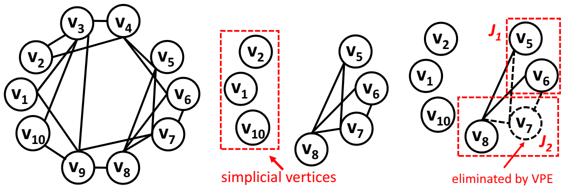
Lemma 4.3. *Given a feasible solution H and a simplicial vertex s, if $y = H \cap N[s]$ and $H' = H - \{y\} \cup \{s\}$, then H' is no worse than H.*

Proof. Let $E(s)$ denote the set of incident edges of s. For a simplicial vertex s and $\forall u \in N[s]$, $|E(s)| = \min_{u \in N[s]} |E(u)|$ and $N[s] \subseteq N[u]$ must hold because each vertex $u \in N[s]$ also connects to every other vertex in $N[s]$, and u may have edges linking to other vertices outside $N[s]$. Since the number of edges incident on s is minimum among $N[s]$, and $N[s] \subseteq N[u], \forall u \in N[s]$, creating $H' = H - \{y\} \cup s$ by substituting y with s still allows H' to satisfy the no-pair constraint, if H already follows the no-pair constraint. Moreover, inequality $\frac{\Delta_k(H')}{|H'|} \leq \frac{\Delta_k(H)}{|H|}$ holds because 1) $|H'| = |H|$, and 2) every k-triangle connecting to s can be adjusted to connect to u (since $N(s) \subseteq N(u)$). □

Based on the above two properties, Simplicial Pruning proceeds as follows. Given the input graph $G = (V, E)$, we first extract the set of all simplicial vertices $\widehat{S} = \{s_1, s_2, ...\}$ according to [31]. Then, Simplicial Pruning removes $\bigcup_{s_i \in \widehat{S}} N(s_i)$ and their incident edges from G to produce G_{SP}. Therefore, for any feasible solution H obtained from G, we can always find a solution H' in G_{SP} no worse than H. That is, the vertices removed from G are indeed redundant and able to be safely removed. Moreover, if $|G_{SP}| < n$ in any instance of MkTG, we guarantee that the instance has no feasible solution.

Example 4.4. Consider the example in Figure 4(a) with $k = 2$ and $n = 4$ again. SP first identifies the set of simplicial vertices, i.e.,

[7] An alternative approach is to add the number of edges in a solution to the objective function, by replacing the tenuity objective of F with $\frac{\Delta_k(F)+E(F)}{|F|}$, where $E(F)$ is the number of edges in F. TERA can solve the above problem by including a set of *virtual nodes* R that links to every vertex in the k-hop graph G_k during the preprocessing step. In this case, if any two vertices $u, v \in F$ share an edge, the edge will be included in $|R|$ k-triangles, and the number of k-triangles thereby increases. Thus minimizing this new objective function would deter F from including edges.



(a) Graph G. (b) After SP. (c) After VPE.

Figure 4: Running example of TERA-ADV.

$\widehat{S} = \{v_1, v_2, v_{10}\}$, and removes $N(v_1)$, $N(v_2)$, and $N(v_{10})$ from G to produce G_{SP}, where G_{SP} is shown in Figure 4(b).

Theorem 4.5. *For any feasible solution H obtained from G, there exists a solution H' in G_{SP} no worse than H. Moreover, if $|G_{SP}| < n$, there is no feasible solution to the MkTG problem.*

Proof. Let H be a feasible solution obtained from G, and assume $y \in N(s_i)$ and $y \in H$, where s_i is y's corresponding simplicial vertex. Based on Lemma 4.3, $H - \{y\} \cup \{s_i\}$ is a better solution than H. Therefore, there is always a solution H' in G_{SP} no worse than H.

On the other hand, if $|G_{SP}| < n$, suppose that the set of simplicial vertices in G_{SP} is $\bar{S} = \{\bar{s}_1, \bar{s}_2, ...\}$. That is, $G_{SP} = \bar{S} \cup W$, where W is the set of vertices that are neither simplicial vertices nor connected to any simplicial vertices (i.e., those not pruned from G). Then, $\bigcup_{\forall \bar{s}_i \in \bar{S}} N[\bar{s}_i] = G - W$ holds. For any feasible solution H, $H \cap N[\bar{s}_i]$ contains at most one vertex. Therefore, $H \cap (\bigcup_{\forall \bar{s}_i \in \bar{S}} N[\bar{s}_i])$ contains at most $|\bar{S}|$ vertices (otherwise, H violates the no-pair constraint). In other words, if $|G_{SP}| = |W \cup (\bigcup_{\forall \bar{s}_i \in \bar{S}} N[\bar{s}_i])| < n$, there is no solution for the instance of the MkTG. The theorem follows. □

Vicinal Partition and Elimination (VPE). We first use *vicinal pre-order* [31] to describe the relation of the common neighbors among two neighbor vertices x and y in any graph. The vicinal pre-order $x \lesssim y$ states that all vertices (except y) adjacent to x are also adjacent to y. In other words, $x \lesssim y$ holds if and only if $N(x) \subseteq N[y]$ holds, i.e., x's 1-hop neighbors are all included in y's closed neighbors. For example, $v_1 \lesssim v_{10}$ in Figure 4(a).

Based on vicinal pre-order, we partition the graph G_{SP} into a set of non-overlapping subgraphs $\{J_1, J_2, ...\}$, where $x \lesssim y$ and $y \lesssim x$ hold for any $x, y \in J_i$, i.e., x and y share the same neighbors. For example, $J_1 = \{v_5, v_6\}$ and $J_2 = \{v_7, v_8\}$ as shown in Figure 4(c). This enables TERA to further eliminate redundant vertices in G_{SP} (detailed later). Please note that an induced subgraph J_i either has no edges or forms a complete graph, where $w(x)$ is the number of k-triangles in which vertex x is involved for G_{SP}.

Based on the above observation, for each J_i that induces a complete graph, we can remove all the vertices in J_i except one vertex. The reason is that if a subgraph $\widehat{H}$ contains two vertices $x, y \in J_i$, $\widehat{H}$ is not a feasible solution because it violates the no-pair constraint. Therefore, since the incident edge sets of any vertices $x, y \in J_i$ are identical, we can remove any $|J_i| - 1$ vertices from J_i.

Vicinal Partition and Elimination (VPE) prunes the input graph and constructs G_v with the following two steps. In Step 1, the *Partition* step, VPE partitions the graph G_{SP} into $\{J_1, J_2, ...\}$ based on vicinal pre-order, i.e., each J_i contains vertices x, y if $x \lesssim y$ and $y \lesssim x$. Given G_{SP} produced by the SP strategy as shown in Figure 4(b), the vertices are partitioned into J_1 and J_2 by VPE, as shown in Figure 4(c). Then, in Step 2, the *Elimination* step, VPE identifies the set of subgraphs $\mathcal{J}_c = \{J_i : (x, y) \in E, \forall x, y \in J_i\}$ whose induced subgraphs are complete graphs. Then for those subgraphs in $\mathcal{J}_c$, VPE removes all the vertices (except one vertex) in each $J_i \in \mathcal{J}_c$ to form G_v. Please note that, VPE is performed offline, removing redundant vertices before the query comes. Moreover, if $|G_v| < n$, no feasible solution exists for the MkTG instance. This condition enables VPE to effectively prune redundant vertices.

Example 4.6. In Figure 4(c), vertex v_7 in J_2 is removed because J_2 induces a complete graph. After VPE, TERA starts on G_v, which contains vertices $\{v_1, v_2, v_5, v_6, v_8, v_{10}\}$, as shown in Figure 4(c). TERA then obtains the solution $\{v_1, v_2, v_8, v_{10}\}$ with objective value 0 following the no-pair constraint, which is an optimal solution.

Theorem 4.7. *If $|G_v| < n$, no feasible solution exists for the MkTG instance, where G_v is the output graph of VPE.*

Proof. Assume $\mathcal{J}_c = \{J_{c,1}, J_{c,2}, ..., J_{c,q}\}$ where each $J_{c,q}$ has its induced subgraph as a complete graph. After removing redundant vertices, let v_i denote a remaining vertex in $J_{c,i}$. Then, G_v can be represented as $G_v = W \cup \bigcup_{\forall i} v_i$, where $W = G_{SP} - J_c$, i.e., W is union of J_i with its induced subgraph having no edges. Since at most one vertex in each $J_{c,i}$ can be included in a feasible solution, if $|G_v| < n$, no solution exists for the MkTG instance. The theorem follows. □

Time Complexity Analysis of TERA-ADV (Offline Processing). Let δ_G denote the maximal degree of a vertex in G. Checking if a vertex v is a simplicial vertex requires $O(\delta_G^2)$ time. SP takes $O(\delta_G^2|V|)$ time, and VPE requires $O(|V|^2)$ time. The overall time complexity of performing SP and VPE is $O(\delta_G^2|V| + |V|^2)$.

5 SOLUTION OPTIMALITY OF MkTG ON THRESHOLD GRAPHS

In the following, we prove that TERA and TERA-ADV can find the optimal solutions of MkTG on *threshold graphs* [31], which are very similar to many well-known online social networks (e.g., LiveJournal, Flickr, Youtube) [32] in terms of important graph properties, such as the degree distribution, largest component size, edge density, and local clustering coefficient. Analyzing the tractability of MkTG on threshold graphs helps us understand the performance of the proposed algorithms on popular online social networks. We first define the threshold graph as follows.

Definition 5.1. A graph G is a threshold graph if there exists a weight $\widehat{w}_v$ for every vertex v in the graph and a real value τ (called the threshold value) such that for every edge (u, v), $\widehat{w}_v + \widehat{w}_u \geq \tau$ always holds.

Specifically, a threshold graph $G_C = (V_C, E_C)$ similar to online social networks can be constructed as follows [32]. For every vertex pair u, v, a larger weight is assigned to the vertex pair if u and v have more common or similar attributes (e.g., the number

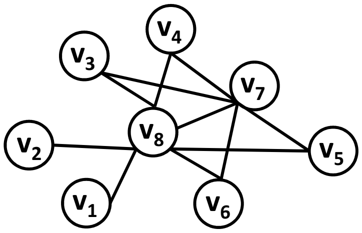


Figure 5: Running example of threshold graph.

of common neighbors). Then, given a threshold τ (not exceeding 10 in [32]), an edge (u, v) is constructed in E_C if the sum of vertex weights associated with u and v is at least τ. Therefore, it is not surprising that the threshold graph is similar to popular social networks because similar and close vertex pairs are inclined to be connected (the intuition is widely exploited in many link prediction algorithms, such as [36, 37]).

Example 5.2. Figure 5 presents an example of a threshold graph with $\tau = 6$, and vertex weights of $\{\widehat{w}_{v_1}, \dots, \widehat{w}_{v_8}\}$ are $\{2, 2, 3, 3, 3, 3, 7, 7\}$, respectively. The parameters of MkTG are $k = 2$ and $n = 4$. After SP and VPE strategies, we have $G_v = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and the vertices are all simplicial vertices. Please note that G_v satisfies the no-pair constraint, indicating that the solutions generated by TERA-ADV (or TERA) afterwards all satisfy the no-pair constraint. TERA-ADV then sets $H_1 = G_v$. H_1 is a feasible solution with objective value $\frac{\Delta_2(H_1)}{6} = 3.3$. Then, v_6 is removed from H_1 to construct H_2, a feasible solution with $\frac{\Delta_2(H_2)}{5} = 2$. Finally, v_5 is removed from H_2 to create H_3, which is a feasible solution with $\frac{\Delta_2(H_3)}{4} = 1$. H_3 is returned by TERA-ADV as the solution (which is the optimal solution).

In the following, the vicinal pre-order of a graph is *linear* if for any two vertices x, y in the graph, $x \lesssim y$, or $y \lesssim x$, or both. The following lemma in the literature first presents the one-to-one correspondence between the vicinal pre-order and a threshold graph.

Lemma 5.3. *[31] A graph G is a threshold graph if and only if the vicinal pre-order of G is linear.*

Lemma 5.3 indicates that the vertices in G to $\{v_1, \dots v_{|G|}\}$ can be relabelled such that $v_1 \lesssim v_2 \lesssim \cdots \lesssim v_{|G|}$. To show the solution optimality of TERA, we first explore the relation of vicinal pre-order and the shortest path distances in a threshold graph.

Lemma 5.4. *Given a threshold graph $G = (V, E)$, for any three vertices $x, y, z \in V$, if $x \lesssim y$, then $d_G(y, z) \leq d_G(x, z)$ holds.*

Proof. We prove this lemma by the following two cases. Case i): if y is on the shortest path from x to z, denoted as $P_G(x, z)$, then $d_G(y, z) < d_G(x, z)$ holds. Case ii): if y is not on $P_G(x, z)$, then there exists a vertex $v \in N(x)$ and $v \in P_G(x, z)$. Since $v \in N(x)$ implies $v \in N(y)$ (because $x \lesssim y$), $d_G(y, z) \leq d_G(x, z)$ holds because the path $P_G(y, z) = \{y\} \cup \{v\} \cup d_G(v, z)$ must have length no larger than that of $P_G(x, z) = \{x\} \cup \{v\} \cup d_G(v, z)$. The lemma follows. □

Given a threshold graph $G = (V, E)$ and three arbitrary subgraphs of G, denoted as $S = (V_S, E_S)$, $B = (V_B, E_B)$, and $C = (V_C, E_C)$, where $V_B = \{b_1, .., b_r\}$, $V_C = \{c_1, .., c_r\}$. Also, $V_B \cap V_C = \varnothing$. Let $\Delta_k(S)$ denote the number of k-triangles in S. Based on the above lemmas, the following lemma connects vicinal pre-order to the number of k-triangles in a threshold graph.

Lemma 5.5. *If $b_i, c_i \notin S$ and $c_i \lesssim b_i$ hold for every i in a threshold graph G, then $\Delta_k(S \cup C) \leq \Delta_k(S \cup B)$ for any k in MkTG.*

Proof. Given a k-triangle $\{x, y, z\}$ in $S \cup C$, let $\lambda = |\{x, y, z\} \cap S|$. We prove the lemma by considering all possible cases of λ as follows.

Case 1: $\lambda = 0$. Let $\{x, y, z\} = \{c_h, c_i, c_j\}$. since $c_h \lesssim b_h$, $c_i \lesssim b_i$, and $c_j \lesssim b_j$, based on Lemma 5.4, we have $d_G(b_h, b_i) \leq d_G(b_h, c_i)$ (because $c_i \lesssim b_i$) and $d_G(b_h, c_i) \leq d_G(c_h, c_i)$ (because $c_h \lesssim b_h$). Therefore, $d_G(b_h, b_i) \leq d_G(c_h, c_i)$ holds. Similarly, after we substitute b_i with b_j and c_i with c_j, $d_G(b_h, b_j) \leq d_G(c_h, c_j)$. Again, if we substitute b_h with b_j and c_h with c_j, we have $d_G(b_i, b_j) \leq d_G(c_i, c_j)$.

Because $\{c_h, c_i, c_j\}$ is a k-triangle, the above inequality implies that each pair of vertices $s, t \in \{c_h, c_i, c_j\}$ must have their distance $d_G(s, t) \leq k$. According to the inequalities obtained above, we can conclude that for any pair of vertices $s', t' \in \{b_h, b_i, b_j\}$, $d_G(s', t') \leq k$ must also hold. For example, since $d_G(c_h, c_i) \leq k$ and $d_G(b_h, b_i) \leq d_G(c_h, c_i)$ (as proved above) hold, $d_G(b_h, b_i) \leq k$ holds. Therefore, $\{b_h, b_i, b_j\}$ is also a k-triangle in $S \cup B$.

Case 2: $\lambda = 1$. Let $\{x, y, z\} = \{s, c_i, c_j\}$ where, without loss of generality, we assume $\{s\} = \{s, c_i, c_j\} \cap S$. Based on Lemma 5.4, we have i) $d_G(b_i, s) \leq d_G(c_i, s) \leq k$ (because $c_i \lesssim b_i$), ii) $d_G(b_j, s) \leq d_G(c_j, s) \leq k$, and iii) $d_G(b_i, b_j) \leq d_G(c_i, b_j) \leq d_G(c_i, c_j) \leq k$. Therefore $\{s, b_i, b_j\}$ is a k-triangle in $S \cup B$.

Case 3: $\lambda = 2$. Let $\{x, y, z\} = \{s_1, s_2, c_i\}$ and $\{s_1, s_2\} = \{x, y, z\} \cap S$. Based on Lemma 5.4, we have i) $d_G(s_1, s_2) \leq k$, ii) $d_G(b_i, s_1) \leq d_G(c_i, s_1) \leq k$, and iii) $d_G(b_i, s_2) \leq d_G(c_j, s_2) \leq k$. Therefore $\{s_1, s_2, b_i\}$ is a k-triangle in $S \cup B$.

Case 4: $\lambda = 3$. let $\{x, y, z\} = \{s_1, s_2, s_3\}$ and $\{s_1, s_2, s_3\} = \{x, y, z\} \cap S$, $\{s_1, s_2, s_3\}$ is a k-triangle in $S \cup B$.

In summary, there is a one-to-one mapping of k-triangles from $S \cup C$ to $S \cup B$. Therefore, $\Delta_k(S \cup C) \leq \Delta_k(S \cup B)$. The lemma follows. □

Equipped with the above lemmas, we now prove that TERA acquires the optimal solution of the MkTG in a threshold graph G.

Theorem 5.6. *Given a threshold graph $G = (V, E)$, TERA and TERA-ADV return the optimal solution of the MkTG problem on G.*

Proof. We first analyze TERA as follows. Given $G = (V, E)$, the size constraint n, and the tenuity parameter k of the MkTG problem, we denote H^* as the subgraph generated by TERA. Let H^{OPT} be an optimal solution and $|H^{OPT}| = m \geq n$. We relabel the vertices in G to $\{v_1, \dots v_{|G|}\}$ such that $v_1 \lesssim v_2 \lesssim \cdots \lesssim v_{|G|}$ according to Lemma 5.3, i.e., a vicinal pre-order is derived.

Suppose $v_i \lesssim v_j$. Given a k-triangle $\{v_i, a, b\}$, there are two cases. i) v_j is also a vertex of this k-triangle. ii) v_j is not a vertex of the k-triangle. In this case, since $d_G(a, b) \leq k$, $d_G(v_i, a) \leq k$, $d_G(v_i, b) \leq k$ (because $\{v_i, a, b\}$ is a k-triangle), while $d_G(v_j, a) \leq d_G(v_i, a)$ and $d_G(v_j, b) \leq d_G(v_i, b)$ hold (by Lemma 5.4), $\{v_j, a, b\}$ is a k-triangle. In summary, if $v_i \lesssim v_j$ holds, v_j must be involved in more k-triangles than v_i.

Note that TERA starts from G and then sequentially removes $v_{|G|}, v_{|G|-1}, ...,$ from G (TERA records the resulting subgraph once a vertex is removed) until there are n vertices left. The reason is that $v_{|G|} \gtrsim v_{|G|-1}$ and so on, indicating that $v_{|G|}$ is involved in more k-triangles than $v_{|G|-1}$ due to Lemma 5.5. Therefore, TERA will generate a subgraph $H^* = \{v_1, v_2, .., v_m\}$ in some iteration after removing $|V| - m$ vertices, and $|H^*| = |H^{OPT}| = m$.

In the following, we prove by contradiction that H^* is an optimal solution to MkTG according to Lemma 5.5. Suppose $|H^*| \neq |H^{OPT}|$ and let $H^{O \cap *} = H^{OPT} \cap H^*$, $H^{O-*} = H^{OPT} - H^*$, and $H^{*-O} = H^* - H^{OPT}$. Then $|H^{O-*}| = |H^{*-O}|$ holds since $|H^{OPT}| = |H^*|$. Moreover, for every $b \in H^{O-*}$, $c \in H^{*-O}$, we have $c \lesssim b$ because $\{v_1, v_2, .., v_m\}$ of H^* is the first m vertices in $v_1 \lesssim v_2 \lesssim .. \lesssim v_{|G|}$, and c is in H^*.

We first prove by contradiction that H^* follows the no-pair constraint. Suppose H^* does not follow the no-pair constraint. Then there exist $x, y \in H^*$ and $(x, y) \in E(G)$. Without loss of generality, we assume that $x \lesssim y$. We then consider every possible case of vertex x as follows.

Case 1: $x \in H^{*-O}$. We select two vertices $w, z \in H^{O-*}$ with $x \lesssim w$ and $y \lesssim z$. Since $(x, y) \in E(G)$ and $x \lesssim w$, $(w, y) \in E(G)$ holds by Lemma 5.4 because $d_G(w, y) \leq d_G(x, y)$. Since $(w, y) \in E(G)$ and $y \lesssim z$, $(w, z) \in E(G)$ holds because $d_G(w, z) \leq d_G(w, y)$, implying that the optimal solution H^{OPT} contains an edge and leads to a contradiction.

Case 2: $x \in H^{O \cap *}$ or $y \in H^{O \cap *}$. We select a vertex $z \in H^{O-*}$. Since $x \lesssim z$ and $y \lesssim z$. Similar to Case 1, we have $(x, z) \in E(G)$ and $(y, z) \in E(G)$. Therefore, for $x \in H^{O \cap *}$ or $y \in H^{O \cap *}$, the optimal solution H^{OPT} contains an edge and leads to a contradiction.

Based on the above two cases, H^* must satisfy the no-pair constraint. Also, since $|H^*| = |H^{OPT}| \geq n$ (i.e., the group size), H^* is a feasible solution.

Then, we prove that the number of k-triangles in H^* does not exceed the number of k-triangles in H^{OPT}. Our algorithm produces $H^* = \{v_1, v_2, ..., v_m\}$ with $v_1 \lesssim v_2 \lesssim ... \lesssim v_m$. Let $S = H^{OPT} \cap H^*$, $B = H^{OPT} - S$, $C = H^* - S$. Therefore, $H^{OPT} = S \cup B$, $H^* = S \cup C$, and $B \cap C = \emptyset$. Let $B = \{b_1, ..., b_{m-|S|}\}$ and $C = \{c_1, ..., c_{m-|S|}\}$. Since for each $b \in B$, the vertex ID of $b > m$, and for each $c \in C$, the vertex ID of $c \leq m$, $c_i \lesssim b_i$ for $i \in [1, m - |S|]$ holds. According to Lemma 5.5, we have $\Delta_k(H^*) \leq \Delta_k(H^{OPT})$. Since $|H^*| = |H^{OPT}|$ and H^{OPT} is an optimal solution, $\Delta_k(H^*)/|H^*| = \Delta_k(H^{OPT})/H^{OPT}$ holds, implying that H^* is also optimal.

The above analysis shows that if a feasible solution exists and thereby the optimal solution H^{OPT} also exists, our algorithm is able to obtain a feasible solution with the objective value identical to the one in H^{OPT}. Therefore, if TERA cannot find a subgraph which satisfies the no-pair constraint, it implies that the instance of MkTG problem on the threshold graph G has no feasible solution. For TERA-ADV, since SP and VPE strategies remove from G only the redundant vertices, TERA-ADV also finds the optimal solution for MkTG in threshold graphs. □

6 EXPERIMENTAL RESULTS

To evaluate the proposed algorithms, we conduct experiments on 5 real datasets. The first two datasets, *IG* and *FB*, are two social network datasets from Instagram and Facebook, respectively. FB contains 63K vertices and 817K edges [33], and IG includes 45K vertices and 678K edges [34]. The third dataset, DBLP, is a co-author network with 317K vertices and 1M edges[8]. The fourth dataset is the Pokec social network with 1.6M vertices and 30M edges[9]. Finally, Youtube video sharing dataset has 1.1M vertices and 3M edges, which is also employed to construct four threshold graphs with different thresholds $\tau = \{2, 4, 6, 8\}$ based on [32].

Table 1: Summary of datasets

Dataset	$\|V\|$	$\|E\|$	Avg. Deg.	$\frac{\Delta_1(G)}{\|V\|}$	CC	Diam
IG	45K	678K	15.1	132.6	0.24	7
FB	63K	817K	13	55.6	0.14	15
DBLP	317K	1M	3.2	7	0.63	21
Pokec	1.6M	30M	18.8	20	0.11	11
Youtube	1.1M	3M	2.6	2.7	0.08	24

Since no algorithm has been proposed for MkTG, we compare Triangle and Edge Reduction Algorithm (TERA) and TERA with Advanced Processing Strategies (TERA-ADV) with four baseline algorithms: Brute-Force (BF), Random (RND), BigClam [13] on the complement graph (BC), and Parallel Maximum Clique [38, 39] on the complement graph (PMC). BF finds the optimal solution of MkTG by enumerating all the subgraphs satisfying the no-pair constraint. RND selects random vertices from G to iteratively expand the subgraphs. It derives the tenuity objective value when a new vertex is added, and the best group following the no-pair constraint is returned.

BC is a community detection algorithm which detects overlapping communities by estimating non-negative latent factors. BC and PMC first construct the k-hop graph G_k of the original social network G and then transforms G_k into its complement graph $\widehat{G}_k$. We then employ BC to find the densest community in the complement graph, while PMC employs efficient heuristic approaches to find a clique of size n in the complement graph. The idea is to extract dense subgraphs (i.e., clique, community) in the complement graph and then return the corresponding tenuous subgraphs in the original social network[10].

In our experiments, the default parameters are $k = 2$ and $n = 20$. The algorithms are implemented on an HP DL580 server with Quadcore Intel X5450 3.0 GHz CPUs and 1TB RAM. Each result is averaged over 50 samples. The k-hop graphs for $2 \leq k \leq 5$ are constructed offline. Moreover, in TERA-ADV, the input graphs are filtered by Simplicial Pruning (SP) and Vicinal Partition and Elimination (VPE) offline as well.

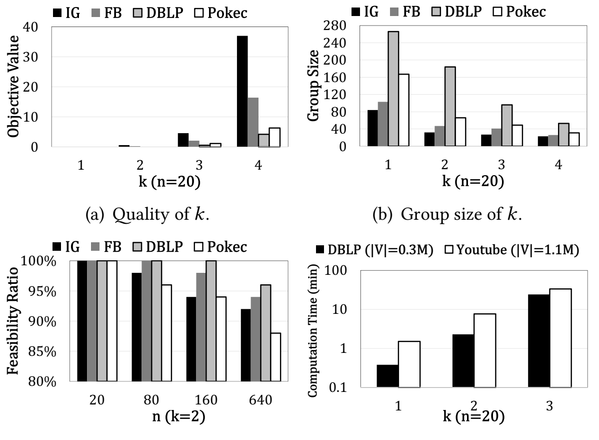
6.1 Sensitivity Tests on Large Graphs

Figure 6 reports the results of TERA-ADV on FB, IG, DBLP, Pokec, and Youtube to help understand the behavior in different datasets. Figure 6(a) compares the objective values in different datasets. No k-triangle is created when $k = 1$. As k increases, the objective values increase because more vertices in F are within k hops on

[8] http://snap.stanford.edu/data/com-DBLP.html.

[9] http://snap.stanford.edu/data/soc-pokec.html.

[10] If no clique of size n can be found, PMC extracts the maximum clique C and randomly chooses other vertices from G to argument C until $|C| = n$.



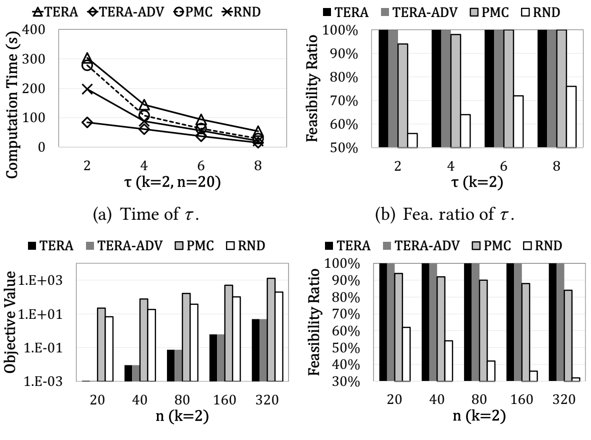
(a) Quality of k.

(b) Group size of k.

(c) Fea. ratio of n.

(d) Time of k.

Figure 6: Comparisons on large datasets.



(a) Time of τ.

(b) Fea. ratio of τ.

(c) Quality of n.

(d) Fea. ratio of n.

Figure 7: Comparisons on large threshold graphs (1.1M).

G. IG incurs the largest objective values because it is denser than others, i.e., the average number of 1-triangles is 132.6, and the average degree is 15. Figure 6(b) compares the group sizes obtained by TERA-ADV in different datasets. When k becomes large, small groups are preferred because larger groups in dense graphs tend to incur much more k-triangles. Moreover, TERA-ADV can find a larger group in DBLP without generating any k-triangle because the diameter is large, but the average degree is small in DBLP.

Figure 6(c) compares the feasibility ratios with different n (size constraint) in various datasets. Here, feasibility ratio is the ratio of the number of returned feasible solutions to the number of tested MkTG instances. For a small n, e.g., $n \leq 80$, the feasibility in each dataset is higher than 90%. TERA-ADV achieves the highest feasibility ratio in DBLP because DBLP has the smallest average degree, and TERA-ADV in this case tends to find large feasible groups easily. To understand the impact on computation time of different datasets, we compare the computation time of TERA-ADV in Figure 6(d) on DBLP (0.3M vertices) and Youtube (1.1M vertices). TERA-ADV is more efficient in DBLP because the number of vertices is only 1/3 that of Youtube. However, the computation time in the two datasets becomes closer when k increases. This is because the average degree and the clustering coefficient of Youtube is smaller than those of DBLP. In this case, SP and VPE can prune more redundant vertices in Youtube.

6.2 Comparisons on Large Threshold Graphs

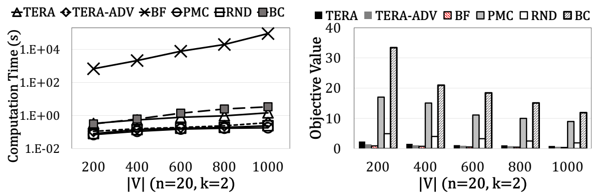
Figure 7 compares the performance of different approaches in threshold graphs (with threshold $\tau = \{2, 4, 6, 8\}$) constructed from Youtube (1.1M vertices). As shown in Figure 7(a), the computation time drops when τ increases because the graph contains fewer edges in this case. The improvement of TERA-ADV over TERA becomes more significant for a smaller τ, because the SP and VPE strategies trim off more vertices from the original graph. Figure 7(b) compares the feasibility ratios of different τ. When τ decreases, the feasibility ratios of PMC and RND drop due to the larger number of edges. Please note that PMC does not achieve 100% feasibility ratio because sometimes it cannot find a clique of size n in the complement graph. However, the feasibility ratios of the proposed algorithms are both 100% because as shown in Section 5, TERA and TERA-ADV can obtain the optimal solution of MkTG on threshold graphs.

Figures 7(c) and 7(d) examine the objective values and the feasibility ratios of different approaches. The objective values of the optimal solutions obtained by TERA and TERA-ADV grow when n increases because more vertices lead to a greater number of k-triangles. In contrast, both PMC and RND incur much larger objective values, and RND has poor feasibility ratios. This is because PMC and RND do not well utilize the information brought by the vicinal pre-order in threshold graphs.

6.3 Different Approaches on Small Graphs

We compare the proposed TERA and TERA-ADV with other baseline approaches. Note that BF does not scale up to large social networks because it examines all possible combinations. Therefore, we randomly sample the IG dataset to generate small networks with different sizes. Figures 8(a) and 8(b) compare the execution time and the solution quality for all algorithms in IG. Figure 8(a) manifests that, even for tiny networks, BF still incurs unacceptable computation time to find the optimal solution, whereas TERA is very efficient. Moreover, TERA-ADV, equipped with Simplicial Pruning (SP) and Vicinal Partition and Elimination (VPE) to remove redundant vertices, requires small running time that is comparable to RND.

Figure 8(b) presents the pruning power of the SP and VPE strategies in TERA-ADV, where TERA-ADV can obtain high-quality solutions very close to the optimal solutions. In contrast, PMC and BC perform poorly because they cannot effectively minimize k-triangles. Therefore, the results confirm that finding dense subgraphs on complement graphs does not work for the MkTG problem. TERA-ADV obtains solutions with better quality than TERA because the SP and VPE strategies effectively remove the redundant vertices, which sometimes are considered by TERA and thus deteriorate the solution quality of TERA.



(a) Time of $|V|$ (IG).

(b) Quality of $|V|$ (IG).

Figure 8: Comparisons of different approaches.

7 CONCLUSION

In contrast to previous works on identifying socially dense groups, research on finding socially tenuous groups has not received much attention in the research communities. This paper makes the first attempt to extract socially tenuous subgraphs from social networks. We introduce the notion of k-triangles for measuring group tenuity and formulate a new research problem, namely *Minimum k-Triangle Disconnected Group (MkTG)* that finds socially tenuous groups. We propose polynomial-time algorithms to obtain the optimal solutions for MkTG on threshold graphs, which are similar to many representative online social networks. We design two efficient and effective algorithms to solve MkTG on general graphs. Experimental results manifest that the proposed algorithms outperform baselines significantly in terms of computation time and solution quality. In the future work, we will incorporate other important dimensions, such as personal attributes, to ensure that the selected group members share similar or different attribute values to collaborate on the required tasks in various applications.

8 ACKNOWLEDGEMENTS

This work was supported in part by Ministry of Science and Technology (MOST), Taiwan, under MOST 105-2218-E-007 -032 -MY2, and MOST 105-2218-E-002 -035 -.

REFERENCES

[1] S. Seidman. Network structure and minimum degree. *Social Networks*, 1983.

[2] J. Cohen. Trusses: cohesive subgraphs for social network analysis, 2008.

[3] V. Batagelj and M. Zaversnik. Generalized cores. *arXiv:cs/0202039*, 2002.

[4] A. Goldberg. Finding a maximum density subgraph. *Technical Report*, 1984.

[5] D. Berlowitz, S. Cohen, B. Kimelfeld. Efficient enumeration of maximal k-plexes. *SIGMOD*, 2015.

[6] K. Reid. *Social work with groups*, 1997.

[7] J. Qiu, Z. Lin, C. Tang, and S. Qiao. Discovering organizational structure in dynamic social network. *ICDM*, 2009.

[8] S. Wasserman and K. Faust. Social network analysis: methods and applications. *Cambridge University Press*, 1994.

[9] Center for substance abuse treatment. Substance abuse treatment: group therapy. *Treatment improvement protocol (TIP) series 41*, 2005.

[10] On finding socially tenuous groups for online social networks (online full version). http://www.cs.nthu.edu.tw/~chihya/KDD2017/paper.pdf.

[11] U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, 2001.

[12] R. Mokken. Cliques, clubs and clans. *Quality and Quantity: International Journal of Methodology*, 1979.

[13] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. *WSDM*, 2013.

[14] J. Xie, S. Kelley, and B. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Survey*, 2013.

[15] Q. Zhu, H. Hu, J. Xu, and W.-C. Lee. Geo-social group queries with minimum acquaintance constraint. *arXiv:1406.7367v1*, 2014.

[16] C.-Y. Shen, D.-N. Yang, L.-H. Huang, W.-C. Lee, and M.-S. Chen. Socio-spatial group queries for impromptu activity planning. *TKDE*, 2016.

[17] H.-H. Shuai, D.-N. Yang, P. S. Yu, and M.-S. Chen. Willingness optimization for social group activity. *VLDB*, 2014.

[18] Y.-L. Chen, M.-S. Chen, and P. Yu. Ensemble of diverse sparsifications for link prediction in large-scale networks, *ICDM*, 2015.

[19] V. Satuluri, S. Parthasarathy, and Y. Ruan. Local graph sparsification for scalable clustering. *SIGMOD*, 2011.

[20] M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen. Sparsification of influence networks. *KDD*, 2011.

[21] N. Ruan, R. Jin, and Y. Huang. Distance preserving graph simplification. *ICDM*, 2011.

[22] P. Berman, S. Raskhodnikova, and G. Ruan. Finding sparser directed spanners. *FSTTCS*, 2010.

[23] M. Hujter and Z. Tuza. The number of maximal independent sets in triangle-free graphs. *SIAM Journal on Discrete Mathematics*, 1993.

[24] H. Hatami, J. Hladky, D. Kral, S. Norine, and A. Razborov. On the number of pentagons in triangle-free graphs. *Journal of Combinatorial Theory*, 2013.

[25] A. Grzesik. On the maximum number of five-cycles in a triangle-free graph. *Journal of Combinatorial Theory*, 2012.

[26] D. Brugmann, C. Komusiewicz, and H. Moser. On generating triangle-free graphs. *Electronic Notes in Discrete Mathematics*, 2009.

[27] M. Bougeret, N. Bousquet, R. Giroudeau, R. Watrigant. Parameterized complexity of the sparsest k-subgraph problem in chordal graphs. *SOFSEM*, 2014.

[28] A. Lee and I. Streinu. Pebble game algorithms and (k, l)-sparse graphs. *DMTCS*, 2005.

[29] R. Watrigant, M. Bougeret, and R. Giroudeau. The k-sparsest subgraph problem. *Tech. Rep.*, 2012.

[30] J. Cheng, Z. Shang, H. Cheng, H. Wang, and J. Yu. K-reach: who is in your small world. *VLDB*, 2012.

[31] N. Mahadev and U. Peled. Threshold graphs and related topics, New York, NY, USA: Elsevier, 1995.

[32] S. Saha, N. Ganguly, and A. Mukherjee. Intergroup networks as random threshold graphs. *Physical Review E*, 2014.

[33] B. Viswanath, A. Mislove, M. Cha, and K. Gummadi. On the evolution of user interaction in Facebook. *WOSN*, 2009.

[34] E. Ferrara, R. Interdonato, and A. Tagarelli. Online popularity and topical interests through the lens of Instagram. *HT*, 2014.

[35] U. Feige. Approximating maximum clique by removing subgraphs. *SIAM J. Discrete Math*, 2004.

[36] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. *Journal of the American Soceity for Information Science and Technology*, 2007.

[37] A. Clause, C. Moore, and M. Newman. Hierarchical structure and the prediction of missing links in network. *Nature*, 2008.

[38] R. A. Rossi, D. F. Gleich, A. H. Gebremedhin, and Md. M. A. Patwary. Fast maximum clique algorithms for large graphs. *WWW*, 2014.

[39] J. Xiang, C. Guo, and A. Aboulnaga. Scalable maximum clique computation using mapreduce. *ICDE*, 2013.