# Finding Connected Dense $k$-Subgraphs

Xujin Chen(✉), Xiaodong Hu, and Changjun Wang

Institute of Applied Mathematics, AMSS,
Chinese Academy of Sciences, Beijing 100190, China
{xchen,xdhu,wcj}@amss.ac.cn

**Abstract.** Given a connected graph $G$ on $n$ vertices and a positive integer $k \leq n$, a subgraph of $G$ on $k$ vertices is called a $k$-subgraph in $G$. We design combinatorial approximation algorithms for finding a connected $k$-subgraph in $G$ such that its density is at least a factor $\Omega(\max\{n^{-2/5}, k^2/n^2\})$ of the density of the densest $k$-subgraph in $G$ (which is not necessarily connected). These particularly provide the first non-trivial approximations for the densest connected $k$-subgraph problem on general graphs.

**Keywords:** Densest $k$-subgraphs · Connectivity · Combinatorial approximation algorithms

## 1 Introduction

Let $G = (V, E)$ be a connected simple undirected graph with $n$ vertices, $m$ edges, and nonnegative edge weights. The (*weighted*) *density* of $G$ is defined as its average (weighted) degree. Let $k \leq n$ be a positive integer. A subgraph of $G$ is called a *$k$-subgraph* if it has exactly $k$ vertices. The *densest $k$-subgraph problem* (D$k$SP) is to find a $k$-subgraph of $G$ that has the maximum density, equivalently, a maximum number of edges. If the $k$-subgraph requires to be connected, then the problem is referred to as the *densest connected $k$-subgraph problem* (DC$k$SP). Both D$k$SP and DC$k$SP have their weighted generalizations, denoted respectively as H$k$SP and HC$k$SP, which ask for a heaviest (connected) $k$-subgraph, i.e., a (connected) $k$-subgraph with a maximum total edge weight. Identifying $k$-subgraphs with high densities is a useful primitive, which arises in diverse applications – from social networks, to protein interaction graphs, to the world wide web, etc. While dense subgraphs can give valuable information about interactions in these networks, the additional connectivity requirement turns out to be natural in various scenarios.

*Related Work.* An easy reduction from the maximum clique problem shows that D$k$SP, DC$k$SP and their weighted generalizations are all NP-hard in general. The NP-hardness remains even for some very restricted graph classes such as

Research supported in part by by NNSF of China under Grant No. 11222109, 11021161 and 10928102, by 973 Project of China under Grant No. 2011CB80800, and by CAS Program for Cross & Cooperative Team of Science & Technology Innovation.


R. Jain et al. (Eds.): TAMC 2015, LNCS 9076, pp. 248–259, 2015.
DOI: 10.1007/978-3-319-17142-5_22

chordal graphs, triangle-free graphs, comparability graphs and bipartite graphs of maximum degree three.

Most literature on finding dense subgraphs focus on the versions without requiring the subgraphs to be connected. For D$k$SP and its generalization H$k$SP, narrowing the large gap between the lower and upper bounds on the approachability is an important open problem.On the negative side, Feige [9] showed that computing a $(1+\varepsilon)$ approximation for D$k$SP is at least as hard as refuting random 3-SAT clauses for some $\varepsilon > 0$. Khot [15] showed that there does not exist any polynomial time approximation scheme (PTAS) for D$k$SP assuming NP does not have randomized algorithms that run in sub-exponential time. Recently, constant factor approximations in polynomial time for D$k$SP have been ruled out by Raghavendra and Steurel [20] under Unique Games with Small Set Expansion conjecture. On the positive side, considerable efforts have been devoted to finding good quality approximations for H$k$SP. Improving the $O(n^{0.3885})$-approximation of Kortsarz and Peleg [17], Feige et al. [11] proposed a combinatorial algorithm with approximation ratio $O(n^{\delta})$ for some $\delta < 1/3$. The latest algorithm of Bhaskara et al. [4] provides an $O(n^{1/4+\varepsilon})$-approximation in $n^{O(1/\varepsilon)}$ time. If allowed to run for $n^{O(\log n)}$ time, their algorithm guarantees an approximation ratio of $O(n^{1/4})$. The $O(n/k)$-approximation algorithm by Asahiro et al. [3] is remarkable for its simple greedy removal method. Linear and semidefinite programming relaxation approaches have been adopted in [10,13,21] to design randomized rounding algorithms.

For some special graph classes, better approximations have been obtained for D$k$SP and H$k$SP. Arora et al. [2] gave a PTAS for the restricted D$k$SP where $m = \Omega(n^2)$ and $k = \Omega(n)$, or each vertex of $G$ has degree $\Omega(n)$. Demaine et al. [8] developed a 2-approximation algorithm for D$k$SP on $H$-minor-free graphs, where $H$ is any given fixed graph. Chen et al. [5] showed that D$k$SP on a large family of intersection graphs admits constant factor approximations.

The work on approximating densest/heaviest connected $k$-subgraphs are relatively very limited. To the best of our knowledge, the existing polynomial time algorithms deal only with special graphical topologies, including: (a) 2-approximation for the metric H$k$SP (HC$k$SP) [14], where the underlying graph $G$ is complete, and the connectivity is trivial; (b) exact algorithms for H$k$SP and HC$k$SP on trees [7], for D$k$SP and DC$k$SP on $h$-trees, cographs and split graphs [7], and for DC$k$SP on interval graphs whose clique graphs are simple paths [19].

Among the well-known relaxations of D$k$SP and H$k$SP is the problem of finding a (connected) subgraph of maximum weighted density that does not have any cardinality constraint. It is strongly polynomial time solvable using max-flow based techniques [12,18]. Andersen and Chellapilla [1] and Khuller and Saha [16] studied two relaxed variants of H$k$SP for finding a weighted densest subgraph with at least or at most $k$ vertices. The former variant was shown to be NP-hard even in the unweighted case, and admit 2-approximation in the weighted setting. The approximation of the latter variant was proved to be as hard as that of D$k$SP/H$k$SP up to a constant factor.

*Our Results.* Given the interest in finding densest/heaviest connected $k$-subgraphs from both the theoretical and practical point of view, a better understanding of the problems is an important challenge for the field. In this paper, we design $O(mn \log n)$ time combinatorial approximation algorithms for finding a connected $k$-subgraph of $G$ whose density (weighted density) is at least a factor $\Omega(\max\{n^{-2/5}, k^2/n^2\})$ ($\Omega(\max\{1/k, k^2/n^2\})$) of the density (weighted density) of the densest (heaviest) $k$-subgraph of $G$ which is not necessarily connected. These particularly provide the first non-trivial approximation ratios for DC$k$SP and HC$k$SP on general graphs: $O(\min\{n^{2/5}, n^2/k^2\})$ for DC$k$SP and $O(\min\{k, n^2/k^2\})$ for HC$k$SP. Note that $\min\{k, n^2/k^2\} \le n^{2/3}$.

To evaluate the quality of our algorithms' performance guarantees $O(n^{2/5})$ and $O(n^{2/3})$, which are compared with the optimums of D$k$SP and H$k$SP, we investigate the maximum ratio $\Lambda$ (resp. $\Lambda_w$), over all graphs $G$ (resp. over all graphs $G$ and all nonnegative edge weights), between the maximum density (resp. weighted density) of *all* $k$-subgraphs and that of *all connected* $k$-subgraphs in $G$. The following examples show $\Lambda \ge \frac{1}{3}n^{1/3}$ and $\Lambda_w \ge \frac{1}{2}n^{1/2}$.

*Example 1.* (a) The graph $G$ is formed from $\ell$ vertex-disjoint $\ell$-cliques $L_1, \ldots, L_\ell$ by adding, for each $i = 1, \ldots, \ell - 1$, a path $P_i$ of length $\ell^2 + 1$ to connect $L_i$ and $L_{i+1}$, where $P_i$ intersects all the $\ell$ cliques only at a vertex in $L_i$ and a vertex in $L_{i+1}$. Let $k = \ell^2$. Note that $G$ has $n = \ell^2 + \ell^2(\ell - 1) = \ell^3$ vertices. The unique densest $k$-subgraph of $G$ is the disjoint union of $L_1, \ldots, L_\ell$ and has density $\ell - 1$. One of densest connected $k$-subgraphs of $G$ is induced by the $\ell$ vertices in $L_1$ and certain $\ell^2 - \ell$ vertices in $P_1$, and has density $(\ell(\ell-1) + 2(\ell^2 - \ell))/\ell^2$. Hence $\Lambda \ge \ell^2/(\ell + 2\ell) = \frac{1}{3}n^{1/3}$.

(b) The graph $G$ is a tree formed from a star on $\ell + 1$ vertices by dividing each edge into a path of length $\ell + 1$. All pendant edges have weight 1 and other edges have weight 0. Let $k = 2\ell$. Note that $G$ has $n = \ell^2 + 1$ vertices. The unique heaviest $k$-subgraph of $G$ is induced by the $\ell$ pendant edges of $G$, and has weighted density 1. Every heaviest connected $k$-subgraph of $G$ is a path containing exactly one pendant edge of $G$, and has weighted density $1/\ell$. Hence $\Lambda_w \ge \ell \ge \frac{1}{2}n^{1/2}$.

The remainder of this paper is organized as follows. Section 2 gives notations, definitions and basic properties necessary for our discussion. Section 3 is devoted to designing approximation algorithms for finding connected dense $k$-subgraphs. Section 4 discusses extension to the weighted case, and future research directions.The omitted details can be found in [6].

## 2 Preliminaries

Graphs studied in this paper are simple and undirected. For any graph $G' = (V', E')$ and any vertex $v \in V'$, we use $d_{G'}(v)$ to denote $v$'s degree in $G'$. The *density* $\sigma(G')$ of $G'$ refers to its average degree, i.e., $\sigma(G') = \sum_{v \in V'} d_{G'}(v)/|V'| = 2|E'|/|V'|$. Following convention, we define $|G'| = |V'|$. By a *component* of $G'$ we mean a maximal connected subgraph of $G'$.

Throughout let $G = (V, E)$ be a connected graph on $n$ vertices and $m$ edges, and let $k \in [3, n]$ be an integer. Our goal is to find a connected $k$-subgraph $C$ of $G$ such that its density $\sigma(C)$ is as large as possible. Let $\sigma^*(G)$ and $\sigma_k^*(G)$ denote the maximum densities of a subgraph and a $k$-subgraph of $G$, respectively, where the subgraphs are not necessarily connected. It is clear that

$$\sigma^*(G) \geq \sigma_k^*(G) \text{ and } n-1 \geq \sigma(G) \geq k \cdot \sigma_k^*(G)/n. \tag{2.1}$$

Let $S$ be a subset of $V$ or a subgraph of $G$. We use $G[S]$ to denote the subgraph of $G$ induced by the vertices in $S$, and use $G \setminus S$ to denote the graph obtained from $G$ by removing all vertices in $S$ and their incident edges. If $S$ consists of a single vertex $v$, we write $G \setminus v$ instead of $G \setminus \{v\}$.

The vertices whose removals increase the density of the graph play an important role in our algorithm design.

**Definition 1.** A vertex $v \in V$ is called *removable* in $G$ if $\sigma(G \setminus v) > \sigma(G)$.

Since $\sigma(G \setminus v) = 2(|E| - d_G(v))/(|V| - 1)$, the following lemma is straightforward. It also provides an efficient way for identifying removable vertices.

**Lemma 1.** *A vertex $v \in V$ is removable in $G$ if and only if $d_G(v) < \sigma(G)/2$.* □

**Lemma 2.** *Let $G_1$ be a connected $k$-subgraph of $G$. For any connected subgraph $G_2$ of $G_1$, it holds that $\sigma(G_1) \geq \sigma(G_2)/\sqrt{k}$.*

*Proof.* Suppose that $G_2$ is a $k_2$-subgraph of $G$ with $m_2$ edges. By the definition of density, $\sigma(G_2) \leq k_2 - 1$. The connectivity of $G_1$ implies $|E(G_1)| \geq |E(G_2)| + |V(G_1 \setminus G_2)|$, and

$$\sigma(G_1) \geq \frac{2(m_2 + k - k_2)}{k} = \frac{k_2 \cdot \sigma(G_2) + 2(k - k_2)}{k}.$$

In case of $k_2 \geq \sqrt{k}$, we have $\sigma(G_1) \geq k_2 \cdot \sigma(G_2)/k \geq \sigma(G_2)/\sqrt{k}$. In case of $k_2 < \sqrt{k}$, since $k \geq 3$, it follows that $G_1$ has no isolated vertices, and $\sigma(G_1) \geq 1 > k_2/\sqrt{k} > \sigma(G_2)/\sqrt{k}$. □

For a cut-vertex $v$ of $G$, we use $G_v$ to denote a densest component of $G \setminus v$, and use $G_{v+}$ to denote the connected subgraph of $G$ induced by $V(G_v) \cup \{v\}$. Note that $G \setminus G_v$ is a connected subgraph of $G$.

## 3 Algorithms

We design an $O(n^2/k^2)$-approximation algorithm (in Sect. 3.1) and further an $O(n^{2/5})$-approximation algorithm (in Sect. 3.2) for D$k$SP that always finds a connected $k$-subgraph of $G$. For ease of description we assume $k$ is even. The case of odd $k$ can be treated similarly. Alternatively, if $k$ is odd, we can first find a connected $(k-1)$-subgraph $G_1$ satisfying $\sigma_{k-1}^*(G)/\sigma(G_1) \leq O(\alpha)$, where $\alpha \in \{n^2/k^2, n^{2/5}\}$. Notice that $\sigma_k^*(G) \leq 3 \cdot \sigma_{k-1}^*(G)$ [6]. It follows that $\sigma_k^*(G)/\sigma(G_1) \leq O(\alpha)$. Then we attach an appropriate vertex to $G_1$, making a connected $k$-subgraph $G_2$ with density $\sigma(G_2) \geq \frac{k-1}{k}\sigma(G_1) \geq \frac{2}{3}\sigma(G_1)$. This guarantees that the approximation ratio is still $\sigma_k^*(G)/\sigma(G_2) \leq O(\alpha)$.

### 3.1 $O(n^2/k^2)$-Approximation

We first give an outline of our algorithm (see Algorithm 1) for finding a connected $k$-subgraph $C$ of $G$ with density $\sigma(C) \geq \Omega(k^2/n^2) \cdot \sigma_k^*(G)$ (see Theorem 1).

*Outline.* We start with a connected graph $G' \leftarrow G$ and repeatedly delete removable vertices from $G'$ to increase its density without destroying its connectivity.

– If we can reach $G'$ with $|G'| = k$ in this way, we output $C$ as the resulting $G'$.
– If we can find a removable cut-vertex $r$ in $G'$ such that $|G'_r| \geq k$, then we recurse with $G' \leftarrow G'_r$.
– If we stop at a $G'$ without any removable vertices, then we construct $C$ from an arbitrary connected $(k/2)$-subgraph by greedily attaching $k/2$ more vertices (see Procedure 1).
– If we are in none of the above three cases, we find a connected subgraph of $G'$ induced by a set $S$ of at most $k/2$ vertices, and then expand the subgraph in two ways: (1) attaching $G'_r$ for all removable vertices $r$ of $G'$ which are contained in $S$, and (2) greedily attaching no more than $k/2$ vertices. From the resulting connected subgraphs, we choose the one that has more edges (breaking ties arbitrarily), and further expand it to be a connected $k$-subgraph (see Procedure 2), which is returned as the output $C$.

*Greedy Attachment.* We describe how the greedy attaching mentioned in the above outline proceeds. Let $S$ and $T$ be disjoint nonempty vertex subsets (or subgraphs) of $G$. Note that $1 \leq |S| < n$. The set of edges of $G$ with one end in $S$ and the other in $T$ is written as $[S, T]$. For any positive integer $j \leq n - |S|$, a set $S^\star$ of $j$ vertices in $G \setminus S$ with *maximum* $|[S, S^\star]|$ can be found greedily by sorting the vertices in $G \setminus S$ as $v_1, v_2, \ldots, v_j, \ldots$ in a non-increasing order of the number of neighbors they have in $S$. For each $i = 1, 2 \ldots, j$, it can be guaranteed that $v_i$ has either a neighbor in $S$ or a neighbor in $\{v_1, v_2 \ldots, v_{i-1}\}$; in the latter case $i \geq 2$. Setting $S^\star = \{v_1, v_2, \ldots, v_j\}$. It is easy to see that

$$|[S, S^\star]| \geq \tfrac{j}{n} \cdot |[S, G \setminus S]|. \tag{3.1}$$

Moreover, if $G[S]$ is connected, the choices of $v_i$'s guarantee that $G[S \cup S^\star]$ is connected. We refer to this $S^\star$ as a *$j$-attachment* of $S$ in $G$. Given $S$, finding a $j$-attachment of $S$ takes $O(m+n \log n)$ time, which implies the following procedure runs in $O(|E(G')| + |G'| \cdot \log |G'|)$ time.

*Procedure 1.* Input: a connected graph $G'$ without removable vertices, where $|G'| > k$. Output: a connected $k$-subgraph of $G'$, written as $\text{PRC1}(G')$.

1. $G_1 = (V_1, E_1) \leftarrow$ an arbitrary connected $(k/2)$-subgraph of $G'$
2. $V_1^\star \leftarrow$ a $(k/2)$-attachment of $V_1$ in $G'$
3. Output $\text{PRC1}(G') \leftarrow G[V_1 \cup V_1^\star]$

Note that the definition of attachment guarantees that $V_1 \cap V_1^\star = \emptyset$, $|[V_1, V_1^\star]|$ is maximum, and $G[V_1 \cup V_1^\star]$ is connected.

**Lemma 3.** $\sigma(\textsc{Prc}1(G')) \geq \frac{k}{4|G'|} \cdot \sigma(G')$.

*Proof.* Since $G'$ has no removable vertices, we deduce from Lemma 1 that every vertex of $G'$ has degree at least $\sigma(G')/2$. Therefore $|[G_1, G' \setminus G_1]| \geq \frac{k}{2} \cdot \frac{\sigma(G')}{2} - 2|E_1|$. Recalling (3.1), we see that the number of edges in $\textsc{Prc}1(G')$ is at least $|[V_1, V_1^\star]| \geq (\frac{k \cdot \sigma(G')}{4} - 2|E_1|) \cdot \frac{k/2}{|G'|} + |E_1| \geq \frac{k^2}{8|G'|} \cdot \sigma(G')$, proving the lemma. □

*Procedure 2.* Input: a connected graph $G'$ with $|G'| > k$, where every removable vertex $r$ is a cut-vertex and satisfies $|G'_r| < k$. Output: a connected $k$-subgraph of $G'$, written as $\textsc{Prc}2(G')$.

---

1. $H \leftarrow G'$, $R' \leftarrow R =$ the set of removable vertices of $G'$
2. **While** $R' \neq \emptyset$ **do**
3. Take $r \in R'$
4. $H \leftarrow H \setminus V(G'_r)$, $R' \leftarrow R' \setminus V(G'_{r+})$
5. **End-While**
6. For each $v \in V(H)$, define $\theta(v) = |G'_{v+}|$ if $v \in R$, and $\theta(v) = 1$ otherwise
7. Let $S$ be a *minimal* subset of $V(H)$ s.t. $H[S]$ is connected & $\sum_{v \in S} \theta(v) \geq \frac{k}{2}$
8. Let $S^*$ be a $\min\{k/2, |H \setminus S|\}$-attachment of $S$ in $H$
9. $V_1 \leftarrow S \cup (\cup_{r \in R \cap S} V(G'_r))$, $V_2 \leftarrow S \cup S^\star$
10. Let $H'$ be one of $G'[V_1]$ and $G'[V_2]$ whichever has more edges (break ties arbitrarily)
11. Expand $H'$ to be a connected $k$-subgraph of $G'$
12. Output $\textsc{Prc}2(G') \leftarrow H'$

---

Under the condition that the resulting graph is connected, the expansion in Step 11 can be done in an arbitrary way. It is easy to see that Procedure 2 runs in $O(|G'| \cdot |E(G')|)$ time.

**Lemma 4.** *At the end of the while-loop (Step 5) in Procedure 2, we have*

*(i) $H$ is a connected subgraph of $G'$.*
*(ii) If $H$ contains two distinct vertices $r$ and $s$ that are removable in $G'$, then (by the condition of the procedure both $r$ and $s$ are cut-vertices of $G'$, and moreover) $G'_r$ and $G'_s$ are vertex-disjoint.*

*Proof.* Note that in every execution of the while-loop, $r \in R'$ is a cut-vertex of $H$, and $V(H) \cap V(G'_r)$ induces a component of $H \setminus r$. Thus $H$ is connected throughout the procedure. For any two removable vertices $r, s$ of $G'$ with $|G'_r| \leq |G'_s|$ and $r, s \in V(H)$, if $G'_r$ and $G'_s$ are not vertex-disjoint, then $V(G'_r) \cup \{r\} \subseteq V(G'_s)$. It follows that all vertices of $V(G'_r) \cup \{r\}$ have been removed by Step 4 delete when considering $s \in R'$, a contradiction. □

Observe that for any two distinct $r, s \in R$, either $G'_{r+}$ and $G'_{s+}$ are vertex-disjoint, or $G'_{r+}$ contains $G'_{s+}$, or $G'_{s+}$ contains $G'_{r+}$. This fact, along with an inductive argument, shows that, throughout Procedure 2, for any $s \in R \setminus V(H)$, there exists at least a vertex $r \in V(H) \cap R$ such that $G'_{r+}$ contains $G'_{s+}$, implying that $(U_{r\in R\cap V(H)}V(G_{r+}))\cup(V(H)\backslash R)=V(G')$ holds always. By Lemma 4(ii), in Step 7, we see that $V(G')$ is the disjoint union of $V(G_{r+})$, $r\in R\cap V(H)$ and $V(H)\backslash R$, giving $\sum_{v\in V(H)}\theta(v)=|G'|>k$. Hence, the connectivity of $H$ (Lemma 4 (i)) implies that the set $S$ in Step 7 does exist.

Take $u\in S$ such that $u$ is not a cut-vertex of $H$. If $|S|\geq(k/2)+1$, then we have $\sum_{v\in S\backslash\{u\}}\theta(v)\geq|S\backslash\{u\}|\geq k/2$, a contradiction to the minimality of $S$. Hence $|S|\leq k/2$.

Since Step 4 has removed from $H$ all vertices in $V(G'_r)$ for all $r\in R$, we see that $V_1$ is the disjoint union of $S$ and $\cup_{r\in R\cap S}V(G'_r)$ Recall that $|G'_r|<k$ for all $r\in R\cap S$. If $|V_1|>k$, then $|S|\geq 2$, and either $\theta_u\geq k/2$ or $\sum_{v\in S\backslash\{u\}}\theta(v)\geq k/2$, contradicting to the minimality of $S$. Noting that $|V_1|=\sum_{v\in S}\theta(v)$, we have

$$k/2\leq|V_1|\leq k. \tag{3.2}$$

We deduce that the output of Procedure 2 is indeed a connected $k$-subgraph of $G'$.

*Algorithm 1.* Input: connected graph $G=(V,E)$ with $|V|\geq k$.
Output: a connected $k$-subgraph of $G$, written as ALG1$(G)$.

1. $G'\leftarrow G$
2. **While** $|G'|>k$ and $G'$ has a removable vertex $r$ that is not a cut-vertex **do**
3. $\quad G'\leftarrow G'\setminus r$
4. **End-While** // either $|G'|=k$ or any removable vertex of $G'$ is a cut-vertex
5. **If** $|G'|=k$ **then** output ALG1$(G)\leftarrow G'$
6. **If** $|G'|>k$ and $G'$ has no removable vertices
   **then** output ALG1$(G)\leftarrow$ PRC1$(G')$
7. **If** $|G'|>k$ and $|G'_r|<k$ for each removable vertex $r$ of $G'$
   **then** output ALG1$(G)\leftarrow$ PRC2$(G')$
8. **If** $|G'|>k$ and $|G'_r|\geq k$ for some removable vertex $r$ of $G'$
   **then** output ALG1$(G)\leftarrow$ ALG1$(G'_r)$

In the while-loop, we repeatedly delete removable non-cut vertices from $G'$ until $|G'|=k$ or $G'$ has no removable non-cut vertex anymore. The deletion process keeps $G'$ connected, and its density $\sigma(G')$ increasing (cf. Definition 1). When the deletion process finishes, there are four possible cases, which are handled by Steps 5, 6, 7 and 8, respectively.

- In case of Step 5, the output $G'$ is clearly a connected $k$-subgraph of $G$.
- In case of Step 6, $G'$ qualifies to be an input of Procedure 1. With this input, Procedure 1 returns the connected $k$-subgraph PRC1$(G')$ of $G'$ as the algorithm's output.
- In case of Step 7, $G'$ qualifies to be an input of Procedure 2. With this input, Procedure 2 returns the connected $k$-subgraph PRC2$(G')$ of $G'$ as the algorithm's output.
- In case of Step 8, the algorithm recurses with smaller input $G'_r$, which satisfies $\sigma(G'_r)\geq\sigma(G')\geq\sigma(G)$ and $k\leq|G'_r|<|G'|\leq|G|$.

Hence after $O(n)$ recursions, the algorithm terminates at one of Steps $5-7$ and outputs a connected $k$-subgraph of $G$.

**Theorem 1.** *Algorithm 1 finds in $O(mn)$ time a connected $k$-subgraph $C$ of $G$ such that $\sigma_k^*(G)/\sigma(C) \le 12n^2/k^2$.*

*Proof.* Let $C = \text{Alg1}(G)$ be the output connected $k$-subgraph of $G$. If $C$ is output at Step 5, then its density is $\sigma(C) \ge \sigma(G) \ge (k/n) \cdot \sigma_k^*(G)$, where the last inequality is by (2.1). If $C$ is output by Procedure 1 at Step 6, then from Lemma 3 we know its density is at least $\frac{k}{4|G'|} \cdot \sigma(G') \ge \frac{k}{4n} \cdot \sigma(G) \ge \frac{k^2}{4n^2} \cdot \sigma_k^*(G)$.

Now we are only left with the case that $C = \text{Prc2}(G')$ is output by Procedure 2 at Step 7 of Algorithm 1. Let $R$ denote the set of removable vertices of $G'$. For every $r \in R$, we see that $r$ is a cut-vertex of $G'$ (cf. the note at Step 4 of the algorithm), and $\sigma(G'_r) \ge \sigma(G' \setminus r) > \sigma(G')$, where the first inequality is from the definition of $G'_r$ (it is the densest component of $G' \setminus r$), and the second inequality is due to the removability of $r$. Thus

$$\sigma(G'_{r+}) > \sigma(G'_r) \cdot |G'_r|/(|G'_r| + 1) \ge \sigma(G')/2 \;\text{ for every } r \in R.$$

Using the notations in Procedure 2, we note that each vertex of $S \setminus R$ is non-removable in $G'$, and therefore has degree at least $\sigma(G')/2$ in $G'$ by Lemma 1. Since $V_1 = S \cup (\cup_{r\in R\cap S} V(G'_r)) = (S \setminus R) \cup (\cup_{r\in S\cap R} V(G'_{r+}))$ contains at least $k/2$ vertices (recall (3.2)), it follows that $G'$ contains at least $(\frac{k}{2} \cdot \frac{\sigma(G')}{2})/2 \ge \frac{k}{8} \cdot \sigma(G) \ge \frac{k^2}{8n} \cdot \sigma_k^*(G)$ edges each with at least one end in $V_1$.

If there are at least $\frac{k^2}{24n} \cdot \sigma_k^*(G)$ edges with both ends in $V_1$, then by Step 10 of Procedure 2 we have $|E(C)| \ge \frac{k^2}{24n} \cdot \sigma_k^*(G)$ and $\sigma(C) = 2|E(C)|/k \ge \frac{k}{12n} \cdot \sigma_k^*(G) \ge \frac{k^2}{12n^2} \cdot \sigma_k^*(G)$. It remains to consider the case where $G'$ contains at least $\frac{k^2}{12n} \cdot \sigma_k^*(G)$ edges between $V_1$ and $G' \setminus V_1$. All these edges are between $S$ and $G' \setminus V_1 = H \setminus S$, since each edge incident with any vertex in $G'_r$ $(r \in R)$ must have both ends in $V_1$. So, by the definition of $S^\star$ at Step 8 of Procedure 2, we deduce from (3.1) that there are at least a number $|[S, S^\star]| \ge \frac{k/2}{n} \cdot |[S, H \setminus S]| \ge \frac{k^3}{24n^2} \cdot \sigma_k^*(G)$ of edges in the subgraph of $G'$ induced by $V_2 = S \cup S^\star$. Hence $\sigma(C) \ge 2|[S, S^\star]|/k \ge \frac{k^2}{12n^2} \cdot \sigma_k^*(G)$, justifying the performance of the algorithm. See [6] for the runtime analysis. □

### 3.2 $O(n^{2/5})$-Approximation

In this subsection we design algorithms for finding connected $k$-subgraphs of $G$ that jointly provide an $O(n^{2/5})$-approximation to D$k$SP. Among the outputs of all these algorithms (with input $G$), we select the densest one, denoted as $C$. Then it can be guaranteed that $\sigma_k^*(G)/\sigma(C) \le O(n^{2/5})$. In view of the $O(n^2/k^2)$-approximation of Algorithm 1, we may focus on the case of $k < n^{4/5}$. (Note that $n^2/k^2 \le n^{2/5}$ if $k \ge n^{4/5}$.)

Let $D$ be a densest connected subgraph of $G$, which is computable in time $O(mn \log(n^2/m))$ [12,18], because every component of a densest subgraph of $G$ is also a densest subgraph of $G$. Thus

$$\sigma(D) = \sigma^*(G) \ge \sigma_k^*(G).$$

Moreover, the maximality of $\sigma(D)$ implies that $D$ has no removable vertices.

*Algorithm 2.* Input: connected graph $G$ along with its densest connected subgraph $D$. Output: a connected $k$-subgraph of $G$, denoted as $\textsc{Alg2}(G)$.

---

1. **If** $|D| \leq k$ **then** Expand $D$ to be a connected $k$-subgraph $H$ of $G$
   Output $\textsc{Alg2}(G) \leftarrow H$
2. **Else** Output $\textsc{Alg2}(G) \leftarrow \textsc{Prc1}(D)$

---

**Lemma 5.** *If $k < n^{4/5}$, then $\sigma(\textsc{Alg2}(G)) \geq \min\{k/(4n), n^{-2/5}\} \cdot \sigma^*(G)$.*

*Proof.* In case of $|D| \leq k$, by Lemma 2, it follows from $\sigma^*(G) \geq \sigma^*_k(G)$ that the density of the output subgraph $\sigma(H) \geq \sigma(D)/\sqrt{k} = \sigma^*(G)/\sqrt{k}$. Since $k \leq n^{4/5}$, we see that $\sigma(H) \geq n^{-2/5} \cdot \sigma^*(G)$.

In case of $|D| > k$, we deduce from Lemma 3 that the connected $k$-subgraph $\textsc{Alg2}(G)$=$\textsc{Prc1}(D)$ of $D$ has density at least $\frac{k}{4|D|} \cdot \sigma(D) \geq \frac{k}{4n} \cdot \sigma^*(G)$. □

Our next algorithm is an expansion of Procedure 2 by Feige et al. [11]. Let $V_h$ be a set of $k/2$ vertices of highest degrees in $G$, and let $d_h = \frac{2}{k} \sum_{v \in V_h} d_G(v)$ denote the average degree of the vertices in $V_h$.

*Algorithm 3.* Input: connected graph $G$ with $|G| \geq k$.
Output: a connected $k$-subgraph of $G$, denoted as $\textsc{Alg3}(G)$.

---

1. $V_h^\star \leftarrow$ a $(k/2)$-attachment of $V_h$ in $G$
2. $H \leftarrow$ a densest component of $G[V_h \cup V_h^\star]$
3. Output $\textsc{Alg3}(G) \leftarrow$ a $k$-connected subgraph of $G$ that is expanded from $H$

---

In the above algorithm, the subgraph $G[V_h \cup V_h^\star]$ is exactly the output of Procedure 2 in [11], for which it has been shown (cf, Lemma 3.2 of [11]) that

$$\bar{\sigma} := \sigma(G[V_h \cup V_h^\star]) \geq kd_h/(2n).$$

Recalling Lemma 2, we have $\sigma(\textsc{Alg3}(G)) \geq \sigma(H)/\sqrt{k} \geq \bar{\sigma}/\sqrt{k}$, which implies the following result.

**Lemma 6.** $\sigma(\textsc{Alg3}(G)) \geq \frac{\bar{\sigma}}{\sqrt{k}} \geq \frac{\sqrt{k}}{2n} \cdot d_h$. □

Our last algorithm is a slight modification of Procedure 3 in [11], where we link things up via a "hub" vertex. For vertices $u, v$ of $G$, let $W(u, v)$ denote the number of walks of length 2 from $u$ to $v$ in $G$.

*Algorithm 4.* Input: connected graph $G = (V, E)$ with $|G| \geq k$.
Output: a connected $k$-subgraph of $G$, denoted as $\textsc{Alg4}(G)$.

---

1. $G_\ell \leftarrow G[V \setminus V_h]$.
2. Compute $W(u, v)$ for all pairs of vertices $u, v$ in $G_\ell$.
3. For every $v \in V \setminus V_h$, construct a connected $k$-subgraph $C^v$ of $G$ as follows:

– Sort the vertices $u \in V \setminus V_h \setminus \{v\}$ with positive $W(v,u)$ as $v_1, v_2, \ldots, v_t$ such that $W(v,v_1) \geq W(v,v_2) \geq \cdots \geq W(v,v_t) > 0$.
– $P^v \leftarrow \{v_1, \ldots, v_{\min\{t,k/2-1\}}\}$
– $B^v \leftarrow$ a set of $\min\{d_{G_\ell}(v), k/2\}$ neighbors of $v$ in $G_\ell$ such that the number of edges between $B^v$ and $P^v$ is maximized.
– $C^v \leftarrow$ the component of $G_\ell[\{v\} \cup B^v \cup P^v]$ that contains $v$
– Expand $C^v$ to be a connected $k$-subgraph of $G$

4. Output ALG4($G$) $\leftarrow$ the densest $C^v$ for $v \in V \setminus V_h$

In the above algorithm, $B^v$ can be found in $O(m + n \log n)$ time, and $v$ is the "hub" vertex ensuring that $C^v$ is connected. Hence the algorithm is correct, and runs in $O(mn + n^2 \log n)$ time, where Step 2 finishes in $O(n^2 \log n)$ time. The key point here is that $C^v$ contains all edges between $B^v$ and $P^v$, where $B^v$ and $P^v$ are not necessarily disjoint. Using a similar analysis to that in [11] (see [6]), we obtain the following.

**Lemma 7.** *If* $k \leq \frac{2}{3}n$, *then* $\sigma(\text{ALG4}(G)) \geq \frac{(\sigma_k^*(G)-2\bar{\sigma})^2}{2\max\{k,2d_h\}} \cdot \frac{k-2}{k} \geq \frac{(\sigma_k^*(G)-2\bar{\sigma})^2}{6\max\{k,2d_h\}}$. □

We are now ready to prove that the four algorithms given above jointly guarantee an $O(n^{2/5})$-approximation.

**Theorem 2.** *A connected $k$-subgraph $C$ of $G$ can be found in $O(mn \log n)$ time such that $\sigma_k^*(G)/\sigma(C) \leq O(n^{2/5})$.*

*Proof.* Let $C$ be the densest connected $k$-subgraph of $G$ among the outputs of Algorithms 1 – 4. As mentioned at the beginning of Sect. 3.2, it suffices to consider the case of $k < n^{4/5}$. The connectivity of $C$ gives $\sigma(C) \geq 1$. Clearly, we may assume $n \geq 8$, which along with $k < n^{4/5}$ implies $k \leq 2n/3$. By Lemmas 5–7, we may assume that

$$\sigma(C) \geq \max\left\{1, \frac{k\sigma^*(G)}{4n}, \frac{\bar{\sigma}}{\sqrt{k}}, \frac{\sqrt{k}d_h}{2n}, \frac{(\sigma_k(G)-2\bar{\sigma})^2}{6\max\{k,2d_h\}}\right\}.$$

If $k \geq n^{3/5}$, then $\sigma(C) \geq k \cdot \sigma^*(G)/(4n) \geq \sigma^*(G)/(4n^{2/5}) \geq \sigma_k^*(G)/(4n^{2/5})$. If $k \leq n^{2/5}$, then $\sigma(C) \geq 1 \geq \sigma_k^*(G)/k \geq \sigma_k^*(G)/n^{2/5}$. So we are only left with the case of $n^{2/5} \leq k \leq n^{3/5}$.

Since $\sigma(C) \geq \bar{\sigma}/\sqrt{k} \geq \bar{\sigma}/n^{3/10} \geq \bar{\sigma}/n^{2/5}$, we may assume $\bar{\sigma} < \sigma_k^*(G)/4$, and hence $\sigma_k^*(G) - 2\bar{\sigma} \geq \sigma_k^*(G)/2$. Next we use the geometric mean to prove the performance guarantee as claimed.

In case of $k \geq 2d_h$, since $\sigma^*(G) \geq \sigma_k^*(G)$, we have

$$\sigma(C) \geq \left(1 \cdot \frac{k\sigma^*(G)}{4n} \cdot \frac{(\sigma_k^*(G)/2)^2}{6k}\right)^{1/3} \geq \frac{\sigma_k^*(G)}{5n^{2/5}},$$

In case of $k < 2d_h$, we have

$$\sigma(C) \geq \left(1 \cdot \frac{\sqrt{k}d_h}{2n} \cdot \frac{(\sigma_k^*(G)/2)^2}{12d_h} \cdot \frac{\sqrt{k}d_h}{2n} \cdot \frac{(\sigma_k^*(G)/2)^2}{12d_h}\right)^{1/5} \geq \frac{\sigma_k^*(G)}{7n^{2/5}},$$

where the last inequality follows from the fact that $k \geq \sigma_k^*(G)$. □

## 4 Conclusion

In Sect. 3, we have given four strongly polynomial time algorithms that jointly guarantee an $O(\min\{n^{2/5}, n^2/k^2\})$-approximation for the unweighted problem – DC$k$SP. The approximation ratio is compared with the maximum density of *all* $k$-subgraphs, and in this case no $O(n^{1/3-\varepsilon})$-approximation for any $\varepsilon > 0$ can be expected (recall $\Lambda \geq \frac{1}{3}n^{1/3}$ in Example 1(a)). When studying the weighted generalization – HC$k$SP, we can extend the techniques developed in Sect. 3.1, and obtain an $O(n^2/k^2)$-approximation for the weighted case. Besides, a simple greedy approach can achieve a $(k/2)$-approximation [6]. As $\min\{n^2/k^2, k\} \leq n^{2/3}$, the following result implies an $O(n^{2/3})$-approximation for HC$k$SP.

**Theorem 3.** *For any connected graph $G = (V, E)$ with weight $w \in \mathbb{Z}_+^E$, a connected $k$-subgraph $H$ of $G$ can be found in $O(nm)$ time such that $\sigma_k^*(G, w)/\sigma(H, w) \leq O(\min\{n^2/k^2, k\})$, where $\sigma(H, w)$ is the weighted density of $H$, and $\sigma_k^*(G, w)$ is the weighted density of a heaviest $k$-subgraph of $G$ (which is not necessarily connected).* □

Since the weighted density of a graph is not necessarily related to its number of edges or vertices, a couple of the results in the previous sections (such as Lemmas 2, 6 and 7) do not hold for the general weighted case. Neither the techniques of extending unweighted case approximations to weighted cases in [11,17] apply to our setting due to the connectivity constraint. An immediate question is whether an $O(n^{2/5})$-approximation algorithm exists for HC$k$SP. Note from $\Lambda_w \geq \frac{1}{2}n^{1/2}$ in Example 1(b) that no one can achieve an $O(n^{1/2-\varepsilon})$-approximation for any $\varepsilon > 0$ if the solution value is compared with the maximum weighted density of *all* $k$-subgraphs. Among other algorithmic approaches, analyzing the properties of densest/heaviest *connected* $k$-subgraphs is an important and challenging task in obtaining improved approximation ratios for DC$k$SP and HC$k$SP.

## References

1. Andersen, R., Chellapilla, K.: Finding dense subgraphs with size bounds. In: Avrachenkov, K., Donato, D., Litvak, N. (eds.) WAW 2009. LNCS, vol. 5427, pp. 25–37. Springer, Heidelberg (2009)
2. Arora, S., Karger, D., Karpinski, M.: Polynomial time approximation schemes for dense instances of NP-hard problems. In: Proceedings of the 27th Annual ACM Symposium on Theory of Computing, pp. 284–293 (1995)
3. Asahiro, Y., Iwama, K., Tamaki, H., Tokuyama, T.: Greedily finding a dense subgraph. J. Algorithms **34**(2), 203–221 (2000)
4. Bhaskara, A., Charikar, M., Chlamtac, E., Feige, U., Vijayaraghavan, A.: Detecting high log-densities: an $O(n^{1/4})$ approximation for densest $k$-subgraph. In: Proceedings of the 42nd Annual ACM Symposium on Theory of Computing, pp. 201–210 (2010)
5. Chen, Danny Z., Fleischer, Rudolf, Li, Jian: Densest $k$-subgraph approximation on intersection graphs. In: Jansen, Klaus, Solis-Oba, Roberto (eds.) WAOA 2010. LNCS, vol. 6534, pp. 83–93. Springer, Heidelberg (2011)

6. Chen, X., Hu, X., Wang, C.: Finding connected dense $k$-subgraphs. CoRR abs/ 1501.07348 (2015)
7. Corneil, D.G., Perl, Y.: Clustering and domination in perfect graphs. Discrete Appl. Math. **9**(1), 27–39 (1984)
8. Demaine, E.D., Hajiaghayi, M., Kawarabayashi, K.i.: Algorithmic graph minor theory: decomposition, approximation, and coloring. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, pp. 637–646 (2005)
9. Feige, U.: Relations between average case complexity and approximation complexity. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing, pp. 534–543 (2002)
10. Feige, U., Langberg, M.: Approximation algorithms for maximization problems arising in graph partitioning. J. Algorithms **41**(2), 174–211 (2001)
11. Feige, U., Peleg, D., Kortsarz, G.: The dense $k$-subgraph problem. Algorithmica **29**(3), 410–421 (2001)
12. Goldberg, A.V.: Finding a Maximum Density Subgraph. University of California Berkeley, CA (1984)
13. Han, Q., Ye, Y., Zhang, J.: An improved rounding method and semidefinite programming relaxation for graph partition. Math. Program. **92**(3), 509–535 (2002)
14. Hassin, R., Rubinstein, S., Tamir, A.: Approximation algorithms for maximum dispersion. Oper. Res. Lett. **21**(3), 133–137 (1997)
15. Khot, S.: Ruling out ptas for graph min-bisection, dense $k$-subgraph, and bipartite clique. SIAM J. Comput. **36**(4), 1025–1071 (2006)
16. Khuller, S., Saha, B.: On finding dense subgraphs. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009, Part I. LNCS, vol. 5555, pp. 597–608. Springer, Heidelberg (2009)
17. Kortsarz, G., Peleg, D.: On choosing a dense subgraph. In: Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, pp. 692–701 (1993)
18. Lawler, E.L.: Combinatorial Optimization: Networks and Matroids. Courier Dover Publications, New York (1976)
19. Liazi, M., Milis, I., Zissimopoulos, V.: Polynomial variants of the densest/heaviest $k$-subgraph problem. In: Proceedings of the 20th British Combinatorial Conference, Durham (2005)
20. Raghavendra, P., Steurer, D.: Graph expansion and the unique games conjecture. In: Proceedings of the 42nd Annual ACM Symposium on Theory of Computing, pp. 755–764 (2010)
21. Srivastav, A., Wolf, K.: Finding dense subgraphs with semidefinite programming. In: Jansen, K., Rolim, J.D.P. (eds.) APPROX 1998. LNCS, vol. 1444, pp. 181–191. Springer, Heidelberg (1998)