Top-k Overlapping Densest Subgraphs: Approximation and Complexity*

Riccardo Dondi[1], Mohammad Mehdi Hosseinzadeh[1], Giancarlo Mauri[2], and Italo Zoppis[2]

[1] Università degli Studi di Bergamo, Bergamo, Italy
[2] Università degli Studi di Milano-Bicocca, Milano - Italy
riccardo.dondi@unibg.it, m.hosseinzadeh@unibg.it, mauri@disco.unimib.it, zoppis@disco.unimib.it

Abstract. A central problem in graph mining is finding dense subgraphs, with several applications in different fields, a notable example being identifying communities. While a lot of effort has been put in the problem of finding a single dense subgraph, only recently the focus has been shifted to the problem of finding a set of densest subgraphs. An approach introduced to find possible overlapping subgraphs is the Top-k-Overlapping Densest Subgraphs problem. Given an integer $k \geq 1$ and a parameter $\lambda > 0$, the goal of this problem is to find a set of k densest subgraphs that may share some vertices. The objective function to be maximized takes into account the density of the subgraphs and the distance between subgraphs in the solution (multiplied by λ). The Top-k-Overlapping Densest Subgraphs problem has been shown to admit a $\frac{1}{10}$-factor approximation algorithm. Furthermore, the computational complexity of the problem has been left open. In this paper, we present contributions concerning the approximability and the computational complexity of the problem. For the approximability, we present approximation algorithms that improve the approximation factor to $\frac{1}{2}$, when k is smaller than the number of vertices in the graph, and to $\frac{2}{3}$, when k is a constant. For the computational complexity, we show that the problem is NP-hard even when $k = 3$.

1 Introduction

One of the most studied and central problems in graph mining is the identification of cohesive subgraphs. This problem has been raised in several contexts, from social network analysis [13] to finding functional motifs in biological networks [6]. Different definitions of cohesive graphs have been proposed and applied in literature. One of the most remarkable example is clique, and finding a maximum size clique is a well-known and studied problem in theoretical computer science [10]. Other interesting definitions of cohesive subgraph have been proposed in literature, for example *relaxed cliques* [1, 15, 12], which are graphs that satisfy a *relaxation* of some clique property. Notable examples of relaxed cliques are s-clubs,

t-cliques, k-core, and s-plex (for an overview of the different clique relaxations, see [12]).

Most of the definitions of cohesive subgraph lead to NP-hard problems, in some cases even hard to approximate. For example, finding a clique of maximum size in a graph $G = (V, E)$ is an NP-hard problem [10] and it is even hard to approximate within factor $O(|V|^{1-\varepsilon})$, for each $\varepsilon > 0$ [18]. A definition of cohesive subgraph that leads to a polynomial-time algorithm is that of average-degree density. For this problem, called Densest Subgraph, Goldberg gave an elegant polynomial-time algorithm [9], while a linear-time greedy algorithm that achieves an approximation factor of $\frac{1}{2}$ for Densest Subgraph has been given in [2, 4].

The Densest Subgraph problem aims at finding a single subgraph, but in many applications it is of interest to find a collection of dense subgraphs of a given graph. More precisely, it is interesting to compute a collection of distinct subgraphs having maximum density in a given graph. A recent approach proposed in [7] asks for a collection of top k densest, possibly overlapping, distinct subgraphs (denoted as Top-k-Overlapping Densest Subgraphs), since in many real-world cases dense subgraphs are related to non-disjoint communities. As pointed out in [14, 7], for example hubs are vertices that may be part of several communities and hence of several densest subgraphs, thus motivating the quest for overlapping distinct subgraphs. Top-k-Overlapping Densest Subgraphs, proposed in [7], addresses this problem by looking for a set of k subgraphs that maximize an objective function that takes into account both the density of the subgraphs and the distance between the subgraphs of the solution, thus allowing an overlap between the subgraphs which depends on a parameter λ. When λ is small, then the density plays a dominant role in the objective function, so the output subgraphs can share a significant part of vertices. On the other hand, if λ is large, then the subgraphs will share few or no vertices, so the subgraphs may be disjoint.

An approach similar to Top-k-Overlapping Densest Subgraphs was proposed in [3], where the goal is to find a set of k subgraphs of maximum density, such that the maximum pairwise Jaccard coefficient of the subgraphs is bounded. A dynamic variant of the problem, whose goal is finding a set of k disjoint subgraphs, has been recently considered in [16].

Other approaches related to Top-k-Overlapping Densest Subgraphs include covering or partitioning an input graph in dense subgraphs, like Minimum Clique Partition [8] or Minimum s-Club Covering [5]. However, notice that these approaches require that all the vertices of the graph belong to some dense subgraph of the solution, which is not the case for Top-k-Overlapping Densest Subgraphs.

Top-k-Overlapping Densest Subgraphs has been shown to be approximable within factor $\frac{1}{10}$ [7], while its computational complexity has been left open [7]. In this paper, we present algorithmic and complexity results for Top-k-Overlapping Densest Subgraphs when k is less than the number of vertices in the graph. This last assumption (required in Section 3) is reasonable, for example notice that in the experimental results presented in [7] k is equal to 20, even for graphs having thousands or millions of vertices. Concerning the approximation of the problem, we provide in Section 3 a $\frac{2}{3}$-approximation algorithm when k is a constant, and

we present a $\frac{1}{2}$-approximation algorithm when k is smaller than the size of the vertex set. From the computational complexity point of view, we show in Section 4 that Top-k Overlapping Densest Subgraphs is NP-hard even if $k = 3$ (that is we ask for three densest subgraphs), when $\lambda = 3|V|^3$, for an input graph $G = (V, E)$. We conclude the paper in Section 5 with some open problems. Some of the proofs and the pseudocode of some algorithms are omitted due to page limit.

2 Definitions

In this section, we present some definitions that will be useful in the rest of the paper. Moreover, we provide the formal definition of the problem we are interested in.

All the graphs we consider in this paper are undirected. Given a graph $G = (V, E)$ and a subset $U \subseteq V$, we denote by $E(U)$ the set of edges of G having both endpoints in U. Given a graph $G = (V, E)$, and a set $V' \subseteq V$, we denote by $G[V']$ the *subgraph* of G induced by V', formally $G[V'] = (V', E(V'))$. If $G[V']$ is a subgraph of $G[V'']$, with $V' \subseteq V'' \subseteq V$, then $G[V'']$ is a *supergraph* of $G[V']$. A subgraph $G[V']$ of G is a *singleton*, if $|V'| = 1$.

Moreover, given $V_1 \subseteq V$, $V_2 \subseteq V$, such that $V_1 \cap V_2 = \emptyset$, define $E(V_1, V_2) = \{\{u, v\} : u \in V_1, v \in V_2\}$, that is the set of edges having exactly one endpoint in V_1 and exactly one endpoint in V_2. Two subgraphs $G[V_1]$ and $G[V_2]$ of a graph $G = (V, E)$ are called *distinct* when $V_1 \neq V_2$.

Next, we present the definition of *crossing subgraphs*, which is fundamental in Section 3.2.

Definition 1. *Given a graph $G = (V, E)$, let $G[V_1]$ and $G[V_2]$ be two subgraphs of $G = (V, E)$. $G[V_1]$ and $G[V_2]$ are* crossing *when $V_1 \cap V_2 \neq \emptyset$, $V_1 \setminus V_2 \neq \emptyset$ and $V_2 \setminus V_1 \neq \emptyset$.*

Consider two crossing subgraphs $G[V_1]$ and $G[V_2]$ of G. Notice that $V_1 \nsubseteq V_2$ and $V_2 \nsubseteq V_1$. Consider the example of Fig. 1. The two subgraphs induced by $\{v_5, v_6, v_7, v_8, v_9, v_{10}\}$ and $\{v_1, v_2, v_3, v_4, v_5\}$ are crossing, while the two subgraphs induced by $\{v_5, v_6, v_7, v_8, v_9, v_{10}\}$ and $\{v_5, v_6, v_7, v_8, v_9\}$ are not crossing.

Now, we present the definition of density of a subgraph.

Definition 2. *Given a graph $G = (V, E)$ and a subgraph $G[V'] = (V', E(V'))$, with $V' \subseteq V$, the density of $G[V']$, denoted by $dens(G[V'])$, is defined as $dens(G[V']) = \frac{|E(V')|}{|V'|}$.*

A *densest subgraph* of a graph $G = (V, E)$ is a subgraph $G[U]$, with $U \subseteq V$, that maximizes $dens(G[U])$, among the subgraphs of G. In the example of Fig. 1 the subgraph induced by $\{v_5, v_6, v_7, v_8, v_9, v_{10}\}$ is the densest subgraph and has density $\frac{11}{6}$.

Given a graph $G = (V, E)$ and a set of subgraphs $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ where each $G[W_i]$ is a subgraph of G, that is $W_i \subseteq V$, with $1 \leq i \leq k$, then the density of $\mathcal{W}$, denoted by $dens(\mathcal{W})$, is defined as $dens(\mathcal{W}) = \sum_{i=1}^{k} dens(G[W_i])$.

The goal of the problem we are interested in is to find a set of k, with $1 \leq k < |V|$, possibly overlapping subgraphs having high density. In order to differentiate these k subgraphs, in [7] a distance function between subgraphs of the solution is included in the objective function (to be maximized). We present here the distance function between two subgraphs presented in [7].

Definition 3. *Given a graph $G = (V, E)$ and two subgraphs $G[U]$, $G[Z]$, with $U, Z \subseteq V$, define the distance function $d : 2^{G[V]} \times 2^{G[V]} \to \mathbb{R}_+$ between $G[U]$ and $G[Z]$ as follows:*

$$d(G[U], G[Z]) = \begin{cases} 2 - \frac{|U \cap Z|^2}{|U||Z|} & \text{if } U \neq Z, \\ 0 & \text{else.} \end{cases}$$

We prove an upper and a lower bound for the distance between two distinct subgraphs.

Lemma 1. *Let $G[U]$, $G[Z]$ be two distinct subgraphs of $G = (V, E)$. Then, it holds that $1 \leq d(G[U], G[Z]) \leq 2$.*

Now, we are able to define the problem we are interested in, introduced [7], where we add the constraint that $k < |V|$.

Problem 1. Top-k-Overlapping Densest Subgraphs
Input: A graph $G = (V, E)$, a parameter $\lambda > 0$.
Output: A set $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ of k subgraphs, with $1 \leq k < |V|$ and $W_i \subseteq V$, $1 \leq i \leq k$, that maximizes the following value

$$r(\mathcal{W}) = dens(\mathcal{W}) + \lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i], G[W_j]).$$

Notice that a solution $\mathcal{W}$ of Top-k-Overlapping Densest Subgraphs consists of k distinct subgraphs, since $\mathcal{W}$ is a set. We denote by (G, λ) an instance of Top-k-Overlapping Densest Subgraphs. Moreover, we assume in what follows that $|V| > 5$ (it is required in the proof of Lemma 5). Notice that, when $|V| \leq 5$, Top-k-Overlapping Densest Subgraphs can be solved optimally in constant time.

2.1 Goldberg's Algorithm and Extended Goldberg's Algorithm

Goldberg's Algorithm [9] computes in polynomial time an optimal solution for Densest-Subgraph. Goldberg's Algorithm reduces Densest-Subgraph to the problem of computing a minimum cut in a weighted auxiliary graph. The time complexity of Goldberg's Algorithm is $O(|V|^3)$ by applying flow algorithm [11].

Given a graph $G = (V, E)$ and a subgraph $G[V']$, with $V' \subseteq V$, we denote by *Densest-Subgraph*$(G[V'])$ a densest subgraph in $G[V']$, which can be computed with Goldberg's Algorithm.

In this paper, we consider also a modification of Goldberg's Algorithm given in [17]. We refer to this algorithm as the Extended Goldberg's Algorithm. Extended Goldberg's Algorithm [17] addresses a constrained variant of

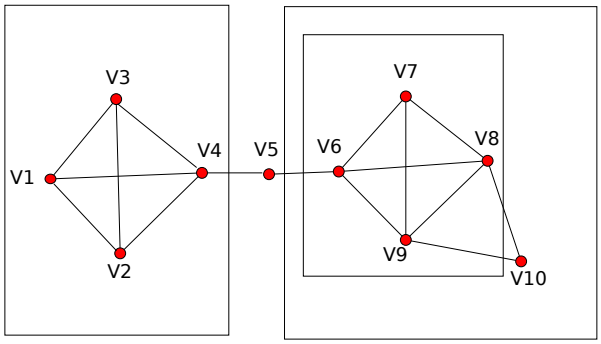

Fig. 1. A graph and a solution $\mathcal{W}$ of Top-k-Overlapping Densest Subgraphs, for $k = 3$, consisting of the three subgraphs included in boxes.

Densest-Subgraph, where some vertices are forced to be in a densest subgraph, that is we want to compute a densest subgraph $G[V']$ constrained to the fact that a set $S \subseteq V'$. We denote by $Densest\text{-}Subgraph(G[V'], C(S))$ a densest subgraph of $G[V']$ that is forced to contain S, where S is called the *constrained set* of $Dense\text{-}Subgraph(G[V'], C(S))$. Notice that $Densest\text{-}Subgraph(G[V'], C(S))$ can be computed with the Extended Goldberg's Algorithm in time $O(|V|^3)$ [17, 11].

3 Approximating Top-k-Overlapping Densest Subgraphs

In this section, we present a $\frac{2}{3}$-approximation algorithm for Top-k-Overlapping Densest Subgraphs when k is a constant and a $\frac{1}{2}$-approximation algorithm when k is not a constant. First, the two approximation algorithms compute a densest subgraph of G, denoted by $G[W_1]$. Then, they iteratively compute a solution for an intermediate problem, called Densest-Distinct-Subgraph. When k is constant we are able to solve the Densest-Distinct-Subgraph problem in polynomial time, while for general k we are able to provide a $\frac{1}{2}$-approximation algorithm for it.

First, we introduce the Densest-Distinct-Subgraph problem, then we present the two approximation algorithms and the analysis of their approximation factors.

Problem 2. Densest-Distinct-Subgraph
Input: A graph $G = (V, E)$ and a set $\mathcal{W} = \{G[W_1], \ldots, G[W_t]\}$, with $1 \leq t \leq k-1$, of subgraphs of G.
Output: A subgraph $G[Z]$ of G such that $Z \neq W_i$, for each $1 \leq i \leq t$, and $dens(G[Z])$ is maximum.

Notice that Densest-Distinct-Subgraph is not identical to compute a densest subgraph of G, as we need to ensure that the returned subgraph $G[Z]$ is distinct from any subgraph in $\mathcal{W}$. Moreover, notice that we assume that $|\mathcal{W}| \leq k-1$, since if $|\mathcal{W}| = k$ we already have k subgraphs in our solution of Top-k-Overlapping Densest Subgraphs.

3.1 Approximation for Constant k

First, we show that Densest-Distinct-Subgraph is polynomial-time solvable when k is a constant. The approximation algorithm for Top-k-Overlapping Densest Subgraphs returns the solution of maximum value between a solution obtained by iteratively solving Densest-Distinct-Subgraph and a solution consisting of k singletons.

A Polynomial-Time Algorithm for Densest-Distinct-Subgraph

We start by proving a property of solutions of Densest-Distinct-Subgraph.

Lemma 2. *Consider a graph $G = (V, E)$ and a set $\mathcal{W} = \{G[W_1], \ldots, G[W_t]\}$, $1 \leq t \leq k-1$, of subgraphs of G. Given a subgraph $G[Z]$ distinct from the subgraphs in $\mathcal{W}$, there exist t vertices $u_1, \ldots, u_t$, not necessarily distinct, with $u_i \in V$, $1 \leq i \leq t$, that can be partitioned into two sets U_1, U_2 such that $Z \supseteq U_1$, $Z \cap U_2 = \emptyset$ and there is no $G[W_j]$ in $\mathcal{W}$, with $1 \leq j \leq t$, such that $W_j \supseteq U_1$ and $W_j \cap U_2 = \emptyset$.*

Next, based on Lemma 2, we show how to compute an optimal solution of Densest-Distinct-Subgraph, when k is a constant. Algorithm 3 iterates over each subset U of at most t vertices (recall that $|\mathcal{W}| = t$) and over the subsets $U_1, U_2 \subseteq U$ such that $U_1 \uplus U_2 = U$. Algorithm 3 computes a densest subgraph $G[Z]$ of G, with constrained set U_1 and with $Z \cap U_2 = \emptyset$, such that there is no subgraph of $\mathcal{W}$ that contains U_1 and whose set of vertices is disjoint from U_2. Algorithm 3 applies the Extended Goldberg's algorithm on the subgraph $G[V \setminus U_2]$, with constrained set U_1. We prove the correctness of Algorithm 3 in the next theorem.

Theorem 1. *Let $G[Z]$ be the solution returned by Algorithm 3. Then $G[Z]$ is an optimal solution of* Densest-Distinct-Subgraph *over instance $(G, \mathcal{W})$.*

We recall that a densest subgraph constrained to a given set can be computed in time $O(|V|^3)$ with the Extended Goldberg's Algorithm [17, 11]. Algorithm 3 is iterared $k-1$ times, so the time complexity to find U is $O(|V|^{k-1})$, and for each U, the possible choices of U_1 and U_2 are $O(2^{k-1})$, which is a constant, since k is a constant. It follows that Algorithm 3 returns an optimal solution of Densest-Distinct-Subgraph in time $O(|V|^{k-1}|V|^3) = O(|V|^{k+2})$.

A $\frac{2}{3}$-Approximation Algorithm when k is a Constant

We show that, by solving the Densest-Distinct-Subgraph problem optimally, we achieve a $\frac{2}{3}$ approximation ratio for Top-k-Overlapping Densest Subgraphs. The approximation algorithm returns the solution of maximum value between the solution returned by Algorithm 1 and a solution consisting of k singletons.

First, we consider the solution returned by Algorithm 1. At each step, Algorithm 1 computes an optimal solution of Densest-Distinct-Subgraph in time

$O(|V|^{k+2})$ and the output subgraph is added to the solution. Since k is a constant, the number of iterations of Algorithm 1 is a constant, the overall time complexity of Algorithm 1 is $O(|V|^{k+2})$.

Algorithm 1: Algorithm that returns an approximate solution of Top-k-Overlapping Densest Subgraphs

Data: a graph G
Result: a set $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ of subgraphs of G
$\mathcal{W} \leftarrow \{G[W_1]\}$ /* $G[W_1]$ is a densest subgraph of G */;
for $i \leftarrow 2$ **to** k **do**
  Compute an optimal solution $G[Z]$ of Densest-Distinct-Subgraph with input $(G, \mathcal{W})$ /* Applying Algorithm 3 */;
  $\mathcal{W} \leftarrow \mathcal{W} \cup \{G[Z]\}$
Return($\mathcal{W}$);

Consider the solution $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ returned by Algorithm 1, we prove a bound on the objective value $r(\mathcal{W})$.

Lemma 3. *Let $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ be a set of subgraphs returned by Algorithm 1 and let $\mathcal{W}^o = \{G[W_1^o], \ldots, G[W_k^o]\}$ be an optimal solution of* Top-k-Overlapping Densest Subgraphs *over instance (G, λ). Then, it holds that $dens(\mathcal{W}) \geq dens(\mathcal{W}^o)$ and*

$$\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i], G[W_j]) \geq \frac{1}{2} \lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i^o], G[W_j^o]).$$

Proof. The second inequality follows from Lemma 1 and from the fact that the subgraphs in $\mathcal{W}$ are all distinct.

We prove the first inequality of the lemma by induction on k. Let $G[W_i]$, with $2 \leq i \leq k$, be the subgraph added to $\mathcal{W}$ by the i-th iteration of Algorithm 1. By construction, $dens(G[W_1]) \geq dens(G[W_2]) \geq \ldots \geq dens(G[W_k])$. Moreover, assume w.l.o.g. that $dens(G[W_1^o]) \geq dens(G[W_2^o]) \geq \ldots \geq dens(G[W_k^o])$.

When $k = 1$, by construction of Algorithm 1, $G[W_1]$ is a densest subgraph of G, it follows that $dens(G[W_1]) \geq dens(G[W_1^o])$. Assume that the lemma holds for $k-1$, we prove that it holds for k. Notice that $\sum_{i=1}^{k} dens(G[W_i]) = \sum_{i=1}^{k-1} dens(G[W_i]) + dens(G[W_k])$ and by induction hypothesis

$$\sum_{i=1}^{k-1} dens(G[W_i]) \geq \sum_{i=1}^{k-1} dens(G[W_i^o]).$$

Notice that $G[W_k]$ is an optimal solution of Densest-Distinct-Subgraph on instance $(G, \{G[W_1], G[W_2], \ldots, G[W_{k-1}]\})$. By the pigeon-hole principle at least one of the distinct subgraphs $G[W_1^o], G[W_2^o], \ldots, G[W_k^o]$ does not belong to the set $\{G[W_1], G[W_2], \ldots, G[W_{k-1}]\}$ of subgraphs, hence, by the optimality of $G[W_k]$, $dens(G[W_k]) \geq dens(G[W_p^o])$, for some p with $1 \leq p \leq k$, and $dens(G[W_p^o]) \geq$

$dens(G[W_k^o])$. Now,

$$\sum_{i=1}^{k} dens(G[W_i]) = \sum_{i=1}^{k-1} dens(G[W_i]) + dens(G[W_k]) \geq$$

$$\sum_{i=1}^{k-1} dens(G[W_i^o]) + dens(G[W_k^o]) \geq \sum_{i=1}^{k} dens(G[W_i^o])$$

thus concluding the proof. □

Consider Algorithm A_T that, given an instance (G, λ) of Top-k-Overlapping Densest Subgraphs, returns a solution $\mathcal{W}_T = \{G[W_{T,1}], \ldots, G[W_{T,k}]\}$ consisting of k distinct singletons. Notice that, since each $G[W_{T,i}]$, with $1 \leq i \leq k$, is a singleton, it follows that $dens(\mathcal{W}_T) = 0$. Moreover, since the subgraphs in $\mathcal{W}_T$ are pairwise disjoint, we have $d(G[W_{T,i}], G[W_{T,j}]) = 2$, for each $G[W_{T,i}]$, $G[W_{T,j}] \in \mathcal{W}_T$ with $1 \leq i \leq k$, $1 \leq j \leq k$ and $i \neq j$.

We can prove now that the maximum between $r(\mathcal{W})$ (where $\mathcal{W}$ is the solution returned by Algorithm 1) and $r(\mathcal{W}_T)$ (where $\mathcal{W}_T$ is the solution returned by Algorithm A_T) is at least $\frac{2}{3}$ of the value of an optimal solution of Top-k-Overlapping Densest Subgraphs.

Theorem 2. *Let $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ be a solution returned by Algorithm 1 and let $\mathcal{W}_T = \{G[W_{T,1}], \ldots, G[W_{T,k}]\}$ be a solution returned by Algorithm A_T. Let $\mathcal{W}^o = \{G[W_1^o], \ldots, G[W_k^o]\}$ be an optimal solution of* Top-k-Overlapping Densest Subgraphs *over instance (G, λ). Then* $\max(r(\mathcal{W}), r(\mathcal{W}_T)) \geq \frac{2}{3}\ r(\mathcal{W}^o)$.

3.2 Approximation When k is not a Constant

Now, we show that Top-k-Overlapping Densest Subgraphs can be approximated within factor $\frac{1}{2}$ when k is not a constant. The approximation algorithm (Algorithm 2), consists of two phases. In the first phase, while $\mathcal{W}$ does not contain crossing subgraphs (see Definition 1 of crossing subgraphs and Property 1 given later), Algorithm 2 adds to $\mathcal{W}$ a subgraph which is an optimal solution of Densest-Distinct-Subgraph. When Property 1 holds, Phase 2 of Algorithm 2 completes $\mathcal{W}$, by adding a set of subgraphs so that $\mathcal{W}$ contains k distinct subgraphs (see the description of Phase 2).

First, we define formally the property on which Algorithm 2 is based.

Property 1. Given a set $\mathcal{W}$ of t subgraphs, with $2 \leq t \leq k-1$, there exist two crossing subgraphs $G[W_i]$ and $G[W_j]$ in $\mathcal{W}$, with $1 \leq i \leq t$, $1 \leq j \leq t$ and $i \neq j$.

Description and Analysis of Phase 1

We show that, while $\mathcal{W}$ does not satisfy Property 1, Densest-Distinct-Subgraph can be solved optimally in polynomial time. First, we prove a property of a solution of Densest-Distinct-Subgraph when Property 1 does not hold.

Algorithm 2: Returns an approximate solution of Top-k-Overlapping Densest Subgraphs

Data: a graph G
Result: $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ of subgraphs of G
$\mathcal{W} \leftarrow \{G[W_1]\}$ /* $G[W_1]$ is a densest subgraph of G */;
Phase 1;
while $|\mathcal{W}| < k$ *and Property 1 does not hold* **do**
  Compute an optimal solution $G[Z]$ of Densest-Distinct-Subgraph with input $(G, \mathcal{W})$ /* Applying Algorithm 4 (described later) */;
  $\mathcal{W} \leftarrow \mathcal{W} \cup \{G[Z]\}$;
Phase 2 (Only if $|\mathcal{W}| < k$);
$W_{i,j} \leftarrow W_i \cap W_j$, with W_i and W_j two crossing subgraphs in $\mathcal{W}$;
if $|W_{i,j}| \leq 3$ **then**
  Complete $\mathcal{W}$ by adding the densest distinct subgraphs (not already in $\mathcal{W}$) induced by $W_i \cup \{v\}$, with $v \in V \setminus W_i$, and by $W_j \cup \{u\}$, with $u \in V \setminus W_j$;
if $|W_{i,j}| \geq 4$ **then**
  Complete $\mathcal{W}$ by adding the densest distinct subgraphs (not already in $\mathcal{W}$) induced by $W_i \cup \{v\}$, with $v \in V \setminus W_i$, by $W_j \cup \{u\}$, with $u \in V \setminus W_j$, and by $W_j \setminus \{w\}$, with $w \in W_{i,j}$;
Return($\mathcal{W}$);

Lemma 4. *Consider a graph $G = (V, E)$ and a set $\mathcal{W} = \{G[W_1], \ldots, G[W_t]\}$, $1 \leq t \leq k-1$, of subgraphs of G that does not satisfy Property 1. Given a subgraph $G[Z]$ distinct from the subgraphs in $\mathcal{W}$, there exist exactly three vertices $v_1, v_2, v_3 \in V$ that can be partitioned in two subsets U_1 and U_2 (U_2 possibly empty) such that $Z \supseteq U_1$, $Z \cap U_2 = \emptyset$ and there is no $G[W_j]$ in $\mathcal{W}$, $1 \leq j \leq t$, with $W_j \supseteq U_1$ and $W_j \cap U_2 = \emptyset$.*

Algorithm 4 computes an optimal solution $G[Z]$ of Densest-Distinct-Subgraph when Property 1 does not hold. Algorithm 4 is a modified variant of Algorithm 3 (see Section 3.1), which considers each set U of three vertices and each possible partition of U into U_1, U_2 (where U_2 can be empty). Based on Lemma 4, we can prove the following result.

Theorem 3. *Let $G[Z]$ be the solution returned by Algorithm 4. Then, an optimal solution of* Densest-Distinct-Subgraph *over instance $(G, \mathcal{W})$ when Property 1 does not hold has density at most $dens(G[Z])$.*

Algorithm 4 returns an optimal solution of Densest-Distinct-Subgraph when Property 1 does not hold in time $O(|V|^6)$, by applying the Extended Goldberg's Algorithm of complexity $O(|V|^3)$ [17, 11] for each subset of three vertices in V.

Description and Analysis of Phase 2

Assuming that Property 1 holds and $|\mathcal{W}| = t < k$, we consider Phase 2 of Algorithm 2. Given two crossing subgraphs $G[W_i]$ and $G[W_j]$ of $\mathcal{W}$, with

$1 \leq i \leq t$, $1 \leq j \leq t$ and $i \neq j$, define $W_{i,j} = W_i \cap W_j$. Algorithm 2 adds $h = k - t$ subgraphs to $\mathcal{W}$ until $|\mathcal{W}| = k$, as follows.

If $|W_{i,j}| \leq 3$, then Phase 2 of Algorithm 2 adds the h densest distinct subgraphs (not already in $\mathcal{W}$) induced by $W_i \cup \{v\}$, for some $v \in V \setminus W_i$, and by $W_j \cup \{u\}$, for some $u \in V \setminus W_j$.

If $|W_{i,j}| \geq 4$, then Phase 2 of Algorithm 2 adds the h densest distinct subgraphs (not already in $\mathcal{W}$) induced by $W_i \cup \{v\}$, for some $v \in V \setminus W_i$, by $W_j \cup \{u\}$, for some $u \in V \setminus W_j$, and by $W_j \setminus \{w\}$, for some $w \in W_{i,j}$ (or equivalently by $W_i \setminus \{w\}$, for some $w \in W_{i,j}$).

Next, we show that, after Phase 2 of Algorithm 2, $|\mathcal{W}| = k$ and the set $\mathcal{W}'$ of subgraphs added by Phase 2 has density at least $\frac{1}{2}|\mathcal{W}'|dens(G[W_j])$, where $G[W_j]$ is a subgraph added to $\mathcal{W}$ in Phase 1.

Lemma 5. *$|\mathcal{W}| = k$ after the execution of Phase 2 of Algorithm 2.*

Lemma 6. *Let $\mathcal{W}'$ be the set of subgraphs added to $\mathcal{W}$ by Phase 2 of Algorithm 2. Then, $dens(\mathcal{W}') \geq |\mathcal{W}'|\frac{1}{2}dens(G[W_j])$, with $G[W_j]$ a subgraph added to $\mathcal{W}$ by Phase 1 of Algorithm 2.*

Phase 2 of Algorithm 2 requires $O(k^2|V|)$ time, since we have to compare each subgraph to be added to $\mathcal{W}$ with the subgraphs already in $\mathcal{W}$ and each of this comparison requires $O(k|V|)$ time. Each iteration of Phase 1 of Algorithm 2 requires time $O(|V|^6)$, hence the overall complexity of Algorithm 2 is $O(|V|^7)$, since Phase 1 is iterated at most $k \leq |V| - 1$ times.

Now, thanks to Lemma 6, we are able to prove that the density of the solution returned by Algorithm 2 is at least half the density of an optimal solution of Top-k-Overlapping Densest Subgraphs.

Lemma 7. *Let $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ be the solution returned by Algorithm 2 and let $\mathcal{W}^o = \{G[W_1^o], \ldots, G[W_k^o]\}$ be an optimal solution of* Top-k-Overlapping Densest Subgraphs *over instance (G, λ). Then $\sum_{i=1}^{k} dens(G[W_i]) \geq \frac{1}{2}\sum_{i=1}^{k} dens(G[W_i^o])$.*

We can conclude the analysis of the approximation factor with the following result.

Theorem 4. *Let $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ be the solution returned by Algorithm 2 and let $\mathcal{W}^o = \{G[W_1^o], \ldots, G[W_k^o]\}$ be an optimal solution of* Top-k-Overlapping Densest Subgraphs *over instance (G, λ). Then $r(\mathcal{W}) \geq \frac{1}{2}r(\mathcal{W}^o)$.*

Proof. First, by Lemma 7, it holds that $dens(\mathcal{W}) \geq \frac{1}{2}dens(\mathcal{W}^o)$. Since the subgraphs in $\{G[W_1], \ldots, G[W_k]\}$ are all distinct, it holds from Lemma 1 that $d(G[W_i], G[W_j]) \geq 1$, for each i, j with $1 \leq i \leq k$, $1 \leq j \leq k$ and $i \neq j$, hence

$$\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i], G[W_j]) \geq \frac{1}{2}\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i^o], G[W_j^o]).$$

We can conclude that $r(\mathcal{W}) \geq \frac{1}{2}r(\mathcal{W}^o)$. □

4 Complexity of Top-k-Overlapping Densest Subgraphs

In this section, we consider the computational complexity of Top-k-Overlapping Densest Subgraphs and we show that the problem is NP-hard even if $k = 3$. We denote this restriction of the problem by Top-3-Overlapping Densest Subgraphs. We prove the result by giving a reduction from 3-Clique Partition, which is NP-complete [10]. Next, we recall the definition of 3-Clique Partition.

Problem 3. 3-Clique Partition
Input: A graph $G_P = (V_P, E_P)$.
Output: A partition of V_P into $V_{P,1}$, $V_{P,2}$, $V_{P,3}$ such that $V_P = V_{P,1} \uplus V_{P,2} \uplus V_{P,3}$ and each $G[V_{P,i}]$, with $1 \leq i \leq 3$, is a clique.

Given an instance $G_P = (V_P, E_P)$ of 3-Clique Partition, define an instance $(G = (V, E), \lambda)$ of Top-3-Overlapping Densest Subgraphs as follows: set $G = G_P$ and $\lambda = 3|V|^3$. In order to define a reduction from 3-Clique Partition to Top-3-Overlapping Densest Subgraphs, we show the following result.

Lemma 8. *Let $G_P = (V_P, E_P)$ be a graph instance of* 3-Clique Partition *and let $(G = (V, E), \lambda)$ be the corresponding instance of* Top-3-Overlapping Densest Subgraphs. *There exist three cliques $G_P[V_{P,1}]$, $G_P[V_{P,2}]$, $G_P[V_{P,3}]$ in G_P such that $V_{P,1}$, $V_{P,2}$, $V_{P,3}$ partition V_P if and only if there exists a set $\mathcal{W} = \{G[V_1], G[V_2], G[V_3]\}$ of subgraphs of G such that $r(\mathcal{W}) \geq \frac{|V|-3}{2} + 18|V|^3$.*

We can conclude that Top-3-Overlapping Densest Subgraphs is NP-hard.

Theorem 5. Top-3-Overlapping Densest Subgraphs *is NP-hard.*

5 Conclusion

We have shown that Top-k-Overlapping Densest Subgraphs is NP-hard when $k = 3$ and we have given two approximation algorithms of factor $\frac{2}{3}$ and $\frac{1}{2}$, when k is a constant and when k is smaller than the number of vertices in the graph, respectively. For future works, it would be interesting to further investigate the approximability of Top-k-Overlapping Densest Subgraphs, possibly improving the approximation factor or improving the time complexity of our approximation algorithms. A second interesting open problem is the computational complexity of Top-k-Overlapping Densest Subgraphs, in particular when λ is a constant and when the subgraphs in the solution overlap. Another open problem of theoretical interest is the computational complexity of Top-k-Overlapping Densest Subgraphs when $k = 2$.

References

1. Alba, R.D.: A graph-theoretic definition of a sociometric clique. Journal of Mathematical Sociology 3, 113–126 (1973)

2. Asahiro, Y., Iwama, K., Tamaki, H., Tokuyama, T.: Greedily finding a dense subgraph. In: Karlsson, R.G., Lingas, A. (eds.) Algorithm Theory - SWAT '96, 5th Scandinavian Workshop on Algorithm Theory, Reykjavík, Iceland, July 3-5, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1097, pp. 136–148. Springer (1996)
3. Balalau, O.D., Bonchi, F., Chan, T.H., Gullo, F., Sozio, M.: Finding subgraphs with maximum total density and limited overlap. In: Cheng, X., Li, H., Gabrilovich, E., Tang, J. (eds.) Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015. pp. 379–388. ACM (2015)
4. Charikar, M.: Greedy approximation algorithms for finding dense components in a graph. In: Jansen, K., Khuller, S. (eds.) Approximation Algorithms for Combinatorial Optimization, Third International Workshop, APPROX 2000, Proceedings. Lecture Notes in Computer Science, vol. 1913, pp. 84–95. Springer (2000)
5. Dondi, R., Mauri, G., Sikora, F., Zoppis, I.: Covering a graph with clubs. Journal of Graph Algorithms and Applications 23(2), 271–292 (2019)
6. Fratkin, E., Naughton, B.T., Brutlag, D.L., Batzoglou, S.: Motifcut: regulatory motifs finding with maximum density subgraphs. Bioinformatics 22(14), 156–157 (2006)
7. Galbrun, E., Gionis, A., Tatti, N.: Top-k overlapping densest subgraphs. Data Min. Knowl. Discov. 30(5), 1134–1165 (2016)
8. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. WH Freeman & Co. (1979)
9. Goldberg, A.V.: Finding a maximum density subgraph. Tech. rep., Berkeley, CA, USA (1984)
10. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Proceedings of a symposium on the Complexity of Computer Computations. pp. 85–103. The IBM Research Symposia Series, Plenum Press, New York (1972)
11. Kawase, Y., Miyauchi, A.: The densest subgraph problem with a convex/concave size function. Algorithmica 80(12), 3461–3480 (2018)
12. Komusiewicz, C.: Multivariate algorithmics for finding cohesive subnetworks. Algorithms 9(1), 21 (2016)
13. Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.: Trawling the web for emerging cyber-communities. Computer Networks 31(11-16), 1481–1493 (1999)
14. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Mathematics 6(1), 29–123 (2009)
15. Mokken, R.: Cliques, clubs and clans. Quality & Quantity: International Journal of Methodology 13(2), 161–173 (1979)
16. Nasir, M.A.U., Gionis, A., Morales, G.D.F., Girdzijauskas, S.: Fully dynamic algorithm for top-k densest subgraphs. In: Lim, E., Winslett, M., Sanderson, M., Fu, A.W., Sun, J., Culpepper, J.S., Lo, E., Ho, J.C., Donato, D., Agrawal, R., Zheng, Y., Castillo, C., Sun, A., Tseng, V.S., Li, C. (eds.) Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017. pp. 1817–1826. ACM (2017)
17. Zou, Z.: Polynomial-time algorithm for finding densest subgraphs in uncertain graphs. In: Proceedings of Internation Workshop on Mining and Learning with Graphs (2013)
18. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. Theory of Computing 3(1), 103–128 (2007)