Journal of Combinatorial Optimization (2021) 41:80–104
https://doi.org/10.1007/s10878-020-00664-3

Top-*k* overlapping densest subgraphs: approximation algorithms and computational complexity

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

Accepted: 16 October 2020 / Published online: 4 November 2020

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 dense subgraphs that may share some vertices. The objective function to be maximized takes into account the density of the subgraphs, the parameter λ and the distance between each pair of subgraphs in the solution. 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 $k = 3$.

A preliminary version of the paper appears in Dondi et al. (2019).

✉ Riccardo Dondi
riccardo.dondi@unibg.it

Mohammad Mehdi Hosseinzadeh
m.hosseinzadeh@unibg.it

Giancarlo Mauri
mauri@disco.unimib.it

Italo Zoppis
zoppis@disco.unimib.it

1 Università degli Studi di Bergamo, Bergamo, Italy

2 Università degli Studi di Milano-Bicocca, Milano, Italy

Keywords Graph mining · Graph algorithms · Densest subgraph · Approximation algorithms · Computational complexity

1 Introduction

Complex systems are usually analyzed with graphs. One of the most studied and central task to understand the behaviour of complex system is the identification of communities, that is cohesive subgraphs. This problem has been raised in several contexts, from social network analysis (Kumar et al. 1999) to finding functional motifs in biological networks (Fratkin et al. 2006). Different definitions of cohesive graphs have been proposed and applied in the literature. One of the most remarkable examples is Clique, and finding a maximum size clique is a well-known and studied problem in theoretical computer science (Karp 1972). Other interesting definitions of cohesive subgraphs have been proposed in the literature, for example *relaxed cliques* (Alba 1973; Mokken 1979; Komusiewicz 2016), which are graphs that satisfy a *relaxation* of some clique property, like the distance between vertices of the clique or the degree of the vertices of the clique. Notable examples of relaxed cliques are s-clubs, t-cliques, k-core, and s-plex [for an overview of the different clique relaxations, see Komusiewicz (2016)].

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 (Karp 1972) and it is even hard to approximate within factor $O(|V|^{1-\varepsilon})$, for each $\varepsilon > 0$ (Zuckerman 2007). Similarly, finding an s-club, with $s \geq 2$, of maximum size in a graph $G = (V, E)$ is an NP-hard problem (Bourjolly et al. 2002) which admits an approximation algorithm of factor $O(|V|^{1/2})$ (Asahiro et al. 2017), while it is not approximable within factor $O(|V|^{1/2-\varepsilon})$, for each $\varepsilon > 0$ (Asahiro et al. 2017). A definition of a dense 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 (Goldberg 1984), that requires $O(|V|^3)$ time (Kawase and Miyauchi 2018), while a linear-time greedy algorithm that achieves an approximation factor of $\frac{1}{2}$ for Densest Subgraph has been given in Asahiro et al. (1996) and Charikar (2000). A related problem, Densest k-Subgraph, is that of finding a densest subgraph with a constraint on the size of the subgraph. The problem becomes NP-hard, if it looks for a densest subgraph of a given size (Asahiro et al. 2002; Feige et al. 2001), of at most a given size (Andersen and Chellapilla 2009) or of at least a given size (Khuller and Saha 2009; Goldstein and Langberg 2009).

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 Galbrun et al. (2016) 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 Leskovec et al. (2009) and Galbrun et al. (2016), 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 Galbrun et al. (2016), 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, compared to the density, 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 compared to the density, 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 Balalau et al. (2015), where the goal is to find a set of k subgraphs of maximum density, with the constraint that the pairwise Jaccard coefficient (originally defined in Jaccard (1912)) between 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 Nasir et al. (2017).

Other approaches related to Top-k-Overlapping Densest Subgraphs include covering or partitioning an input graph in dense subgraphs, like Minimum Clique Partition (Garey and Johnson 1979) or Minimum s-Club Covering (Dondi et al. 2019). 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 $\frac{1}{10}$ (Galbrun et al. 2016), while its computational complexity has been left open (Galbrun et al. 2016). 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 Sect. 3) is reasonable, for example notice that in the experimental results presented in Galbrun et al. (2016) k is equal to 20, even for graphs having thousands or millions of vertices. Concerning the approximation of the problem, we provide in Sect. 3 a $\frac{2}{3}$-approximation algorithm when k is a constant (notice that the time complexity of this algorithm depends exponentially on k), and we present a $\frac{1}{2}$-approximation algorithm when $k < |V|$. From the computational complexity point of view, we show in Sect. 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 $G = (V, E)$. Notice that, since λ is large, the three subgraphs computed by the reduction are disjoint. The rest of the paper is organized as follows. In Sect. 2, we present some definitions and we give the formal definition of the Top-k-Overlapping Densest Subgraphs problem. In Sect. 3, we present the two approximation algorithms. In Sect. 4, we present the complexity result for Top-k-Overlapping Densest Subgraphs and we show that it is NP-hard even if $k = 3$, when $\lambda = 3|V|^3$.

We conclude the paper in Sect. 5 with some open problems.

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 set $V' \subseteq V$, we denote by $G[V'] = (V', E')$ the *subgraph* of G induced by V', where E' is defined as follows: $E' = \{\{u, v\} : \{u, v\} \in E \wedge u, v \in V'\}$. If $G[V']$ is a subgraph of $G[V'']$, with $V' \subseteq V''$, then $G[V'']$ is a *supergraph* of $G[V']$. A subgraph $G[V']$ of G is a *singleton*, if $|V'| = 1$.

Given a subset $U \subseteq V$, we denote by $E(U)$ the set of edges of G having both endpoints in U. Moreover, given $V_1 \subseteq V$, $V_2 \subseteq V$, such that $V_1 \cap V_2 = \emptyset$, define $E(V_1, V_2) = \{\{u, v\} : \{u, v\} \in E \wedge u \in V_1 \wedge v \in V_2\}$, 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 Sect. 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 $V_1 \subseteq V_2$ and $V_2 \nsubseteq V_1$ (see an example in Fig. 1).

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

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

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 k pairwise distinct 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 follows:

$$dens(\mathcal{W}) = \sum_{i=1}^{k} dens(G[W_i]).$$

The goal of the problem we are interested in is to find a set of k, with $1 \leq k < |V|$, pairwise distinct and possibly overlapping subgraphs having high density. In order to differentiate these k subgraphs, in Galbrun et al. (2016) a distance function between subgraphs of the solution is included in the objective function. The problem we consider maximizes an objective function that includes the sum of the densities of the subgraphs and the distances between subgraphs. We present here the distance function between two subgraphs introduced in Galbrun et al. (2016).

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

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

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 $1 \leq d(U, Z) \leq 2$.*

Proof By the definition of distance d, since $G[U]$ and $G[Z]$ are distinct subgraphs of G, it follows that $d(U, Z) = 2 - \frac{|U \cap Z|^2}{|U||Z|}$, where $0 \leq \frac{|U \cap Z|^2}{|U||Z|} \leq 1$. □

Now, we are able to define the problem we are interested in, introduced in Galbrun et al. (2016), 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 pairwise distinct subgraphs, with $1 \leq k < |V|$ and $W_i \subseteq V$, $1 \leq i \leq k$, that maximizes the following value

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

Notice that a solution $\mathcal{W}$ of Top-k-Overlapping Densest Subgraphs (see Fig. 1 for an example) 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 (Goldberg 1984) computes in polynomial time an optimal solution for Densest-Subgraph. Densest-Subgraph, given as input a graph $G = (V, E)$, asks for a subgraph $G[V']$ in G having maximum density. 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 (Kawase and Miyauchi 2018).

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

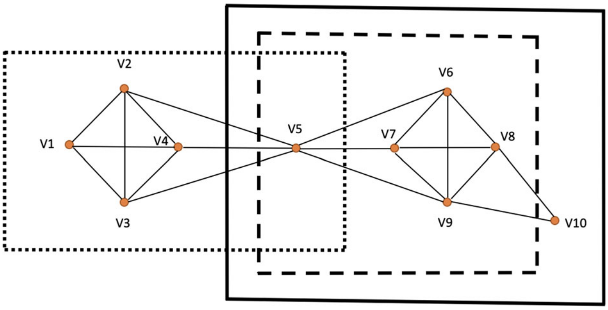

Fig. 1 A graph and a solution $\mathcal{W}$ of Top-k-Overlapping Densest Subgraphs, for $k = 3$, consisting of the three subgraphs included in boxes. Notice that the two subgraphs induced by $\{v_5, v_6, v_7, v_8, v_9, v_{10}\}$ and $\{v_1, v_2, v_3, v_4, v_5\}$ are crossing, while the two subgraphs 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

Goldberg's Algorithm in $O(|V|^3)$ time. Notice that *Densest-Subgraph*$(G[V])$ denotes a densest subgraph of G.

In the approximation algorithm, we will apply a modification of Goldberg's Algorithm given in Zou (2013). We refer to this algorithm as the Extended Goldberg's Algorithm. Extended Goldberg's Algorithm (Zou 2013) addresses a constrained variant of Densest-Subgraph, that, given as input a graph $G = (V, E)$ and a subset $S \subseteq V$, asks for a subgraph $G[V']$ in G having maximum density such that $S \subseteq V'$. We denote by $Densest\text{-}Subgraph(G[V_c], S)$ a densest subgraph of $G[V_c]$, with $V_c \subseteq V$, that is forced to contain S, where S is called the *constrained set* of $Dense\text{-}Subgraph(G[V_c'], S)$. Notice that *Densest-Subgraph*$(G[V'], S)$ can be computed with the Extended Goldberg's Algorithm in time $O(|V|^3)$ (Zou 2013; Kawase and Miyauchi 2018).

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, the two approximation algorithms 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 $1 \le t \le k-1$, of pairwise distinct subgraphs of G.

Output: A subgraph $G[Z]$ of G such that $Z \ne W_i$, for each $1 \le i \le t$, 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 $t \leq k-1$, since if $t = k$ 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 (see Algorithm 2) and a solution consisting of k singletons.

3.1.1 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 $W_j \cap U_2 = \emptyset$.*

Proof Consider $G[Z]$ and a subgraph $G[W_j]$, $1 \leq j \leq t$, in $\mathcal{W}$. Construct the sets U_1, U_2 as follows. First, set $U_1, U_2 = \emptyset$. For each j with $1 \leq j \leq t$, consider $G[W_j]$. Since $G[Z]$ is distinct from $G[W_j]$, it follows that: (1) there exists $u_j \in Z \setminus W_j$, then add u_j to U_1, or (2) there exists $u_j \in W_j \setminus Z$, then add u_j to U_2. By construction, the two sets U_1 and U_2 satisfy the lemma. □

Next, based on Lemma 2, we provide Algorithm 1 that computes an optimal solution of Densest-Distinct-Subgraph, when k is a constant. Algorithm 1 iterates over each subset U of at most t vertices (recall that $|\mathcal{W}| = t < k$) and over the subsets $U_1, U_2 \subseteq U$ such that $U_1 \uplus U_2 = U$. Algorithm 1 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 1 applies the Extended Goldberg's algorithm on the subgraph $G[V \setminus U_2]$, with constrained set U_1.

Algorithm 1: Returns an optimal solution for Densest-Distinct-Subgraph when k is a constant

Data: A graph G and a set $\mathcal{W} = \{G[W_1], \ldots, G[W_t]\}$ of subgraphs of G
Result: A subgraph $G[Z]$ of G, with $Z \neq W_i$, for each $1 \leq i \leq t$, and $dens(Z)$ is maximum

$Z = \emptyset$;
$dens = 0$;
for $U_1, U_2 \subseteq V$, *with* $U_1 \cap U_2 = \emptyset$, $|U_1 \cup U_2| \leq t$, *such that there is no subgraph* $G[W_i]$ *in* $\mathcal{W}$ *with* $W_i \supseteq U_1$ *and* $W_i \cap U_2 = \emptyset$ **do**
  $G[X] \leftarrow$ Densest-subgraph$(G[V \setminus U_2], U_1)$;
  $dens' \leftarrow dens(G[X])$;
  if $dens' > dens$ **then**
    $dens \leftarrow dens'$;
    $Z \leftarrow X$;
Return$(G[Z])$;

We prove the correctness of Algorithm 1 in the next theorem.

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

Proof Consider a set $\mathcal{W} = \{G[W_1], \ldots, G[W_t]\}$ of subgraphs of G and let $G[Z]$ be the solution returned by Algorithm 1. By Lemma 2 it follows that for each subgraph distinct from those in $\mathcal{W}$, hence also for an optimal solution $G[X]$ of Densest-Distinct-Subgraph over instance $(G, \mathcal{W})$, there exist t (non necessarily distinct) vertices $u_1, \ldots, u_t$, that can be partitioned into two sets U_1, U_2 such that $X \supseteq U_1$, $X \cap U_2 = \emptyset$ and there is no $G[W_j]$ in $\mathcal{W}$, with $1 \leq j \leq t$, such that $W_j \supseteq U_1$ and $W_j \cap U_2 = \emptyset$. The subgraph $G[Z]$ returned by Algorithm 1 is computed as a densest subgraph over each subset U of at most t vertices and for each partition of U into two sets U'_1 and 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 $W_j \cap U'_2 = \emptyset$. This holds also when $U'_1 = U_1$ and $U'_2 = U_2$, hence $dens(G[Z]) \geq dens(G[X])$. □

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 (Zou 2013; Kawase and Miyauchi 2018). The set U can be computed in $O(|V|^{k-1})$ time, by selecting t elements from V, since there are $|V|^t \leq |V|^{k-1}$ many of these subsets. 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 1 returns an optimal solution of Densest-Distinct-Subgraph in time $O(|V|^{k-1}|V|^3) = O(|V|^{k+2})$.

3.1.2 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 2 and a solution consisting of k singletons.

First, we consider the solution returned by Algorithm 2. At each step, Algorithm 2 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 2 is a constant, the overall time complexity of Algorithm 2 is $O(|V|^{k+2})$.

Algorithm 2: 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

```
W ← {G[W_1]} /* G[W_1] is a densest subgraph of G */;
for i ← 2 to k do
    Compute an optimal solution G[Z] of Densest-Distinct-Subgraph with
    input (G, W) /* Applying Algorithm 1 */;
    W ← W ∪ {G[Z]}
Return(W);
```

Consider the solution $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ returned by Algorithm 2, 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 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, it holds*

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

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 the number $h \leq k$ of subgraphs added to $\mathcal{W}$. Let $G[W_i]$, with $2 \leq i \leq h$, be the subgraph added to $\mathcal{W}$ by the i-th iteration of Algorithm 2. By construction, $dens(G[W_1]) \geq dens(G[W_2]) \geq \ldots \geq dens(G[W_h])$. Moreover, assume w.l.o.g. that $dens(G[W_1^o]) \geq dens(G[W_2^o]) \geq \ldots \geq dens(G[W_h^o])$.

When $h = 1$, by construction of Algorithm 2, $G[W_1]$ 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 $h - 1$, we prove that it holds for h. Notice that $\sum_{i=1}^{h} dens(G[W_i]) = \sum_{i=1}^{h-1} dens(G[W_i]) + dens(G[W_h])$ and by induction hypothesis

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

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

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

thus concluding the proof. □

Consider a trivial algorithm, called Algorithm A_T[1], 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 2) 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 2 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* $\max(r(\mathcal{W}), r(\mathcal{W}_T)) \geq \frac{2}{3}\, r(\mathcal{W}^o)$.

Proof By Lemma 3, it holds $dens(\mathcal{W}) \geq dens(\mathcal{W}^o)$. Moreover, by Lemma 1 it holds

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

Algorithm A_T returns solution $\mathcal{W}_T = \{G[W_{T,1}], \ldots, G[W_{T,k}]\}$ such that

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

First, assume that $\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i^o], G[W_j^o]) \geq 2\, dens(\mathcal{W}^o)$. Then

$$\frac{1}{3}\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_{T,i}], G[W_{T,j}]) \geq \frac{2}{3}\, dens(\mathcal{W}^o)$$

[1] The T in A_T means Trivial

thus,

$$\begin{aligned}
&\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_{T,i}], G[W_{T,j}]) \\
&\geq \frac{2}{3}\, \lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i^o], G[W_j^o]) + \frac{1}{3}\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_{T,i}], G[W_{T,j}]) \\
&\geq \frac{2}{3}\, \lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i^o], G[W_j^o]) + \frac{2}{3}\, dens(\mathcal{W}^o)
\end{aligned}$$

thus in this case A_T returns a solution having approximation factor $\frac{2}{3}$.

Second, assume that $\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(W_i^o, W_j^o) < 2\, dens(\mathcal{W}^o)$. It holds

$$\begin{aligned}
dens(\mathcal{W}) \geq dens(\mathcal{W}^o) &= \frac{2}{3}\, dens(\mathcal{W}^o) + \frac{1}{3}\, dens(\mathcal{W}^o) \\
&> \frac{2}{3}\, dens(\mathcal{W}^o) + \frac{1}{6}\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i^o], G[W_j^o]).
\end{aligned}$$

By Lemma 3

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

Since

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

we can conclude that

$$\begin{aligned}
r(\mathcal{W}) > \frac{2}{3}\, dens(\mathcal{W}^o) &+ \frac{1}{2}\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i^o], G[W_j^o]) \\
&+ \frac{1}{6}\lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d(G[W_i^o], G[W_j^o])
\end{aligned}$$

hence $r(\mathcal{W}) \geq \frac{2}{3}\, r(\mathcal{W}^o)$. □

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 3), consists of two phases. In the first phase, while $\mathcal{W}$ does not contain crossing subgraphs (see Definition 1 of crossing subgraphs), Algorithm 3 adds to $\mathcal{W}$ a subgraph which is an optimal solution of Densest-Distinct-Subgraph. When $\mathcal{W}$ contains crossing subgraphs (Property 1 holds), Phase 2 of Algorithm 3 completes $\mathcal{W}$, by adding a set of subgraph so that $\mathcal{W}$ contains k distinct subgraphs (see the description of Phase 2). We prove that the subgraphs added by Phase 2 are sufficiently dense (see Lemma 6). Notice that the subgraphs added by the algorithm are only distinct, that is a subgraph may be contained or have almost the same vertex set of another subgraph.

Algorithm 3: 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 */;
Phase 1;
while $|\mathcal{W}| < k$ *and* $\mathcal{W}$ *does not contain two crossing subgraphs* **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 $k - |\mathcal{W}|$ densest distinct subgraphs (not already in $\mathcal{W}$) induced by $W_i \cup \{v\}$, with $v \in V \backslash W_i$, and by $W_j \cup \{u\}$, with $u \in V \backslash W_j$;
if $|W_{i,j}| \geq 4$ **then**
  Complete $\mathcal{W}$ by adding the $k - |\mathcal{W}|$ densest distinct subgraphs (not already in $\mathcal{W}$) induced by $W_i \cup \{v\}$, with $v \in V \backslash W_i$, by $W_j \cup \{u\}$, with $u \in V \backslash W_j$, and by $W_j \backslash \{w\}$, with $w \in W_{i,j}$ (or equivalently by $W_i \backslash \{w\}$);
Return($\mathcal{W}$);

First, we define formally the property on which Algorithm 3 is based.

Property 1 $\mathcal{W}$ contains two crossing subgraphs.

3.2.1 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. We assume that a solution of Densest-Distinct-Subgraph contains at least two vertices, otherwise such a subgraph can be easily computed in polynomial time, since it consists of a single vertex and has density 0. First, we prove a property of a solution of Densest-Distinct-Subgraph when Property 1 does not hold.

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

Proof Consider a subgraph $G[Z]$ distinct from the subgraphs in $\mathcal{W}$ and a vertex $v_1 \in Z$. Set $U = \{v_1\}$. Notice that, for each subgraph in $\mathcal{W}$ that does not contain v_1, the lemma holds. Now, we consider the set $\mathcal{W}'$ of subgraphs in $\mathcal{W}$ that contain v_1, and we assume in the following that $\mathcal{W}' \neq \emptyset$.

Consider the pair $(\mathcal{W}', \subseteq)$, where $\subseteq$ is the subgraph inclusion relation[2]. $(\mathcal{W}', \subseteq)$ is a well-ordered set[3]. Clearly, $\subseteq$ is reflexive, antysimmetric and transitive on $\mathcal{W}'$. We show that $(\mathcal{W}', \subseteq)$ is comparable, that is, given $G[W_x], G[W_y] \in \mathcal{W}'$ with $W_x \neq W_y$, either $W_x \subset W_y$ or $W_y \subset W_x$. Indeed, consider two subgraphs $G[W_x], G[W_y] \in \mathcal{W}'$, such that neither $W_x \subset W_y$ nor $W_y \subset W_x$. It follows that they are crossing subgraphs, since they both contain v_1, contradicting the hypothesis that Property 1 does not hold. Since $\mathcal{W}'$ is a finite set, it follows that $(\mathcal{W}', \subseteq)$ is a well-ordered set.

Consider now the set $\mathcal{W}'_C$ of subgraphs in $\mathcal{W}'$ that are subgraphs of $G[Z]$ and notice that, since $(\mathcal{W}', \subseteq)$ is a well-ordered set, then also $(\mathcal{W}'_C, \subseteq)$ is a well-ordered set. Let $G[W_v]$ be the largest subgraph in $\mathcal{W}'_C$. Since $G[W_v]$ is a subgraph of $G[Z]$, there exists a vertex $v_2 \in Z \backslash W_v$. Since $(\mathcal{W}'_C, \subseteq)$ is a well-ordered set, each subgraph in $\mathcal{W}'_C \backslash \{G[W_v]\}$ is a subgraph of $G[W_v]$, thus each subgraph in $\mathcal{W}'_C$ does not contain v_2. Hence add v_2 to U and set $U_1 = \{v_1, v_2\}$. Notice that if $Z = V$ then the lemma holds, since each element in $\mathcal{W}'$ is a subgraph of $G[Z]$, hence it is in $\mathcal{W}'_C$.

Consider now the set $\mathcal{W}'_N$ of subgraphs in $\mathcal{W}'$ which are not subgraphs of $G[Z]$. Notice that $(\mathcal{W}'_N, \subseteq)$ is a well-ordered set and let $G[W_y]$ be the graph of minimum cardinality in $\mathcal{W}'_N$. It follows that there exists a vertex $v_3 \in W_y \backslash Z$, and notice that, since $(\mathcal{W}'_N, \subseteq)$ is a well-ordered set, v_3 belongs to each subgraph in $\mathcal{W}'_N$. Hence add v_3 to U and set $U_2 = \{v_3\}$.

Since we have shown that there exists $U_1 \subseteq Z$ that is not contained in any subgraph of $\mathcal{W}'_C$ and there exists $U_2 \nsubseteq Z$ that is contained in each subgraph of $\mathcal{W}'_N$, the lemma follows. □

[2] Given $A, B \subseteq V$, $G[A] \subseteq G[B]$ if and only if $A \subseteq B$

[3] We recall that a well-ordered set is a pair $(S, \le)$, where S is a set and $\le$ is a binary relation on S such that (1) Relation $\le$ satisfies the following properties: reflexivety, antisymmetry, transitivity and comparability; (2) every non-empty subset of S has a least element based on relation $\le$.

Algorithm 4: Returns an optimal solution for Densest-Distinct-Subgraph when Property 1 does not hold

Data: A graph G and a set $\mathcal{W} = \{G[W_1], \ldots, G[W_t]\}$, $1 \leq t \leq k-1$, of subgraphs of G, such that Property 1 does not hold

Result: A subgraph $G[Z]$ of G, with $Z \neq W_i$, for each $1 \leq i \leq t$, and $dens(Z)$ is maximum

$Z \leftarrow \emptyset$;
$dens \leftarrow 0$;
for *Each subset $U \subseteq V$ of at most three vertices, and each partition of U in U_1, U_2 where $U_1 \neq \emptyset$, such that there is no subgraph $G[W_i]$ in $\mathcal{W}$ with $U_1 \subseteq W_i$ and $U_2 \cap W_i = \emptyset$* **do**
  $G[X] \leftarrow$ Densest-subgraph$(G[V \setminus U_2], U_1)$;
  $dens' \leftarrow dens(G[X])$;
  if $dens' > dens$ **then**
    $dens \leftarrow dens'$;
    $Z \leftarrow X$;
Return $(G[Z])$;

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 1 (see Sect. 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 $dens(G[Z])$.*

Proof Given $(G, \mathcal{W})$, consider a subgraph $G[X]$ of maximal density distinct from the subgraphs in $\mathcal{W}$. By Lemma 4, it follows that there exists a set U of at most three vertices that can be partitioned into subsets U_1, U_2 such that $U_1 \subseteq X$ and $U_2 \cap X = \emptyset$ and there is no subgraph in $\mathcal{W}$ satisfying the same property. The subgraph $G[Z]$ returned by Algorithm 4 is computed as a densest subgraph over each subset U' of three vertices and each bipartition U'_1, U'_2 of U' such that $U'_1 \subseteq Z$ and $U'_2 \cap Z = \emptyset$ and there is no subgraph in $\mathcal{W}$ satisfying the same property. This holds also in the case $U'_i = U_i$, with $1 \leq i \leq 2$. It follows that $dens(G[Z]) \geq dens(G[X])$. □

Notice that Algorithm 4 returns an optimal solution of Densest-Distinct-Subgraph when Property 1 does not hold in time $O(|V|^6)$, since it applies the Extended Goldberg's Algorithm of complexity $O(|V|^3)$ (Zou 2013; Kawase and Miyauchi 2018) for each subset of three vertices in V.

3.2.2 Description and analysis of phase 2

Assuming that Property 1 holds and $|\mathcal{W}| = t < k$, we consider Phase 2 of Algorithm 3. 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 3 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 3 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 3 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 3, $|\mathcal{W}| = k$ and the set $\mathcal{W}'$ of subgraphs added by Phase 2 has density at least $\frac{1}{2}|\mathcal{W}'| \min(dens(G[W_i]), dens(G[W_j]))$ (recall that $G[W_i]$ and $G[W_i]$ are added in Phase 1).

We start by proving that, after Phase 2 of Algorithm 3, $|\mathcal{W}| = k$. Lemma 5 is based on the size of $W_{i,j} = W_i \cap W_j$. When $|W_{i,j}| \leq 3$, we distinguish two cases depending on the number of vertices that belong to $|W_i \setminus W_{i,j}|$ and $|W_j \setminus W_{i,j}|$. If one of these sets has at least two vertices, then there are enough subgraphs obtained by adding a vertex to W_i and W_j. In the other case (that is $|W_i \setminus W_{i,j}| = |W_j \setminus W_{i,j}| = 1$), then $|W_i \cup W_j| = 5$, thus there are $|V| - 5$ vertices that can be added to W_i and to W_j.

When $|W_{i,j}| \geq 4$, we can show that there are at least $|V \setminus W_{i,j}|$ subgraphs obtained by adding a vertex to W_i or to W_j. Then, we show that there are $|W_{i,j}|$ subgraphs induced by $W_j \setminus \{w\}$ (which are added by Phase 2).

Lemma 5 *$|\mathcal{W}| = k$ after the execution of Phase 2 of Algorithm 3.*

Proof Recall that we have assumed $|V| > 5$ and that $G[W_i]$ and $G[W_j]$ are two crossing subgraphs added in Phase 1 of Algorithm 3, with $W_{i,j} = W_i \cap W_j$. Next, we consider three cases depending on the size of $W_{i,j}$.

Consider the case that $|W_{i,j}| \leq 3$. If $|W_i \setminus W_{i,j}| \geq 2$ or $|W_j \setminus W_{i,j}| \geq 2$, then $W_i \cup \{v\}$, with $v \in V \setminus W_i$, and $W_j \cup \{u\}$, with $u \in V \setminus W_j$ induce distinct subgraphs. Hence there exist at least $|V| - 3$ distinct subgraphs induced by $W_i \cup \{v\}$, with $v \in V \setminus W_i$, or by $W_j \cup \{u\}$, with $u \in V \setminus W_j$. Since $G[W_i]$ and $G[W_j]$ are in $\mathcal{W}$ and $k \leq |V| - 1$, it follows that in this case k subgraphs belong to $\mathcal{W}$ after Phase 2 of Algorithm 3.

If both $|W_i \setminus W_{i,j}| = 1$ and $|W_j \setminus W_{i,j}| = 1$, then there exist one subgraph induced by $W_i \cup W_j$, since we have assumed that $|W_{i,j}| \leq 3$, at least $|V| - 5$ distinct subgraphs induced by $W_i \cup \{v\}$, with $v \in V \setminus (W_i \cup W_j)$, and at least $|V| - 5$ distinct subgraphs induced by $W_j \cup \{u\}$, with $u \in V \setminus (W_i \cup W_j)$. Since $|V| > 5$, it follows that at least $|V| - 5 + |V| - 5 + 1 \geq |V| - 5 + 2 \geq |V| - 3$ distinct subgraphs are induced by $W_i \cup \{v\}$, with $v \in V \setminus W_i$, or by $W_j \cup \{u\}$, with $u \in V \setminus W_j$. Since $G[W_i]$ and $G[W_j]$ are in $\mathcal{W}$ and $k \leq |V| - 1$, it follows that in this case k subgraphs belong to $\mathcal{W}$ after Phase 2 of Algorithm 3.

Consider now the case that $|W_{i,j}| \geq 4$. There exist at least $|V \setminus W_i|$ subgraphs induced by $W_i \cup \{v\}$, with $v \in V \setminus W_i$, and at least $|V \setminus W_j|$ subgraphs induced by $W_j \cup \{u\}$, with $u \in V \setminus W_j$. Hence there exist at least $|V \setminus W_{i,j}| - 1$ distinct subgraphs induced by $W_i \cup \{v\}$, with $v \in V \setminus W_i$, or by $W_j \cup \{u\}$, with $u \in V \setminus W_j$ (notice that the value -1 is due to the fact that $W_i \cup \{v\}$ and $W_j \cup \{u\}$ induce identical subgraphs when $W_i \setminus W_j = \{u\}$ and $W_j \setminus W_i = \{v\}$).

There exist at least $|W_{i,j}|$ subgraphs induced by $W_j \backslash \{w\}$, for some $w \in W_{i,j}$. Since $k \leq |V| - 1$, it follows that in this case k subgraphs belong to $\mathcal{W}$ after Phase 2 of Algorithm 3. □

Now, we show that the density of the set $\mathcal{W}'$ of subgraphs added by Phase 2 of Algorithm 3 is 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 6 *Let $\mathcal{W}'$ be the set of subgraphs added to $\mathcal{W}$ by Phase 2 of Algorithm 3. 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 3.*

Proof Consider $G[W_i]$ and $G[W_j]$, two crossing subgraphs added to $\mathcal{W}$ by Phase 1 of Algorithm 3, and $W_{i,j} = W_i \cap W_j$. Consider the case that $|W_{i,j}| \leq 3$. The density of a subgraph induced by $W' = W_j \cup \{u\}$, added by Phase 2 of Algorithm 3 can be bounded as follows:

$$\begin{aligned} dens(G[W']) &\geq \frac{|E(W_j)|}{|W_j|+1} = \frac{|E(W_j)|}{|W_j|}\frac{|W_j|}{|W_j|+1} = dens(W_j)\frac{|W_j|}{|W_j|+1} \\ &\geq \frac{1}{2}dens(G[W_j]) \end{aligned}$$

as $|W_j| \geq 1$.

Similarly, if $W' = W_i \cup \{u\}$ then

$$dens(G[W']) \geq \frac{1}{2}dens(G[W_i]).$$

Now, consider the case that $|W_{i,j}| \geq 4$. For a subgraph $G[W']$ added by Phase 2 of Algorithm 3 to $\mathcal{W}$ and induced by either $W_j \cup \{u\}$ or $W_i \cup \{v\}$, it holds the same argument of the case $|W_{i,j}| \leq 3$, thus, it holds

$$dens(G[W']) \geq \frac{1}{2}dens(G[W_j])$$

or

$$dens(G[W']) \geq \frac{1}{2}dens(G[W_i]).$$

Now, we consider the density of subgraphs $G[W']$, with $W' = W_j \backslash \{u\}$, where $u \in W_{i,j}$, added to $\mathcal{W}$ by Phase 2 of Algorithm 3. In order to show that $dens(G[W']) \geq \frac{1}{2}dens(G[W_j])$, with $W' = W_j \backslash \{u\}$, we show a bound on the sum over $u \in W_{i,j}$ of densities of the subgraphs $G[W']$. Since Algorithm 2 picks h densest of these subgraphs, it follows that the bound holds for the subgraphs added to $\mathcal{W}$ by Algorithm 2.

Consider the sum of the densities of the subgraphs $G[W']$ over the vertices $u \in W_{i,j}$:

$$\sum_{u \in W_{i,j}} dens(G[W_j \backslash \{u\}])$$

$$= \sum_{u \in W_{i,j}} \frac{1}{|W_j| - 1} \big(|E(W_j \backslash W_{i,j})| + |E(W_{i,j} \backslash \{u\})| \\ + |E(W_j \backslash W_{i,j}, W_{i,j} \backslash \{u\})|\big).$$

Each edge $\{v, w\}$, with $v, w \in W_{i,j}$, is skipped in the sum

$$\sum_{u \in W_{i,j}} \frac{1}{|W_j| - 1} |E(W_{i,j} \backslash \{u\})|$$

exactly twice, once for $u = v$ and once for $u = w$. It follows that

$$\sum_{u \in W_{i,j}} |E(W_{i,j} \backslash \{u\})| = (|W_{i,j}| - 2)|E(W_{i,j})|.$$

Each edge $\{w, v\}$, with $v \in W_{i,j}$ and $w \in W_j \backslash W_{i,j}$, is skipped in the sum

$$\sum_{u \in W_{i,j}} \frac{1}{|W_j| - 1} |E(W_j \backslash W_{i,j}, W_{i,j} \backslash \{u\})|$$

once, when $u = v$, thus

$$\sum_{u \in W_{i,j}} |E(W_j \backslash W_{i,j}, W_{i,j} \backslash \{u\})| = (|W_{i,j}| - 1)|E(W_j \backslash W_{i,j}, W_{i,j})|.$$

Thus

$$\begin{aligned} &\sum_{u \in W_{i,j}} dens(G[W_j \backslash \{u\}]) \\ &= \sum_{u \in W_{i,j}} \frac{1}{|W_j| - 1} \big(|E(W_j \backslash W_{i,j})| + |E(W_{i,j} \backslash \{u\})| \\ &\quad + |E(W_j \backslash W_{i,j}, W_{i,j} \backslash \{u\})|\big) \\ &= \frac{1}{|W_j| - 1} (|W_{i,j}||E(W_j \backslash W_{i,j})| + (|W_{i,j}| - 2)|E(W_{i,j})| \\ &\quad + (|W_{i,j}| - 1)|E(W_j \backslash W_{i,j}, W_{i,j})|) \\ &\geq \frac{|W_{i,j}| - 2}{|W_j| - 1} (|E(W_j \backslash W_{i,j})| + |E(W_{i,j})| + |E(W_j \backslash W_{i,j}, W_{i,j})|). \end{aligned}$$

Thus

$$\sum_{u \in W_{i,j}} dens(G[W_j \backslash \{u\}]) \geq \frac{|W_{i,j}| - 2}{|W_j| - 1} (|E(W_j \backslash W_{i,j})| + |E(W_{i,j})| \\ + |E(W_j \backslash W_{i,j}, W_{i,j})|)$$

$$\geq (|W_{i,j}| - 2)(dens(G[W_j]))$$

since

$$dens(G[W_j]) = \frac{1}{|W_j|}\left(|E(W_j \setminus W_{i,j})| + |E(W_{i,j})| + |E(W_j \setminus W_{i,j}, W_{i,j})|\right)$$
$$\leq \frac{1}{|W_j| - 1}\left(|E(W_j \setminus W_{i,j})| + |E(W_{i,j})| + |E(W_j \setminus W_{i,j}, W_{i,j})|\right).$$

It follows that

$$\sum_{u \in W_{i,j}} dens(G[W_j \setminus \{u\}]) \geq (|W_{i,j}| - 2)(dens(G[W_j]))$$
$$= \frac{(|W_{i,j}| - 2)}{|W_{i,j}|}|W_{i,j}|(dens(G[W_j])).$$

Since $|W_{i,j}| \geq 4$, it follows that $\frac{(|W_{i,j}|-2)}{|W_{i,j}|} \geq \frac{1}{2}$, thus

$$\sum_{u \in W_{i,j}} dens(G[W_j \setminus \{u\}]) \geq \frac{1}{2} \sum_{x \in W_{i,j}} dens(G[W_j])$$

since $\sum_{u \in W_{i,j}} dens(G[W_j]) = |W_{i,j}| dens(G[W_j])$.

Since Algorithm 3 adds the h most dense subgraphs among the choice of $u \in W_{i,j}$ so that $|\mathcal{W}| = k$, this completes the proof. □

Now, we consider the time complexity of Algorithm 3.

Lemma 7 *Algorithm 3 requires $O(|V|^7)$ time.*

Proof Phase 2 of Algorithm 3 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 3 requires time $O(|V|^6)$, hence the overall complexity of Algorithm 3 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 3 is at least half the density of an optimal solution of Top-k-Overlapping Densest Subgraphs.

Lemma 8 *Let $\mathcal{W} = \{G[W_1], \ldots, G[W_k]\}$ be the solution returned by Algorithm 3 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 $\sum_{i=1}^{k} dens(G[W_i]) \geq \frac{1}{2}\sum_{i=1}^{k} dens(G[W_i^o])$.*

Proof First, notice that we are assuming $dens(G[W_i^o]) \geq dens(G[W_j^o])$, when $1 \leq i < j \leq k$. Assume that Phase 1 of Algorithm 3 adds $k_1 \leq k$ subgraphs to $\mathcal{W}$.

We start by proving the following claim.

Claim 1 *For each h, with $1 \leq h < k$, given an optimal solution $G[W'_{h+1}]$ of Densest-Distinct-Subgraph over instance $(G, \{G[W_1], G[W_2], \ldots, G[W_h]\})$, it holds $dens(G[W'_{h+1}]) \geq dens(G[W^o_{h+1}])$.*

Proof Assume that this is not the case, and that $dens(G[W'_{h+1}]) < dens(G[W^o_{h+1}])$. Notice that at least one of $G[W^o_1], G[W^o_2], \ldots, G[W^o_{h+1}]$ does not belong to the set $\{G[W_1], G[W_2], \ldots, G[W_h]\}$ of subgraphs. Since the subgraphs are in non increasing order of density, it follows that an optimal solution of Densest-Distinct-Subgraph over instance $(G, \{G[W_1], G[W_2], \ldots, G[W_h]\})$ is a subgraph of G having density at least $dens(G[W^o_p])$, for some p with $1 \leq p \leq h+1$, and that $dens(G[W^o_p]) \geq dens(G[W^o_{h+1}]) > dens(G[W'_{h+1}])$, contradicting the optimality of $G[W'_{h+1}]$. ◻

We prove that the lemma holds for the subgraphs added by Phase 1 of Algorithm 3 by induction on k_1. When $k_1 = 1$, since $G[W_1]$ is a densest subgraph of G, it follows that $dens(G[W_1]) \geq \frac{1}{2} dens(G[W^o_1])$. Assume that the lemma holds for $h < k_1$, we prove that it holds for $h+1$.

Notice that

$$\sum_{i=1}^{k_1} dens(G[W_i]) = \sum_{i=1}^{h} dens(G[W_i]) + \sum_{i=h+1}^{k_1} dens(G[W_i]).$$

By induction hypothesis

$$\sum_{i=1}^{h} dens(G[W_i]) \geq \frac{1}{2} \sum_{i=1}^{h} dens(G[W^o_i]). \tag{1}$$

We prove that

$$\sum_{i=h+1}^{k_1} dens(G[W_i]) \geq \frac{1}{2} \sum_{i=h+1}^{k_1} dens(G[W^o_i]).$$

Consider subgraph $G[W_i]$, with $h+1 \leq i \leq k_1$, added to $\mathcal{W}$ by Phase 1 of Algorithm 3. By Claim 1 then $dens(G[W_i]) \geq dens(G[W^o_i])$, for each i with $h+1 \leq i \leq k_1$, thus

$$\sum_{i=h+1}^{k_1} dens(G[W_i]) \geq \frac{1}{2} \sum_{i=h+1}^{k_1} dens(G[W^o_i]). \tag{2}$$

Hence the lemma holds for the subgraphs added by Phase 1 of Algorithm 3.

Consider the subgraphs $G[W_{k_1+1}], \ldots, G[W_k]$ that are added to $\mathcal{W}$ by Phase 2 of Algorithm 3. By Lemma 6 it follows that there exists a subgraph $G[W_j]$ added to $\mathcal{W}$ by Phase 1 of Algorithm 3 such that $\sum_{i=k_1+1}^{k} dens(G[W_i]) \geq (k-k_1)\frac{1}{2} dens(G[W_j])$. Consider an optimal solution $G[W'_{k_1+1}]$ of Densest-Distinct-Subgraph over instance

$(G, \{G[W_1], G[W_2], \ldots, G[W_{k_1}]\})$. Since $G[W_j]$ is added to $\mathcal{W}$ by Phase 1 of Algorithm 3, by Theorem 3 it follows that $dens(G[W_j]) \geq G[W'_{k_1+1}]$. Hence,

$$\sum_{i=k_1+1}^{k} dens(G[W_i]) \geq \frac{1}{2}(k-k_1)dens(G[W'_{k_1+1}]).$$

Moreover, by Claim 1 it holds $dens(G[W'_{k_1+1}]) \geq dens(G[W^o_{k_1+1}])$. Hence it must hold

$$\sum_{i=k_1+1}^{k} dens(G[W_i]) \geq \frac{1}{2}(k-k_1)dens(G[W'_{k_1+1}]) \geq \frac{1}{2}(k-k_1)dens(G[W^o_{k_1+1}])$$

thus

$$\sum_{i=k_1+1}^{k} dens(G[W_i]) \geq \frac{1}{2}\sum_{i=k_1+1}^{k} dens(G[W^o_i]). \tag{3}$$

Combining Inequalities 2, 3, we obtain

$$\begin{aligned}\sum_{i=1}^{k} dens(G[W_i]) &= \sum_{i=1}^{k_1} dens(G[W_i]) + \sum_{i=k_1+1}^{k} dens(G[W_i]) \\ &\geq \frac{1}{2}\sum_{i=1}^{k_1} dens(G[W^o_i]) + \frac{1}{2}\sum_{i=k_1+1}^{k} dens(G[W^o_i]) \\ &\geq \frac{1}{2}\sum_{i=1}^{k} dens(G[W^o_i])\end{aligned}$$

thus concluding the proof. □

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 3 and let* $\mathcal{W}^o = \{G[W^o_1], \ldots, G[W^o_k]\}$ *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 8, it holds $dens(\mathcal{W}) \geq \frac{1}{2}dens(\mathcal{W}^o)$. 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

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

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 $k = 3$, when $\lambda = 3|V|^3$. We denote this restriction of the problem by Top-3-Overlapping Densest Subgraphs. Notice that our hardness result applies when λ is large ($\lambda = 3|V|^3$) and hence an optimal solution of Top-3-Overlapping Densest Subgraphs consists of three disjoint subgraphs.

We prove the result by giving a reduction from 3-Clique Partition, which is NP-complete (Karp 1972). 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 9 *Let $G_P = (V_P, E_P)$ be an 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 $r(\mathcal{W}) \geq \frac{|V|-3}{2} + 18|V|^3$.*

Proof We start by proving the first direction of the lemma. By construction the three subgraphs $G_P[V_{P,1}]$, $G_P[V_{P,2}]$, $G_P[V_{P,3}]$ of G_P are disjoint. Construct three subgraphs $G[V_1]$, $G[V_2]$, $G[V_3]$ of G as follows:

$$V_i = \{u_j \in V_i : v_j \in V_{P,i}\}$$

It follows that $G[V_1]$, $G[V_2]$, $G[V_3]$ are disjoint and that $V_1 \uplus V_2 \uplus V_3 = V$. Hence

$$r(\mathcal{W}) = dens(\mathcal{W}) + \lambda \sum_{i=1}^{2} \sum_{j=2}^{3} d(G[V_i], G[V_j])$$

where

$$dens(\mathcal{W}) = dens(G[V_1]) + dens(G[V_2]) + dens(G[V_3]) = \frac{|E_1|}{|V_1|} + \frac{|E_2|}{|V_2|} + \frac{|E_3|}{|V_3|}.$$

Since $G_P[V_{P,i}]$, with $1 \leq i \leq 3$, is a clique and, by construction, $G[V_i]$ is also a clique, it follows that

$$\frac{|E_i|}{|V_i|} = \frac{|V_i|(|V_i| - 1)}{2|V_i|} = \frac{|V_i| - 1}{2}$$

thus

$$
\begin{aligned}
& dens(G[V_1]) + dens(G[V_2]) + dens(G[V_3]) \\
& \quad = \frac{|V_1|-1}{2} + \frac{|V_2|-1}{2} + \frac{|V_3|-1}{2} = \frac{|V|-3}{2}.
\end{aligned}
$$

For each $i, j \in \{1, 2, 3\}$, with $i \neq j$, $G[V_i]$ and $G[V_j]$ are disjoint, hence:

$$
d(G[V_i], G[V_j]) = 2
$$

Thus, $r(\mathcal{W}) \geq \frac{|V|-3}{2} + 18|V|^3$.

Now, we prove the second direction of the lemma. First, notice that

$$
\lambda \sum_{i=1}^{2} \sum_{j=2}^{3} d(G[V_i], G[V_j]) \leq 18|V|^3
$$

since $d(G[V_i], G[V_j]) \leq 2$, for each i, j with $1 \leq i < j \leq 3$, and $\lambda = 3|V|^3$.

Next, we prove that $G[V_1], G[V_2], G[V_3]$ are disjoint and that $V_1 \uplus V_2 \uplus V_3 = V$. Assume to the contrary that two subgraphs in $\mathcal{W}$, w.l.o.g. $G[V_1]$ and $G[V_2]$, share at least one vertex. Then

$$
d(G[V_1], G[V_2]) = 2 - \frac{|V_1 \cap V_2|^2}{|V_1||V_2|} \leq 2 - \frac{1}{|V|^2}.
$$

Since $dens(\mathcal{W}) \leq \frac{3(|V|-1)}{2}$, as $\frac{|E_i|}{|V_i|} \leq \frac{|V_i|-1}{2} \leq \frac{|V|-1}{2}$. Moreover, since $\lambda = 3|V|^3$, it follows that

$$
\begin{aligned}
\lambda \sum_{i=1}^{2} \sum_{j=2}^{3} d(G[V_i], G[V_j]) &= \lambda(6 - \frac{3}{|V|^2}) = 4\lambda + \lambda \left(2 - \frac{3}{|V|^2}\right) \leq 4\lambda \\
&\quad + \lambda \left(2 - \frac{1}{|V|^2}\right) \\
&= 12|V|^3 + 3|V|^3 \left(2 - \frac{1}{|V|^2}\right).
\end{aligned}
$$

Hence

$$
r(\mathcal{W}) \leq 3\frac{|V|-1}{2} + 12|V|^3 + 3|V|^3 \left(2 - \frac{1}{|V|^2}\right) < 3|V| + 18|V|^3 - 3|V| = 18|V|^3.
$$

Thus $r(\mathcal{W}) < \frac{|V|-3}{2} + 18|V|^3$, contradicting the hypothesis that $r(\mathcal{W}) \geq \frac{|V|-3}{2} + 18|V|^3$. Thus we can assume that $G[V_1], G[V_2], G[V_3]$ are disjoint.

Now, we show that $V_1 \uplus V_2 \uplus V_3 = V$. Assume that this is not the case. Let $dens(G[V_i]) = z_i$, with $1 \leq i \leq 3$. Since $G[V_1], G[V_2], G[V_3]$ are disjoint, it follows that $z_1 + z_2 + z_3 < \frac{|V|-3}{2}$, since $|V_1| + |V_2| + |V_3| \leq |V|$ and $z_i \leq \frac{|V_i|-1}{2}$, with

$1 \leq i \leq 3$. Indeed notice that $z_i = \frac{|V_i|-1}{2}$ if and only if $G[V_i]$ is a clique. Thus $r(\mathcal{W}) < \frac{|V|-3}{2} + 18|V|^3$ contradicting the hypothesis that $r(\mathcal{W}) \geq \frac{|V|-3}{2} + 18|V|^3$. Moreover, notice that, since $r(\mathcal{W}) \geq \frac{|V|-3}{2} + 18|V|^3$, $dens(\mathcal{W}) \geq \frac{|V|-3}{2}$, thus each $G[V_i]$, with $1 \leq i \leq 3$, is a clique in G.

Now, we define $G_P[V_{P,1}]$, $G_P[V_{P,2}]$, $G_P[V_{P,3}]$:

$$V_{P,i} = \{v_j : u_j \in V_i\}.$$

By construction of G, it follows that $G[V_{P,1}]$, $G[V_{P,2}]$, $G[V_{P,3}]$ are disjoint, $V_{P,1} \uplus V_{P,2} \uplus V_{P,3} = V_P$ and that $G[V_{P,i}]$, with $1 \leq i \leq 3$, is a clique. □

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

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

Proof From Lemma 9, it follows that we have described a polynomial-time reduction from 3-Clique Partition to Top-3-Overlapping Densest Subgraphs. Since 3-Clique Partition is NP-complete (Karp 1972), it follows that also 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, it remains open whether the problem admits a polynomial-time approximation scheme. 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 $k = 2$.

Another direction is the investigation of the problem with other distance functions. The distance function we have considered has been introduced and applied in Galbrun et al. (2016) and, thanks to its properties (see Lemma 1), we were able to improve the constant-factor approximation of Top-k-Overlapping Densest Subgraphs, since it is enough to return distinct subgraphs. However, for other distance functions alternative algorithmic strategies may be needed to provide approximation algorithms. For example, one may consider the following distance function:

$$d(U, Z) = \begin{cases} 1 - \frac{|U \cap Z|^2}{|U||Z|} & \text{if } U \neq Z, \\ 0 & \text{else.} \end{cases}$$

Notice that Lemma 1 does not hold for this distance function, so the approximation results we have given cannot be applied.

Acknowledgements We thank the anonymous reviewers for their valuable comments on the paper.

Funding Open access funding provided by Universitá degli Studi di Bergamo within the CRUI-CARE Agreement.

References

Alba RD (1973) A graph-theoretic definition of a sociometric clique. J Math Sociol 3:113–126

Andersen R, Chellapilla K (2009) Finding dense subgraphs with size bounds. In: Avrachenkov, K., Donato, D., Litvak, N. (eds.) Algorithms and models for the web-graph, 6th international workshop, WAW 2009, Barcelona, Spain, February 12–13, 2009. Proceedings. Lecture notes in computer science, vol 5427. Springer, pp 25–37

Asahiro Y, Doiya Y, Miyano E, Samizo K, Shimizu H (2017) Optimal approximation algorithms for maximum distance-bounded subgraph problems. Algorithmica 80:1834–1856

Asahiro Y, Hassin R, Iwama K (2002) Complexity of finding dense subgraphs. Discrete Appl Math 121(1–3):15–26

Asahiro Y, Iwama K, Tamaki H, Tokuyama T (1996) Greedily finding a dense subgraph. In: Karlsson RG, 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. Springer, pp 136–148

Balalau OD, Bonchi F, Chan TH, Gullo F, Sozio M (2015) 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. ACM, pp 379–388

Bourjolly J, Laporte G, Pesant G (2002) An exact algorithm for the maximum k-club problem in an undirected graph. Eur J Oper Res 138(1):21–28

Charikar M (2000) 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. Springer, pp 84–95

Dondi R, Hosseinzadeh MM, Mauri G, Zoppis I (2019) Top-k overlapping densest subgraphs: approximation and complexity. In: Proceeding of ICTCS 2019 (to appear)

Dondi R, Mauri G, Sikora F, Zoppis I (2019) Covering a graph with clubs. J Graph Algorithms Appl 23(2):271–292

Feige U, Kortsarz G, Peleg D (2001) The dense k-subgraph problem. Algorithmica 29(3):410–421

Fratkin E, Naughton BT, Brutlag DL, Batzoglou S (2006) Motifcut: regulatory motifs finding with maximum density subgraphs. Bioinformatics 22(14):156–157

Galbrun E, Gionis A, Tatti N (2016) Top-k overlapping densest subgraphs. Data Min Knowl Discov 30(5):1134–1165

Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. WH Freeman & Co., London

Goldberg AV (1984) Finding a maximum density subgraph. Tech. rep, Berkeley, CA, USA

Goldstein D, Langberg M (2009) The dense k subgraph problem. CoRR abs/0912.5327

Jaccard P (1912) The distribution of the flora in the alpine zone. New Phytol 11(2):37–50

Karp RM (1972) Reducibility among combinatorial problems. In: Miller RE, Thatcher JW (eds) Proceedings of a symposium on the complexity of computer computations. The IBM research symposia series, Plenum Press, New York, pp 85–103

Kawase Y, Miyauchi A (2018) The densest subgraph problem with a convex/concave size function. Algorithmica 80(12):3461–3480

Khuller S, Saha B (2009) On finding dense subgraphs. In: Albers S, Marchetti-Spaccamela A, Matias Y, Nikoletseas SE, Thomas W (eds) Automata, languages and programming, 36th international colloquium, ICALP 2009, Rhodes, Greece, July 5–12, 2009, Proceedings, Part I. Lecture notes in computer science, vol 5555. Springer, pp 597–608

Komusiewicz C (2016) Multivariate algorithmics for finding cohesive subnetworks. Algorithms 9(1):21

Kumar R, Raghavan P, Rajagopalan S, Tomkins A (1999) Trawling the web for emerging cyber-communities. Comput Netw 31(11–16):1481–1493

Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Math 6(1):29–123

Mokken R (1979) Cliques, clubs and clans. Qual Quant Int J Methodol 13(2):161–173

Nasir MAU, Gionis A, Morales GDF, Girdzijauskas S (2017) Fully dynamic algorithm for top-k densest subgraphs. In: Lim E, Winslett M, Sanderson M, Fu AW, Sun J, Culpepper JS, Lo E, Ho JC, Donato D, Agrawal R, Zheng Y, Castillo C, Sun A, Tseng VS, Li C (eds) Proceedings of the 2017 ACM on conference on information and knowledge management, CIKM 2017. ACM, pp 1817–1826

Zou Z (2013) Polynomial-time algorithm for finding densest subgraphs in uncertain graphs. In: Proceedings of international workshop on mining and learning with graphs

Zuckerman D (2007) Linear degree extractors and the inapproximability of max clique and chromatic number. Theory Comput 3(1):103–128