Densest k-Subgraph Approximation on Intersection Graphs

Danny Z. Chen[1]⋆, Rudolf Fleischer[2]⋆⋆, and Jian Li[3]

[1] University of Notre Dame, Indiana, USA. E-mail: dchen@cse.nd.edu
[2] Fudan University, SCS and IIPL, Shanghai, China. E-mail: rudolf@fudan.edu.cn
[3] University of Maryland, College Park, MD, USA. E-mail: lijian@cs.umd.edu

Abstract. We study approximation solutions for the densest k-subgraph problem (DS-k) on several classes of intersection graphs. We adopt the concept of σ-quasi elimination orders, introduced by Akcoglu et al. [1], generalizing the perfect elimination orders for chordal graphs, and develop a simple $O(\sigma)$-approximation technique for graphs admitting such a vertex order. This concept allows us to derive constant factor approximation algorithms for DS-k on many intersection graph classes, such as chordal graphs, circular-arc graphs, claw-free graphs, line graphs of ℓ-hypergraphs, disk graphs, and the intersection graphs of fat geometric objects. We also present a PTAS for DS-k on unit disk graphs using the shifting technique.

1 Introduction

The *(connected) densest k-subgraph problem* (DS-k) is defined as follows: Given an undirected graph $G = (V, E)$ with n nodes and m edges and a positive integer k, find an induced (connected) subgraph with k vertices in G maximizing the number of edges. Reduction from the maximum clique problem shows that this problem is NP-hard. The weighted version of DS-k in which each edge has a positive weight and the goal is to maximize the sum of edge weights in the induced subgraph is called the *heaviest k-subgraph problem* (HS-k).

Considerable work has been done on finding good quality approximation algorithms for DS-k. The first non-trivial approximation algorithm by Kortsarz and Peleg achieved an approximation ratio of $O(n^{0.3885})$ [23].

⋆ The research of this author was supported in part by the US National Science Foundation under Grants CCF-0515203 and CCF-0916606. This work was partially done while the author was visiting the Shanghai Key Laboratory of Intelligent Information Processing at Fudan University, China.

⋆⋆ The research of this author was supported by the NSFC (no. 60973026), the Shanghai Leading Academic Discipline Project (no. B114), and the Shanghai Committee of Science and Technology (no. 09DZ2272800).

Arora et al. [3] used random sampling techniques to obtain a polynomial time approximation scheme (PTAS) for dense graphs with $k = \Omega(|V|)$ and $|E| = \Omega(|V|^2)$. Asahiro et al. [4] showed that the greedy method achieves an approximation ratio of $O(\frac{n}{k})$.

Feige et al. proposed an $\frac{n}{k}$-approximation algorithm based on semidefinite programming [15] and an n^δ-approximation algorithm for some $\delta < \frac{1}{3}$ [14]. Recently, Bhaskara et al. [7] proposed an $O(n^{1/4+\epsilon})$ approximation algorithm that runs in time $n^{O(1/\epsilon)}$. In [11], Demaine et al. gave a 2-approximation algorithm for H-minor-free graphs, for any fixed H. It is unlikely that there exists a PTAS for general graphs [21].

For some special graph classes and special values of k, better algorithms are known [19, 32, 34]. Maffioli proposed an $O(nk^2)$ time algorithm for connected HS-k on trees [28]. This algorithm can easily be generalized to solve the unconnected case. Corneil and Pearl gave a polynomial time algorithm for DS-k on co-graphs, a subclass of perfect graphs [10].

Keil and Brecht developed polynomial time algorithms for HS-k on graphs with bounded treewidth based on dynamic programming [20]. Liazi et al. [25] presented a polynomial time algorithm for DS-k/HS-k on chains (i.e., graphs with maximum degree 2), and a subclass of proper interval graphs. They also obtained a PTAS for chordal graphs if the maximal clique intersection graph is a star, and polynomial time algorithms if the maximal clique intersection graph is a tree of bounded degree [26]. Recently, they showed that a simple greedy algorithm achieves an approximation factor of 3 for DS-k on chordal graphs [27].

Finding dense subgraphs with upper or lower bounds on their sizes has also been studied by several researchers [2, 22].

Our Results. In this paper, we focus on DS-k on several intersection graph classes: chordal graphs, circular-arc graphs, line graphs, disk graphs, and unit disk graphs. The closely related maximum clique problem is polynomial time solvable on these graph classes, except on disk graphs. Note that interval graphs are chordal graphs, and chordal graphs are perfect graphs. Although the maximum clique problem is polynomial time solvable on perfect graphs [18], DS-k is NP-hard on perfect graphs, since it is NP-hard on bipartite graphs [30] and chordal graphs [10]. Connected DS-k is NP-hard on planar graphs [20]. The complexity status of unconnected DS-k on planar graphs, interval graphs, and proper interval graphs has been a long-standing open problem [10].

Since the complexity status of these problems is unknown, it is worthwhile to consider efficient approximation algorithms for them. We adopt the notion of σ-quasi elimination orders, for $\sigma \geq 1$, proposed by Akcoglu et al. [1], generalizing the perfect elimination orders for chordal graphs. It turns out that many intersection graph classes mentioned above have $O(1)$-quasi elimination orders [35]. This type of vertex order allows us to derive new approximation algorithms for DS-k. Our main result is an $O(\sigma)$-approximation algorithm for DS-k if the graph has a polynomial time computable σ-quasi elimination order. This immediately implies constant factor approximation ratios for many intersection graph classes. These classes include chordal graphs (with $\sigma = 1$), circular-arc graphs (with $\sigma = 2$), claw-free graphs (with $\sigma = 2$), line graphs of ℓ-hypergraphs (with $\sigma = \ell$), disk graphs (with $\sigma = 5$), unit disk graphs (with $\sigma = 3$), and the intersection graphs of λ-fat objects in d-dimensional space (with $\sigma = (3\lambda)^d$). We also propose a PTAS for DS-k on unit disk graphs based on the shifting technique [6] combined with a result by Arora et al. [3], if a disk representation is given. This improves the recent 1.5-approximation for DS-k on proper interval graphs [5]. Note that the class of proper interval graphs is equivalent to the class of unit interval graphs [31] which is a subset of the class of unit disk graphs.

2 Preliminaries

For a graph $G = (V, E)$, we denote its vertex set by $V(G) = V$ and its edge set by $E(G) = E$. Let $n = |V|$ and $m = |E|$. We denote the degree of a vertex v in G by $deg_G(v)$. For any $v \in V$ and subsets $S, W \subseteq V$, let $d(v, W)$ be the number of edges (v, w) with $w \in W$, and $d(S, W) = \sum_{u \in S} d(u, W)$. Let $G[S]$ denote the subgraph of G induced by $S \subset V$. Let $\alpha(G)$ be the *independence number* of G, i.e., the size of a maximum independent set in G. The classic Turán bound states that

$$\alpha(G) \geq \sum_{v \in V} \frac{1}{deg(v) + 1} \geq \frac{n}{\bar{d} + 1}$$

where $\bar{d}$ is the average degree of the nodes in the graph [33]. For convenience, we rephrase the bound in the following lemma.

Lemma 1. *For any graph G, $m \geq \dfrac{n^2 - n\alpha(G)}{2\alpha(G)}$.* □

3 Elimination Orders and Intersection Graphs

If $\mathcal{L} = (v_1, v_2, \ldots, v_n)$ is an ordering of the vertices in V, we define $Pred_{\mathcal{L}}(v_i) = \{v_i\} \cup \{v_j \mid j < i \text{ and } (v_j, v_i) \in E\}$, the *predecessors* of

v_i, and $Succ_{\mathcal{L}}(v_i) = \{v_j \mid j > i \text{ and } (v_i, v_j) \in E\}$, the *successors* of v_i. In a *perfect elimination order*, every set $Pred_{\mathcal{L}}(v_i)$ forms a clique (note that sometimes in the literature it is required that every set $Succ_{\mathcal{L}}(v_i)$ forms a clique, instead, which just reverses the order). We can generalize this definition by allowing some slack. Let σ be a positive integer.

Definition 2. *A* σ-quasi elimination order (σ-QEO) *of G is an ordering $\mathcal{L} = (v_1, v_2, \ldots, v_n)$ of the vertices in V such that $\alpha(G[Pred_{\mathcal{L}}(v_i)]) \leq \sigma$ for $i = 2, \ldots, n$.*

A perfect elimination order is just a 1-QEO. This notion was introduced by Akcoglu et al. [1] who proposed a σ-approximation for the weighted maximum independent set problem. Recently, Ye and Borodin explored many properties of QEOs and initiated a more comprehensive study on their algorithmic aspects [35]. In particular, they considered the maximum σ-colorable subgraphs problem, the minimum vertex covering problem and the minimum vertex coloring problem and obtained improved approximation algorithms on graphs with $O(1)$-QEO. Lemma 1 implies that $G[Pred_{\mathcal{L}}(v_i)]$ has at least $\frac{1}{2\sigma} \cdot \binom{|Pred_{\mathcal{L}}(v_i)|}{2}$ edges, for every v_i in $\mathcal{L}$, if $|Pred_{\mathcal{L}}(v_i)| \geq 2\sigma - 1$. Note that any induced subgraph of G has a σ-QEO if G has one. In this paper, we study the following graph classes.

Chordal graphs. G is a *chordal graph* if it does not contain an induced cycle of length k, for any $k \geq 4$. Chordal graphs are exactly the intersection graphs of subtrees in a tree. A graph is chordal if and only if it has a perfect elimination order [17].

Circular-arc graphs. A *circular-arc graph* is the intersection graph of arcs of a circle. Circular-arc graphs are not always chordal, for example any chordless cycle of length greater than four is a circular-arc graph. It is easy to see that any circular-arc graph has a 2-QEO.

Line graphs. A graph L is the *line graph* of the (hyper-)graph G if L is the intersection graph of the (hyper-)edges of G.

Claw-free graphs. A graph G is *claw-free* if it excludes $K_{1,3}$ as an induced subgraph. Claw-free graphs generalize line graphs, which initially motivated the study of claw-free graphs. They have many nice properties, for example, claw-free graphs always have a perfect matching and we can find a maximum independent set in polynomial time. However, it is NP-hard to compute a largest clique in a claw-free graph. For a survey on more results on claw-free graph, see [24], for example. Conveniently, any ordering of the vertices of a claw-free graph is a 2-QEO.

(Unit) Disk graphs. G is a (unit) disk graph if it is the intersection graph of a set of closed (unit) disks in the plane. The *disk representation* specifies the centers and radii of the disks. If the disks are not given, the recognition problem of (unit) disk graphs is NP-hard [9]. Disk graphs are a two-dimensional generalization of interval graphs. However, in general, they are neither planar nor perfect. Some NP-hard problems become tractable on unit disk graphs (e.g., the maximum clique problem [8]), and some problems admit significantly better approximation algorithms (e.g., there is a PTAS for the maximum independent set problem on unit disk graphs [29] and on arbitrary disk graphs [13]). Ye and Borodin showed that any (unit) disk graph has a (3-QEO) 5-QEO [35].

Fat intersection graphs. Practical instances of geometric problems often deal with objects of "reasonable" shape. One way to formalize this is the notion of fat objects. There are several different definitions of fat objects in computational geometry literature (e.g., see [12]). In this paper, we say a d-dimensional convex object K is λ-*fat*, for some parameter $\lambda \geq 1$ (the *fatness*), if the ratio between the radii of B_K^+ and B_K^- is at most λ, where B_K^+ is a smallest sphere containing K and B_K^- is a largest sphere contained in K. Examples of objects of bounded fatness are spheres (fatness 1), cubes (fatness $\sqrt{d}$), and ellipsoids with bounded aspect ratio.

A *fat intersection graph* is the intersection graph of a set of fat objects. For example, disk graphs are fat intersection graphs.

Lemma 3. *Every fat intersection graph of λ-fat convex objects in d-dimensional space has an $O((3\lambda)^d)$-QEO.*

Proof. We sort the vertices of the graph in non-increasing order of the largest disk contained in each corresponding fat object. Then, for each vertex v_i, $\alpha(G[Pred_{\mathcal{L}}(v_i)]) = O((3\lambda)^d)$. Since similar ideas have been used before in the literature on algorithms for fat objects (e.g., see [12]), we omit the details of the proof. □

4 Approximating DS-k on Graphs with σ-QEO

In this section, we present a constant factor approximation technique for DS-k on chordal graphs and fat intersection graphs. We focus on presenting the general framework and do not emphasize on fine-tuning the parameters for the smallest possible approximation factor. We use the maximum density subgraph problem, which is polynomial time solvable, as a key subroutine.

4.1 The Maximum Density Subgraph Problem (MDSP)

The *maximum density subgraph problem (MDSP)* is defined as follows: Given a graph $G = (V, E, w)$ with non-negative vertex weights $w : V \to \mathbb{R}^+ \cup \{0\}$, find an induced subgraph $H = (W, F)$ maximizing the density

$$\rho(H) = \frac{\sum_{v \in W} w(v) + |F|}{|W|}.$$

This problem can be solved optimally in $O(nm \log(\frac{n^2}{m}))$ time by a reduction to the parametric maximum flow algorithm [16] which produces an induced subgraph $H = (W, F)$ maximizing $\frac{\sum_{e \in F} w(e)}{\sum_{v \in W} w(v)}$, where w is a weight function on the vertices and edges (we set the weights of all original vertices and edges to 1; then we create a sibling with weight zero for each vertex in V, connected to the original vertex by an edge of weight $w(v)$).

Note that $w(v) + deg_H(v) \geq \rho$, for each vertex $v \in W$, for any optimal MDSP solution $H = (W, F)$ with maximum density ρ. This is because we could delete from H all vertices violating this inequality to obtain an induced subgraph of higher density.

4.2 A Constant Factor Approximation Framework

In this subsection, we show how to compute an $O(\sigma)$-approximation for DS-k on any graph $G = (V, E)$ for which we can efficiently compute a σ-quasi elimination order.

At a high level, our framework works as follows. If we solve MDSP on G with $w(v) = 0$ for all $v \in V$ and obtain a subgraph H of k' vertices, then H is also an optimal DS-k' solution. If H is smaller than the sought DS-k solution (i.e., $k' < k$), then we repeat the MDSP algorithm on the remaining vertices of G and combine the solution with H (Phase 1). If H is larger (i.e., $k' > k$), then we select some vertices in H to satisfy the cardinality constraint without losing too much density (Phase 2).

Let $G^\star = (V^\star, E^\star)$ be an optimal DS-k solution on $G = (V, E)$ with density $\rho^\star = \frac{|E^\star|}{|V^\star|}$. Without loss of generality assume $\rho^\star \geq 8\sigma$; otherwise, we can trivially get an $O(\sigma)$-approximation.

Phase 1: Growing U_t. Let $V_0 = V$, $E_0 = E$, and $w_0(v) = 0$ for all $v \in V_0$. Starting with $i = 0$, let G_{i+1} be obtained from G_i by removing the vertices and adjacent edges of an optimal MDSP solution $H_i = (W_i, F_i)$

of $G_i = (V_i, E_i, w_i)$ of density ρ_i, where $w_i(v) = d(v, U_{i-1})$ for $v \in V_i$. Let $U_i = \cup_{j=0}^{i} W_j$ be the set of all removed nodes, and $n_i = |U_i|$. We stop at the first time t such that $n_t \geq \frac{k}{2}$. If $n_t \leq k$, then we return U_t plus some arbitrary $k - n_t$ nodes from V_{t+1} as our DS-k approximation.

Lemma 4. *If $n_t \leq k$, then U_t is a 4-approximation for DS-k on G.*

Proof. If $G[U_t \cap V^\star]$ has at least $\frac{|E^\star|}{2}$ edges, then U_t is a 2-approximation for DS-k on G. If not, then let $I_i = U_i \cap V^\star$ and $R_i = V^\star \setminus I_i$, for all i. Since $|E(I_t)| = |E(U_t \cap V^\star)| < \frac{|E^\star|}{2}$, we have for $i \leq t$

$$\begin{aligned}\rho_i &= \frac{|F_i| + d(U_{i-1}, W_i)}{|W_i|} \geq \frac{|E(R_{i-1})| + d(U_{i-1}, R_{i-1})}{|R_{i-1}|} \\ &\geq \frac{|E(R_{i-1})| + d(I_{i-1}, R_{i-1})}{|R_{i-1}|} \geq \frac{|E(R_{i-1})| + d(I_{i-1}, R_{i-1})}{k} \\ &= \frac{|E^\star| - |E(I_{i-1})|}{k} \geq \frac{|E^\star| - |E(I_t)|}{k} \geq \frac{|E^\star|}{2k} = \frac{\rho^\star}{2} .\end{aligned}$$

Hence,

$$|E(U_t)| \geq \sum_{i \leq t} (\rho_i \cdot |U_i|) \geq \min_{i \leq t}\{\rho_i\} \cdot \sum_{i \leq t} |U_i| \geq \min_{i \leq t}\{\rho_i\} \cdot k/2 \geq \frac{|E^\star|}{4} . \quad \square$$

Phase 2: Shrinking U_t. If $n_t > k$, then we must delete some vertices from U_t without decreasing the density too much. We first compute a σ-quasi elimination order $\mathcal{L} = \{v_1, \ldots, v_{n_t}\}$ for U_t. If some vertex in $\mathcal{L}$ has a large predecessor set in this order, then we are done, as shown by Lemma 5.

Lemma 5. *If there is a vertex $v \in U_t$ with $|Pred_{\mathcal{L}}(v)| \geq \frac{k}{2}$, then we can efficiently find a subgraph of $\frac{k}{2}$ vertices in $Pred_{\mathcal{L}}(v)$ that is an $O(\sigma)$-approximation for DS-k on G.*

Proof. Let $\mathcal{A} = Pred_{\mathcal{L}}(v)$. Since $|\mathcal{A}| \geq \frac{k}{2} \geq \rho^\star \geq 2\sigma - 1$, the σ-quasi elimination order property implies that $G[\mathcal{A}]$ has at least $\frac{1}{2\sigma} \cdot \binom{|\mathcal{A}|}{2}$ edges by Lemma 1. We randomly and uniformly choose a subset $\mathcal{B}$ of the $\frac{k}{2}$ vertices in $\mathcal{A}$. Then, $G[\mathcal{B}]$ has an expected number of $\Theta(\frac{1}{\sigma}) \cdot \Theta(k^2)$ edges:

$$\sum_{e \in E(\mathcal{A})} \frac{k}{2|\mathcal{A}|} \cdot \frac{k}{2|\mathcal{A}|} \geq \frac{k^2}{16\sigma} \cdot (1 - \frac{1}{|\mathcal{A}|}) \geq \frac{1}{\sigma} \cdot (\frac{k^2}{16} - \frac{k}{8}) .$$

It is straightforward to derandomize this algorithm using the conditional probability technique. We omit the details. $\square$

If no vertex v in $\mathcal{L}$ has a predecessor set of size at least $\frac{k}{2}$, then we must work a bit harder to find a dense subgraph.

Lemma 6. *If there is no vertex $v \in U_t$ with $|Pred_{\mathcal{L}}(v)| \geq \frac{k}{2}$, then we can efficiently find a subset of U_t of size at most k that is an $O(\sigma)$-approximation for DS-k on G.*

Proof. From the remark at the end of Subsection 4.1, we see that for any vertex $v \in U_t$, either (1) $|Succ_{\mathcal{L}}(v)| > \frac{\rho_t}{2}$, or (2) $|Pred_{\mathcal{L}}(v)| \geq \frac{\rho_t}{2}$. We now process the vertices of U_t in the reverse order of $\mathcal{L}$, i.e., beginning at v_{n_t}. If a vertex satisfies condition (1) above, then we take it. If it satisfies condition (2), then we take it together with a certain subgraph of high-degree vertices of its predecessor set (see Lemma 7 below).

We stop if we have collected at least $\frac{k}{2}$ vertices. In every step, we either add a single vertex v or a subset of its predecessors to the solution. Since no vertex has a predecessor set of size at least $\frac{k}{2}$, we select at most k vertices in total, i.e., we obtain a feasible solution SOL for DS-k.

In $G[SOL]$, each vertex v has a degree either at least $\frac{\rho_t}{2}$ if it was selected by condition (1), or $\frac{|Pred_{\mathcal{L}}(v)|-1}{4\sigma} \geq \frac{\rho_t-2}{8\sigma}$ if it was selected by condition (2). Thus,

$$|SOL| = \frac{1}{2} \sum_{v \in SOL} deg(v) \geq \frac{\rho_t - 2}{8\sigma} \cdot k \geq \frac{\rho^\star - 4}{8\sigma} \cdot k = O(\frac{1}{\sigma}) \cdot |E^\star| \, .$$

□

Lemma 7. *If $|Pred_{\mathcal{L}}(v)| \geq \frac{\rho_t}{2}$ for some vertex v, then we can efficiently identify a non-empty subset $\mathcal{H}$ of $Pred_{\mathcal{L}}(v)$ such that every vertex in $G[\mathcal{H}]$ has a degree at least $\frac{|Pred_{\mathcal{L}}(v)|-1}{4\sigma}$.*

Proof. We repeatedly delete a vertex of degree less than $\frac{|Pred_{\mathcal{L}}(v)|-1}{4\sigma}$. Since $|Pred_{\mathcal{L}}(v)| \geq \frac{\rho_t}{2} \geq \frac{\rho^\star}{4} \geq 2\sigma$, $G[Pred_{\mathcal{L}}(v)]$ contains at least $\frac{1}{2\sigma} \cdot \binom{|Pred_{\mathcal{L}}(v)|}{2}$ edges, and thus we cannot delete all vertices (and their edges) of $Pred_{\mathcal{L}}(v)$. □

Theorem 8. *If G has a polynomial time computable σ-QEO, then we can efficiently compute an $O(\sigma)$-approximation for DS-k on G.* □

It is known that a σ-QEO can be constructed in $O(\sigma^2 n^{\sigma+2})$ time if there is one [35]. In particular, we can find an $O(1)$-QEO in polynomial time. Combined with Theorem 8, we obtain the claimed results on intersection graphs.

Corollary 9. *There is an $O(1)$-approximation algorithm for* DS-k *on the following intersection graph classes, even if the intersection models are not given as input: chordal graphs, circular-arc graphs, claw-free graphs, line graphs of ℓ-hypergraphs (with $\ell = O(1)$), disk graphs (with $\sigma = 5$), unit disk graphs (with $\sigma = 3$), and the intersection graphs of λ-fat objects in d-dimensional space (with $\lambda = O(1)$ and $d = O(1)$).* □

5 A PTAS for DS-k on Unit Disk Graphs

A PTAS for DS-k on unit disk graphs can be obtained by a standard shifting technique [6], combined with a result by Arora et al. [3]. This technique was also used to develop a PTAS for the maximum independent set problem on unit disk graphs [29]. We give a brief sketch of our algorithm. The following lemma indicates how to combine the optimal solutions for HS on independent subgraphs into a global optimal solution.

Lemma 10. *Let G be a graph with connected components $G_1, \ldots, G_p$. If we can efficiently solve* HS-ℓ *on all G_i, for any ℓ, then we can efficiently solve* HS-ℓ *on G, for any ℓ.*

Proof. Let $OPT(G, \ell)$ denote an optimal solution of HS-ℓ on G. Then, $OPT(\cup_{i=1}^{j} G_i, \ell)$ can be computed by the following dynamic program, for any j and ℓ:

$$OPT(\cup_{i=1}^{j} G_i, \ell) = \max_{x} \{OPT(\cup_{i=1}^{j-1} G_i, x) + OPT(G_j, \ell - x)\} \ .$$

□

We may assume that the given disks have diameter one and the disk centers do not have integral coordinates. Let h be a constant to be fixed later. For all $0 \leq i, j \leq k-1$, we define $\mathcal{D}_{i,j}$ to be the set of disks obtained by removing all disks intersecting a vertical line $x = i + ha$ for some integer a or a horizontal line $y = j + hb$ for some integer b.

Let $OPT(G, k)$ be an optimal DS-k solution for G. We can show that $\sum_{i=0}^{h-1} \sum_{j=0}^{h-1} |OPT(G, k) \cap \mathcal{D}_{i,j}| \geq (h-2)^2 \cdot |OPT(G, k)|$. Therefore, there exist i, j such that $|OPT(\mathcal{D}_{i,j}, k)| \geq |OPT(G, k) \cap \mathcal{D}_{i,j}| \geq (1 - \frac{2}{h})^2 \cdot |OPT(G, k)|$. By choosing $h = 2/\epsilon$, we see that $\max_{i,j} |OPT(\mathcal{D}_{i,j}, k)|$ is a $(1-\epsilon)$-approximation. Now, we have reduced DS-k on G to computing $OPT(\mathcal{D}_{i,j}, k)$. In the following, we will give a PTAS for computing $OPT(\mathcal{D}_{i,j}, k)$. This gives us a PTAS for DS-k on G.

$\mathcal{D}_{i,j}$ may consist of several connected components, each of which is contained in an $h \times h$ square. Let C be one of the components with n_c vertices. An $h \times h$ square can be covered by $(h+1)^2+h^2 = 2h^2+2h+1$ unit disks. Thus, C can be covered by no more than $2h^2+2h+1$ disjoint cliques, since a set of disks whose centers lie in a common unit circle induces a clique. Therefore, one of these cliques contains no less than $\frac{n_c}{2h^2+2h+1}$ vertices. If $k \le \frac{n_c}{2h^2+2h+1}$, then $OPT(C, k)$ is a clique. If $k > \frac{n_c}{2h^2+2h+1}$, the size of a maximum independent set in C is no more than $2h^2+2h+1$. By Lemma 1, C contains $\Theta(\frac{n_c^2}{h^2})$ edges. Since h is a constant, we can use the algorithm in [3] to obtain a PTAS for problem instances with $\Theta(n_c^2)$ edges and satisfying $k = \Theta(n_c)$. Now, by Lemma 10, we have a PTAS for computing $OPT(\mathcal{D}_{i,j}, k)$.

We note that similar ideas can be used to obtain a PTAS for unit square intersection graphs. Erlebach et al. [13] used a new subdivision of the plane and the shifting strategy to obtain a PTAS for the maximum independent set problem and the vertex cover problem for disk graphs. However, it is not clear whether their methods can be applied to obtaining a PTAS for DS-k on disk graphs.

6 Conclusions

In this paper, we studied approximation algorithms for the densest k-subgraph (DS-k) problem on several classes of intersection graphs. One of our main contributions is a simple $O(\sigma)$-approximation framework for graphs admitting σ-QEOs, which leads to improved approximation DS-k algorithms for these graph classes.

One future research direction is to find more algorithmic applications for σ-QEO. It is worthwhile noting that after the MDSP preprocessing phase, our algorithm is essentially based on local decisions guided by the vertex ordering. This is similar to the approximation algorithms for various graph problems developed in [1, 35]. Therefore, we conjecture that there might be a deeper reason to explain this, or even a unified characterization of the problem structures that allows us to apply certain local decision-based approximation algorithms on graphs with QEOs.

Note that all graph classes we considered have σ-quasi elimination orders with some constant $\sigma \ge 1$. Thus, another direction of research is to identify other graph classes with a σ-QEO such that σ is $o(n^{1/4})$ (this ensures an approximation better than $O(n^{1/4})$ for DS-k [7]).

References

1. K. Akcoglu, J. Aspnes, B. DasGupata, and M.-Y. Kao. Opportunity cost algorithm for combinatorial auctions. *Applied Optimization 74: Computational Methods in Decision-Making, Economics and Finance*, pp. 455–479, 2002.
2. R. Andersen, K. Chellapilla. Finding dense subgraphs with size bounds. In *Proceedings of the 6th International Workshop on Algorithms and Models for the Web-Graph (WAW'09)*, pp. 25–36, 2009.
3. S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing (STOC'95)*, pp. 284–293, 1995.
4. Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. In *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT'96)*, Springer LNCS 1097, pp. 136–148, 1996.
5. J. Backer and J. M. Keil. Constant factor approximation algorithms for the densest k-subgraph problem on proper interval graphs and bipartite permutation graphs. *Information Processing Letters*, 110(16):635–638, 2010.
6. B. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.
7. A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. 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 (STOC'10)*, pp. 201–210, 2010.
8. H. Breu. Algorithmic aspects of constrained unit disk graphs. PhD thesis, University of British Columbia, 1996.
9. H. Breu and D. G. Kirkpartrick. Unit disc graph recognition is NP-hard. *Computational Geometry: Theory and Applications*, 9:3–24, 1998.
10. D. G. Corneil and Y. Perl. Cluster and domination in perfect graphs. *Discrete Applied Mathematics*, 9:27–39, 1984.
11. E. D. Demaine, M. Hajiaghayi, and K. Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pp. 637–646, 2005.
12. A. Efrat, M. J. Katz, F. Nielsenc, and M. Sharir. Dynamic data structures for fat objects and their applications. *Computational Geometry: Theory and Applications*, 15:215–227, 2000.
13. T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation scheme for geometric graphs. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, pp. 671–679, 2001.
14. U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, 29:410–421, 2001.
15. U. Feige and M. Selter. On the densest k-subgraph problem. Technical report, Department of Applied Mathematics and Computer Science, the Weizmann Institute, Rehovot, September 1997.
16. G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–50, 1989.
17. M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
18. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, Springer Verlag, 1988.

19. Q. Han, Y. Ye, and J. Zhang. Approximation of dense-k-subgraph. Manuscript, 2000.
20. J. M. Keil and T. B. Brecht. The complexity of clustering in planar graphs. *The Journal of Combinatorial Mathematics and Combinatorial Computing*, 9:155–159, 1991.
21. S. Khot. Ruling out PTAS for graph min-bisection, densest subgraph and bipartite clique. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, pp. 136–145, 2004.
22. S. Khuller and B. Saha. On finding dense subgraphs. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09)*, pp. 597–608, 2009.
23. G. Kortsarz and D. Peleg. On choosing a dense subgraph. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS'93)*, pp. 692–701, 1993.
24. R. Faudree, E. Flandrin, and Z. Ryjáek. Claw-free graphs — A survey. *Discrete Mathematics*, 164(1–3):87–147, 1997.
25. M. Liazi, I. Milis, and V. Zissimopoulos. Polynomial variants of the densest/heaviest k-subgraph problem. In *Proceedings of the 20th British Combinatorial Conference*, Durham, 2005.
26. M. Liazi, I. Milis, and V. Zissimopoulos. The densest k-subgraph problem on clique graphs. *Journal of Combinatorial Optimization*, 14(4):465–474, 2007.
27. M. Liazi, I. Milis, and V. Zissimopoulos. A constant approximation algorithm for the densest k-subgraph problem on chordal graphs. *Information Processing Letters*, 108(1), 29–32, 2008.
28. F. Maffioli. Finding a best subtree of a tree. Technical Report 91.041, Politecnico di Milano, Dipartimento di Elettronica, Italy, 1991.
29. T. Matsui. Approximation algorithms for maximum independent set problem and fractional coloring problems on unit disk graphs. In *Proceedings of the 1st Japanese Conference on Discrete and Computational Geometry (JCDCG'98)*, Springer LNCS 1763, pp. 194–200, 2000.
30. R. Peeters. The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131(33):651–654, 2003.
31. F. S. Roberts. Indifference graphs. In *Proof Techniques in Graph Theory,* F. Harary, ed., Academic Press, New York, 1969.
32. A. Srivastav and K. Wolf. Finding dense subgraphs with semidefinite programming. In *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'98)*, Springer LNCS 1444, pp. 181–191, 1998.
33. P. Turán. On the theory of graphs. *Colloquium Mathematicum*, 3:19–30, 1954.
34. Y. Ye and J. Zhang. Approximation of dense $\frac{n}{2}$-subgraph problem and the complement of min-bisection. *Journal of Global Optimization*, 25(1):55–73, 2003.
35. Y. Ye and A. Borodin. Sequential elimination graphs. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09)*, pp. 774–785, 2009.