A THE CDS ALGORITHM FOR ALL k VALUES

Algorithm 4: CCAS-A

input : A graph G, two integers T, and ω.
output: Approximate CDS for all k values.

1 $\mathcal{S}_\omega \leftarrow$ an approximate ω-clique CDS of G;
2 $\gamma \leftarrow \max\{x \mid \binom{x}{\omega} \le \rho_\omega(\mathcal{S}_\omega) \cdot |\mathcal{S}_\omega|\}$;
3 $G_3 \leftarrow G$;
4 **foreach** $k \leftarrow 3, 4, \cdots, \omega - 1$ **do**
5 SCT $\leftarrow$ build_SCT(G_k); // build the SCT for G_k
6 $\mathcal{S}_k(G) \leftarrow$ run lines 3-19 in Algorithm 3 ;
7 $\mu_{k+1} \leftarrow \max\{\mu \mid \binom{\mu}{k} \le \frac{\binom{\gamma}{k+1}}{|\mathcal{S}_\omega|}\}$;
8 $V_i \leftarrow$ the first i-th vertices being removed in the first iteration;
9 $\lambda \leftarrow$ the largest i such that $\max_{v \in V_i} l^1(v) \le \binom{\mu_{k+1}}{k}$;
10 $G_{k+1} \leftarrow$ remove all vertices in V_λ from G_k;
11 **return** $\mathcal{S}_3(G), \mathcal{S}_4(G), \cdots, \mathcal{S}_\omega(G)$

Algorithm 4 illustrates the CCAS-A algorithm for finding CDS over all k values. It first computes an approximate CDS of ω-cliques, denoted as S_ω, where ω represents the maximum clique size (line 1). Next, based on Lemma 6.1, it calculates the γ value, which is used for the subsequent graph reduction process (line 2). CCAS-A process the k value from 3 to ω one by one (lines 4-10). Specifically, for each k, it invokes the CCAS to compute the approximate k-CDS (lines 5-6). Afterwards, it updates the μ value and further reduces the search space based on Theorem 6.2 (lines 7-10). Finally, it returns the approximate CDS for all k values (line 11).

B ADDITIONAL PROOFS OF LEMMAS AND THEOREMS

Lemma 4.2. Given a graph $G = (V, E)$, if $|\Psi_k(G)| \ge \binom{\mu}{k}$ where μ is an arbitrary integer no less than k, then $|\Psi_{k-1}(G)| \ge \binom{\mu}{k-1}$.

Proof. We prove Lemma 4.2 holds by proving its contrapositive, that is, given a graph $G = (V, E)$, if $|\Psi_{k-1}(G)| < \binom{\mu}{k-1}$, then $|\Psi_k(G)| < \binom{\mu}{k}$. Since $|\Psi_{k-1}(G)| < \binom{\mu}{k-1}$, the maximum clique size in G is less than μ. Suppose $\mathcal{R}(G)$ is the set of all maximal cliques in G. For each k-clique in G, it must be included in at least one maximal clique of G, so we have:

$$|\Psi_k(G)| = \left| \bigcup_{R \in \mathcal{R}(G)} \Psi_k(R) \right| \tag{34}$$

That is, $\forall \Phi \subseteq \mathcal{R}(G)$, we denote $g(\Phi) = \left| \bigcap_{R \in \Phi} \Psi_k(R) \right|$, then:

$$g(\Phi) = \left| \bigcap_{R \in \Phi} \Psi_k(R) \right| \tag{35}$$

$$= \left| \Psi_k\left(\bigcap_{R \in \Phi} R \right) \right| \tag{36}$$

$$= \binom{\left| \bigcap_{R \in \Phi} R \right|}{k} \tag{37}$$

This is because a k-clique is contained in all maximal cliques in $\mathcal{S}$, if and only if it is included in the subgraph formed by the intersection of those maximal cliques. Besides, the intersection of some cliques must either be a clique or an empty set, and the number of k-cliques in this subgraph is $\binom{\left| \bigcap_{R \in \mathcal{S}} R \right|}{k}$.

By applying the inclusion-exclusion principle on equation (34),

$$|\Psi_k(G)| = \left| \bigcup_{R \in \mathcal{R}(G)} \Psi_k(R) \right| \tag{38}$$

$$= \sum_{\mathcal{S} \subseteq \mathcal{R}(G)} (-1)^{|\mathcal{S}|} g(\mathcal{S}) \tag{39}$$

$$= \sum_{\mathcal{S} \subseteq \mathcal{R}(G)} (-1)^{|\mathcal{S}|} \binom{\left| \bigcap_{R \in \mathcal{S}} R \right|}{k} \tag{40}$$

Similarly,

$$|\Psi_{k-1}(G)| = \sum_{\mathcal{S} \subseteq \mathcal{R}(G)} (-1)^{|\mathcal{S}|} \binom{\left| \bigcap_{R \in \mathcal{S}} R \right|}{k-1} \tag{41}$$

Therefore, there exist a vector $\beta = \{\beta_{k-1}, \beta_k, \cdots, \beta_{\mu-1}\}$, such that,

$$|\Psi_{k-1}(G)| = \sum_{i=k-1}^{\mu-1} \beta_i \binom{i}{k-1} \tag{42}$$

$$|\Psi_k(G)| = \sum_{i=k-1}^{\mu-1} \beta_i \binom{i}{k} \tag{43}$$

Since, $\forall i \in [k-1, \mu-1]$, we have $\frac{\binom{i}{k}}{\binom{i}{k-1}} = \frac{i-k+1}{k} < \frac{\mu-k+1}{k}$, that is:

$$\binom{i}{k} < \frac{\mu - k + 1}{k} \cdot \binom{i}{k-1} \tag{44}$$

Therefore, we can conclude that:

$$\frac{|\Psi_k(G)|}{|\Psi_{k-1}(G)|} = \frac{\sum_{i=k-1}^{\mu-1} \beta_i \binom{i}{k}}{\sum_{i=k-1}^{\mu-1} \beta_i \binom{i}{k-1}} \tag{45}$$

$$< \frac{\sum_{i=k-1}^{\mu-1} \beta_i \frac{\mu-k+1}{k} \binom{i}{k-1}}{\sum_{i=k-1}^{\mu-1} \beta_i \binom{i}{k-1}} \tag{46}$$

$$< \frac{\mu - k + 1}{k} \tag{47}$$

Hence, we have $\frac{|\Psi_k(G)|}{|\Psi_{k-1}(G)|} < \frac{\mu-k+1}{k}$, therefore:

$$|\Psi_k(G)| < \frac{\mu - k + 1}{k} |\Psi_{k-1}(G)| \tag{48}$$

$$< \frac{\mu - k + 1}{k} \binom{\mu}{k-1} = \binom{\mu}{k} \tag{49}$$

□

Lemma 4.3. Given a graph $G = (V, E)$, $\forall v \in V$, if $|\Psi_k(v, G)| \ge \rho$, then for any $r \in \mathbb{N}$, with $3 \le r \le k$, we have $|\Psi_r(v, G)| \ge \binom{\mu}{r-1}$, where μ denotes the maximum integer such that $\binom{\mu}{k-1} \le \rho$.

Proof. We use mathematical induction to prove this lemma. (1) we first prove it holds when $r = k$:

$$\Psi_r(v, G) = \Psi_k(v, G) \ge \rho \ge \binom{\mu}{k-1} \ge \binom{\mu}{r-1} \tag{50}$$

(2) Next, we show it holds for arbitrary $r \le k$, and we suppose that the lemma holds for $r = s$, so $|\Psi_s(v, G)| \ge \binom{\mu}{s-1}$. Let $\mathcal{Y}_s = \{u \mid u \in V \wedge \exists C \in \Psi_s(v, G), u \in C\}$, Let $G[\mathcal{Y}_s]$ denotes the induced

subgraph of $\mathcal{Y}_s$. Since $|\Psi_s(v)| \geq \binom{\mu}{s-1}$, we have $|\Psi_{s-1}(G[\mathcal{Y}_s])| \geq \binom{\mu}{s-1}$. By Lemma 4.2, we have $|\Psi_{s-2}(G[\mathcal{Y}_s])| \geq \binom{\mu}{s-2}$.

On the other hand, since all vertices in $\mathcal{Y}_s$ are connected with v, so we have $|\Psi_{s-1}(v, G)| = |\Psi_{s-2}(G[\mathcal{Y}_s])| \geq \binom{\mu}{s-2}$. Hence, the lemma holds. □

Lemma 5.2. Given a graph G with n vertices and degeneracy of δ, CCAS costs $O(n \cdot 3^{\delta/3})$ space, and $O(n \cdot 3^{\delta/3} \cdot \delta^2)$ time for each iteration.

Proof. Consider iteratively removing all vertices in the graph. For each root-to-leaf path Γ, we need to process it each time a node it contains is removed. Therefore, each path is processed for $O(\delta)$ node, with each cost $O(\delta)$ time. Since there are $O(n \cdot 3^{\delta/3})$ nodes in the SCT. Therefore the time complexity for CCAS for one iteration is $O(n \cdot 3^{\delta/3})$. □

Lemma 5.3 Given any $t \geq 1$, denote $\bar{\alpha}$ by the vector that has a minimum inner product with the gradient of $Q_G(\alpha^{(t)})$

$$\bar{\alpha} = \left[\bar{\alpha}^{C_1}, \quad \bar{\alpha}^{C_2}, \quad \cdots \quad, \bar{\alpha}^{C_{|\Psi_k(G)|}}\right] = \arg\min_{\beta \in D(G,\, k)} \langle \beta, \nabla Q_G(\alpha^{(t)}) \rangle.$$

We have $\widehat{\alpha}$ is an approximate linear minimizer, i.e.,

$$\langle \widehat{\alpha}, \nabla Q_G(\alpha^{(t)}) \rangle \leq \langle \bar{\alpha}, \nabla Q_G(\alpha^{(t)}) \rangle + \frac{1}{2}\gamma_t \theta \xi(Q_G),$$

where $\gamma_t = \frac{1}{t+1}$, $\theta = 2$,and $\xi(Q_G) = 2\Delta|\Psi_k(G)|$.

Proof. We first use $\nabla_C Q_G(\alpha)$ to denote the projection of $\nabla Q_G(\alpha)$ onto $\mathbb{R}^C$. By a straightforward calculation from proof of Lemma 4.5 [23], the (C, u)-coordinate of $\nabla Q_G(\alpha)$ is

$$2 \cdot \frac{l(u)}{t} = 2 \cdot \sum_{\bar{C} \in \Psi_k(G): u \in \bar{C}} \alpha_u^{\bar{C}} \tag{51}$$

It has been noted in the proof of Lemma 4.5 in [23] that one can consider each k-clique $C \in \Psi_k(G)$ independently and Therefore, for $t \geq 1$, we have:

$$< \widehat{\alpha}, \nabla Q_G(\alpha^{(t)}) > - < \bar{\alpha}, \nabla Q_G(\alpha^{(t)}) > \tag{52}$$

$$= \sum_{i=1}^{|\Psi_k(G)|} < \widehat{\alpha}^i, \nabla_{C^i} Q_G(\alpha^{(t)}) > - < \bar{\alpha}^i, \nabla_{C^i} Q_G(\alpha^{(t)}) > \tag{53}$$

$$= \sum_{i=1}^{|\Psi_k(G)|} < \widehat{\alpha}^i, \nabla_{C^i} Q_G(\alpha^{(t)}) > -\frac{2}{t} \min_{v \in C_i} l^{(t)}(v) \tag{54}$$

Consider the value of $\langle \widehat{\alpha}^i, \nabla_{C^i} Q_G(\alpha^{(t)}) \rangle$, it should be equal to $\frac{2l^{(t)}(v)}{t}$, where v is the first vertex removed from C_i in the t-th iteration. Therefore, we have $l^{(t)}(v) \leq \min_{v' \in C_i} l^{(t)}(v') + |\Psi_k(v', G)| \leq \min_{v' \in C_i} l^{(t)}(v') + \Delta$.

$$< \widehat{\alpha}, \nabla Q_G(\frac{\alpha^{(t)}}{t}) > - < \bar{\alpha}, \nabla Q_G(\frac{\alpha^{(t)}}{t}) > \tag{55}$$

$$= \sum_{i=1}^{|\Psi_k(G)|} < \widehat{\alpha}^i, \nabla_{C^i} Q_G(\alpha^{(t)}) > -\frac{2}{t} \min_{v \in C_i} l^{(t)}(v) \tag{56}$$

$$\leq \frac{2}{t} \sum_{i=1}^{|\Psi_k(G)|} \min_{v \in C_i} l^{(t)}(v) + \Delta - \min_{v \in C_i} l^{(t)}(v) \tag{57}$$

$$= \frac{2}{t}|\Psi_k(G)|\Delta \leq \frac{1}{2}\gamma_t \theta \xi(Q_G), \tag{58}$$

□

Theorem 5.4. CCAS improves the overall running time at least by $(\rho_k^*(G)\sqrt{k})$ over KCCA and by $(\frac{(\frac{\delta}{2})^{k-2}}{\xi})$ over SuperGreedy++.

Proof. To obtain a $(1-\epsilon)$-approximation ratio solution, theoretically, KCCA, SuperGreedy++, and CCAS take $O\left(\frac{1}{\epsilon^2} \cdot \sqrt{k}\Delta|\Psi_k(G)| \cdot \xi \cdot \delta^2 \log \delta\right)$, $O\left(\frac{\Delta \log |\Psi_k(G)|}{\rho_k^*(G) \cdot \epsilon^2} \cdot km \cdot (\frac{\delta}{2})^{k-1}\right)$, and $O\left(\frac{\Delta \log |\Psi_k(G)|}{\rho_k^*(G) \cdot \epsilon^2} \cdot \xi \cdot \delta^3\right)$ time, respectively. Compared to KCCA, our method archives an improvement of $O(\frac{\sqrt{k}\Psi_k(G)\rho_k^*(G)\log\delta}{\log\Psi_k(G)\delta})$, which can be simplified to $O(\sqrt{k}\rho_k^*(G))$. This is because, $\frac{\Psi_k(G)\log\delta}{\log\Psi_k(G)\delta} > 1$, as shown by the function $f(x) = \frac{x}{\log x}$. Since $f'(x) = \frac{\log x - 1}{(\log x)^2}$ is positive for $x > 2$, $f(x)$ is an increasing function, and thus $\frac{\Psi_k(G)}{\log\Psi_k(G)} > \frac{\delta}{\log\delta}$.

Besides, compared to SuperGreedy++, our method achieves an improvement of $O(\frac{km\cdot(\frac{\delta}{2})^{k-1}}{\xi\delta^3})$, which simplifies to $O(\frac{k\cdot(\frac{\delta}{2})^{k-1}}{\xi\cdot\delta})$, since $\delta^2 < m$. This equation can be further simplified to $O(\frac{(\frac{\delta}{2})^{k-2}}{\xi})$. □

Lemma 6.1. Given a graph G, and an approximate k-clique CDS, $\mathcal{S}_k(G)$, for any $k' < k$, we have $\rho_{k'}^*(G) \geq \frac{\binom{\gamma}{k'}}{|\mathcal{S}_k(G)|}$, where γ is the maximum integer such that $\binom{\gamma}{k} \leq \rho_k(\mathcal{S}_k(G)) \cdot |\mathcal{S}_k(G)|$.

Proof. The number of k-cliques contained in approximate k-clique CDS $\mathcal{S}_k(G)$ should be $|\Psi_k(\mathcal{S}_k(G))| = \rho_k(\mathcal{S}_k(G)) \cdot |\mathcal{S}_k(G)|$. Then, we compute the largest integer γ such that $|\Psi_k(\mathcal{S}_k(G))| \geq \binom{\gamma}{k}$. By Lemma 4.2 and mathematical induction, we can conclude that $|\Psi_{k'}(\mathcal{S}_k(G))| \geq \binom{\mu}{k'}$ and $\rho_{k'}^*(G) \geq \rho_{k'}(\mathcal{S}_k(G)) \geq \frac{\binom{\gamma}{k'}}{|\mathcal{S}_k(G)|}$. □

Lemma 6.2 Given a graph G, an approximate k-clique CDS, $\mathcal{S}_k(G)$, for any integers $3 \leq x < k$, we use $\underline{\rho_x}(G)$ and $\underline{\rho_{x+1}}(G)$ to denote the lower bound of optimal densities obtained by the lemma 6.1, then we have $\mu_x \leq \mu_{x+1}$, where μ_x and μ_{x+1} are derived by Lemma 4.3, using $\underline{\rho_x}(G)$ and $\underline{\rho_{x+1}}(G)$.

Proof. Recall that μ_x is the maximum integer such that:

$$\binom{\mu_x}{x-1} \leq \frac{\binom{\gamma}{x}}{|\mathcal{S}_\omega(G)|}, \tag{59}$$

where γ is the maximum integer s.t. $\binom{\gamma}{\omega} \leq \rho_\omega(\mathcal{S}_\omega(G)) \cdot |\mathcal{S}_\omega(G)|$. Based on the formulation (59) we have:

$$\frac{\mu_x!}{(x-1)!(\mu_x - x + 1)!} \leq \frac{\gamma!}{x!(\gamma - x)!|\mathcal{S}_\omega(G)|} \tag{60}$$

$$\frac{\mu_x!}{(\mu_x - x + 1)!} \leq \frac{\gamma!}{x(\gamma - x)!|\mathcal{S}_\omega(G)|} \tag{61}$$

Similarly, μ_{x+1} is the maximum integer satisfy,

$$\frac{\mu_{x+1}!}{(\mu_{x+1} - x)!} \leq \frac{\gamma!}{(x+1)(\gamma - x - 1)!|\mathcal{S}_\omega(G)|} \tag{62}$$

Therefore, to show $\mu_x \leq \mu_{x+1}$, we only need to prove the following formulation:

$$\frac{\mu_x!}{(\mu_x - x)!} \leq \frac{\gamma!}{(x+1)(\gamma - x - 1)!|\mathcal{S}_\omega(G)|} \tag{63}$$

Table 8: Additional two graphs.

Dataset	Category	$\|V\|$	$\|E\|$	δ
web-Google (WG)	Web	916,428	4,322,051	44
Wikipedia (WP)	Hyperlink	3,033,374	43,845,958	175

Here, we can transfer the formulation (45) to (47) by multiplying $(\mu_x - x + 1)$ and $\frac{x(\gamma-x)}{x+1}$ for the left and right hand side of it, respectively. That is to say , if we can prove:

$$\mu_x - x + 1 \leq \frac{x(\gamma - x)}{x+1} \tag{64}$$

$$\mu_x(x+1) + 1 - x^2 \leq x\gamma - x^2 \tag{65}$$

$$\mu_x \leq \frac{x\gamma - 1}{x+1} \tag{66}$$

$$\mu_x \leq \gamma - \frac{\gamma + 1}{x+1} \tag{67}$$

then the formulation (63) can also be proved.

Since $\rho_\omega(\mathcal{S}_\omega(G)) \cdot |\mathcal{S}_\omega(G)| \geq 1$, we have $\gamma \geq \omega > x$. Therefore, we can rewrite γ in terms of $a \cdot (x+1) + b$, where $a \geq 1$ and $0 \leq b \leq k$.

$$\mu_x \leq a(x+1) + b - \frac{ax + a + b + 1}{x+1} \tag{68}$$

$$\mu_x \leq ax + b - \frac{b+1}{x+1} \tag{69}$$

Since μ_x is an integer, we only need to prove:

$$\mu_x \leq ax + b - 1 \tag{70}$$

We prove this by contradiction, where we aim to show that if $\mu_x > ax + b - 1$, it must violate formulation (61). Since the left-hand side of (61) is monotonically non-decreasing with μ_x, we only need to prove the case for $\mu_x = ax + b$, i.e.,:

$$\frac{(ax+b)!}{(ax+b-x+1)!} > \frac{(a\cdot(x+1)+b)!}{x(a\cdot(x+1)+b-x)!|\mathcal{S}_\omega(G)|} \tag{71}$$

Recall that $\rho_\omega(\mathcal{S}_\omega(G)) \cdot |\mathcal{S}_\omega(G)| = |\Psi_\omega(\mathcal{S}_\omega(G))| \geq \binom{\gamma}{\omega}$, so we have $|\mathcal{S}_\omega(G)| \geq \gamma = ax + b$. Hence a stricter inequality can be built:

$$\frac{(ax+b)!}{(ax+b-x+1)!} > \frac{(a\cdot(x+1)+b-1)!}{x(a\cdot(x+1)+b-x)!} \tag{72}$$

$$x \cdot \frac{(ax+b)!}{(ax+b-x+1)!} > \frac{(a\cdot(x+1)+b-1)!}{(a\cdot(x+1)+b-x)!} \tag{73}$$

$$x \cdot \frac{(a\cdot(x+1)+b-x)!}{(ax+b-x+1)!} > \frac{(a\cdot(x+1)+b-1)!}{(ax+b)!} \tag{74}$$

$$x \cdot \prod_{i=ax+b-x+2}^{ax+b-x+a} i > \prod_{i=ax+b+1}^{ax+b+a-1} i \tag{75}$$

$$x \cdot \prod_{i=ax+b-x+2}^{ax+b-x+a} i > \prod_{i=ax+b-x+2}^{ax+b-x+a} (i+x-1) \tag{76}$$

$$x > \prod_{i=ax+b-x+2}^{ax+b-x+a} \frac{(i+x-1)}{i} \tag{77}$$

If $a = 1$, the right-hand side of formulation (77) is 1, which is clearly smaller than x. For the case of $a \geq 2$, we complete our proof by bounding on the right-hand side of formulation (77).

$$\prod_{i=ax+b-x+2}^{ax+b-x+a} \frac{(i+x-1)}{i} \leq \prod_{i=ax+b-x+2}^{ax+b-x+a} \frac{ax+b+1}{ax+b-x+2} \tag{78}$$

$$= \left(\frac{ax+b+1}{ax+b-x+2}\right)^{a-1} \tag{79}$$

$$\leq \left(\frac{ax}{(a-1)\cdot x}\right)^{a-1} \tag{80}$$

$$= \left(1 + \frac{1}{a-1}\right)^{a-1} \tag{81}$$

$$\leq \lim_{a\to\infty} \left(1 + \frac{1}{a-1}\right)^{a-1} = e \tag{82}$$

Hence, we finish our proof since $x \geq 3 > e$. □

C ADDITIONAL EXPERIMENTS

C.1 Additional Datasets.

We present the statistics of the additional two graphs on Table 8 from the different domains. They are available on the Stanford Network Analysis Platform, and Laboratory of Web Algorithmics.

C.2 The k-clique densities

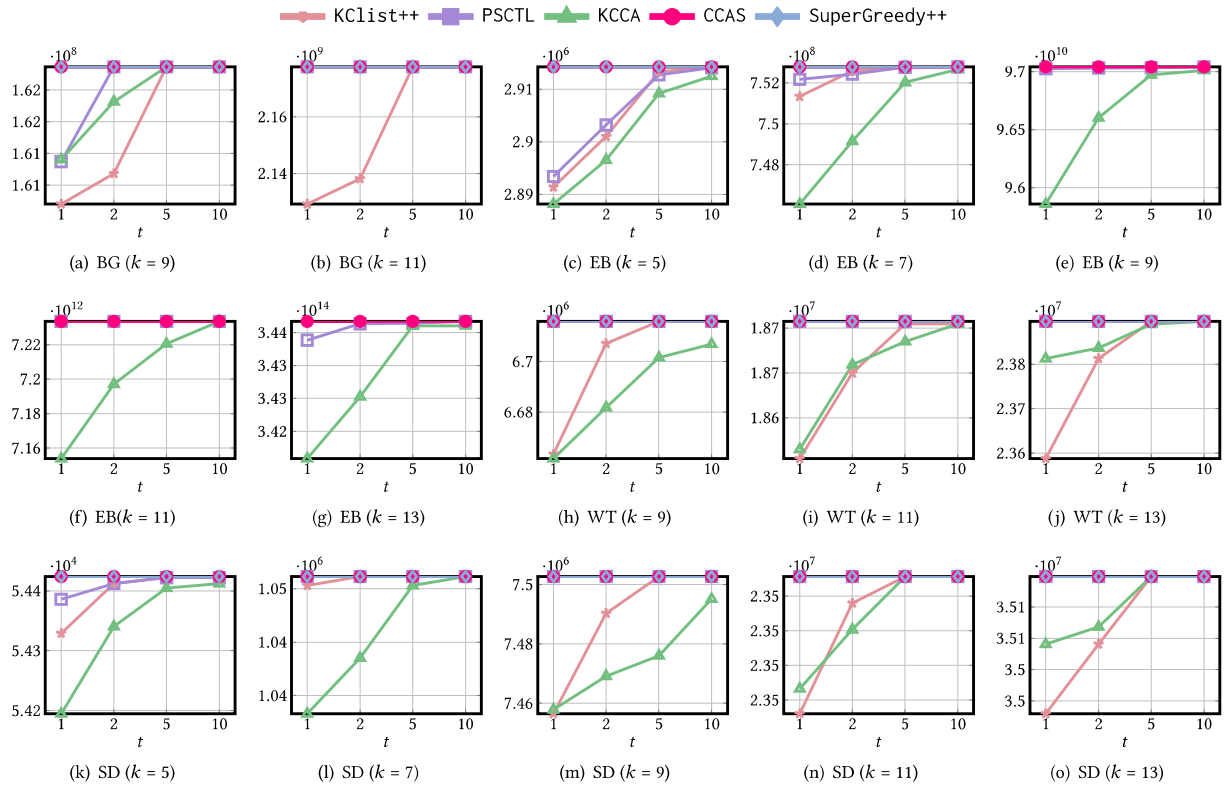
In Figure 12, we report the actual k-clique-densities of the approximation solutions returned by all the CDS algorithms on BG, EB, WT, and SD datasets. We observe that `CCAS` and `SuperGreedy++` always perform better than the `Frank-Wolfe`-based algorithms, and typically, `CCAS` and `SuperGreedy++` only require one iteration to achieve a solution with the same density that other algorithms obtain after ten iterations. In addition, both `KClist++` and `PSCTL` yield comparable performance, and they slightly outperform `KCCA`, since they can enable a more balanced weight distribution among vertices, making them converge faster.

C.3 Effectiveness of the graph reduction

In this experiment, we evaluate the effectiveness of our graph reduction technique, by reporting the number of vertices in the $(k-1)$-core, CDS, and `CCAS` for $k = 7$. We present results only for $k = 7$, as other k values yield similar results. In addition, more results for different k values can be found in our supplementary material. As shown in Table 9, we observe that: (1) the number of vertices in the CDS is remarkably smaller, usually no more than a few thousand; (2) across nearly all datasets, the number of vertices after `HCGR` closely matches the number of vertices in the CDS; (3) the number of vertices in the 6-core remains very high, being up to $100,000\times$ larger than the number of vertices remaining after `HCGR`.

C.4 Case studies

Here, we would conduct the case studies on two real datasets, namely S-DBLP and Yeast. S-DBLP ($|V|$=478, $|E|$=1,086) is a subgraph of the DBLP dataset, which is about the co-authorship network of authors who published at least two DB/DM papers between 2013 and 2015. The triangle densest subgraph is depicted in Figure 13(a). In a triangle (3-clique), every pair of vertices is connected, so the CDS tends to be a near-clique [28, 33, 71]. The researchers involved in this CDS possess a close collaboration relationship: any two researchers have published papers together. In addition, Figure 13(b) shows the 4-clique densest subgraph in the yeast PPI network ($|V|$=1,116, $|E|$=2,148). This CDS shows a protein-subnetwork with the "subcellular localization", "cellular transport", and "protein synthesis" functional classes, which are all 4-cliques. Hence, tasks such as analyzing the conservation and evolution of cellular components [28] can then be performed on this subnetwork.



(a) BG (k = 9) (b) BG (k = 11) (c) EB (k = 5) (d) EB (k = 7) (e) EB (k = 9)

(f) EB(k = 11) (g) EB (k = 13) (h) WT (k = 9) (i) WT (k = 11) (j) WT (k = 13)

(k) SD (k = 5) (l) SD (k = 7) (m) SD (k = 9) (n) SD (k = 11) (o) SD (k = 13)

Figure 12: The k-clique densities of CDS obtained by SuperGreedy++, KClist++, KCCA, PSCTL, and CCAS.

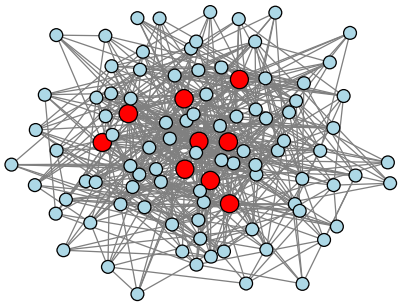
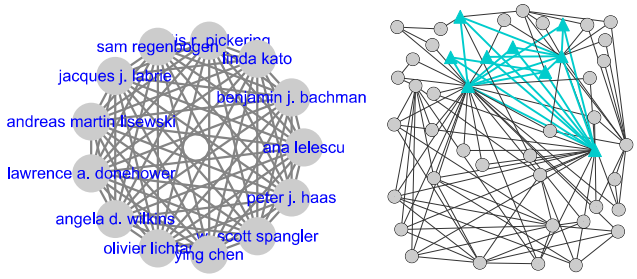


Figure 14: A case study to illustrate the HCGR.



(a) triangle (3-clique) (b) 4-clique (cyan)

Figure 13: The densest subgraphs found in small DBLP and Yeast networks.

Besides, we also present a case study to illustrate our HCGR algorithm, as shown in Figure 14. The k-core-based method does not remove any vertices, whereas HCGR eliminates the blue vertices while retaining the red ones. This demonstrates that HCGR effectively removes nearly all vertices that are not part of the CDS.

C.5 Effect of ϵ

We evaluate the effect of ϵ using five datasets from different domains, where each domain has a dataset, and the values of ϵ are set to 1, 0.1, 0.05, and 0.01, respectively. The experimental results are reported in Table 10, which clearly shows that CCAS outperforms the other algorithms on all datasets. For around half of the datasets, CCAS is over three orders of magnitude faster than its competitors, those results are similar to the results for 0.1 and 0.01 cases, as shown in Table 6.

C.6 Efficiency of finding the CDS's for all k.

We compare the efficiency of our algorithm CCAS-A and others for finding the CDS's for all k values. Specifically, for each graph, we record the running time of the three algorithms PSCTL, KCCA, and CCAS-A as they process the k values from 3 to 25% $\cdot\omega$, 50% $\cdot\omega$, 75% $\cdot\omega$, and 100% $\cdot\omega$ across six datasets in Table 11. We present the results for ϵ from 1 to 0.01. The conclusions are similar to the Experiment 3 in the Section 7.2. Besides, to our best knowledge, CCAS is the first algorithm that can produce solutions of all k values

Table 9: # vertices in the ($k-1$)-core, CDS, and CCAS (k=7).

Datasets	# vertices in 6-core	$\lvert V(\mathcal{D}_k(G))\rvert$	CCAS
BG	1,306	116	268 (43.4 %)
EB	487	231	444 (52.0 %)
WT	2,619	127	551 (23.0 %)
SD	21,686	117	575 (20.3 %)
DP	68,070	114	216 (52.8 %)
HT	21,239	563	563 (100 %)
WG	367,361	116	256 (45.3 %)
WP	1,466,413	178	1,668 (10.7 %)
HW	1,069,126	2209	2209 (100 %)
ZB	3,391,200	268	325 (82.4 %)
UK	11,404,146	944	1,665 (56.7 %)
AC	16,063,729	3,250	3,250 (100 %)
IT	27,201,499	4,279	4,279 (100 %)
FS	36,821,624	141	141 (100 %)

with an approximation ratio of 0.99 for graphs with billions of edges in one hour.

D MORE DISCUSSIONS

The CDS solution has been used in many fundamental graph data management tasks, such as supporting graph visualization [82, 83], community detection (or search) [4, 15, 28, 71, 72], identifying near-cliques [27, 46, 48, 53, 71], and path/reachability queries [17, 42]. In addition, the CDS problem also as a key task in graph data management has gained notable attention at top-tier data management conferences, including VLDB 2019 [28], 2020 [66], 2022 [27], SIGMOD 2023 [36], and 2024 [84].

Finding CDS efficiently is very useful in many graph data mining applications. Specifically, it can help identify research communities in the DBLP network [23,59,60], detect subnetworks with a specific function in the biology network [28, 71, 72] and clusters in senators' networks on US bill voting [71], and discover compact dense subgraphs from e-commerce and social networks [28] when k is relatively small. In addition, identifying CDS with large k also has many applications. As shown in [27, 46, 48, 53, 71], a CDS becomes more akin to a large near-clique as k grows, useful for tasks like finding biologically relevant groups in protein interactions [19, 41, 71, 72], community detection [4, 15, 72], and anomaly detection [31, 67, 79], etc. In many of these applications, finding a "near-clique" is very important since a "near-clique" can be considered a clique in the forming stage or one with missing edges due to data corruption.

Table 10: Effect of ϵ and k. (Processing time (in seconds); Best performers are highlighted in bold.)

Dataset	Method	$k=7$				$k=11$				$k=15$				$k=19$			
		1	0.1	0.05	0.01	1	0.1	0.05	0.01	1	0.1	0.05	0.01	1	0.1	0.05	0.01
HT	PSCTL	79,752	79,752	155,885	268,701	—	—	—	—	—	—	—	—	—	—	—	—
	KCCA	76,191	76,191	150,316	261,231	—	—	—	—	—	—	—	—	—	—	—	—
	CCAS	**8.6**	**8.6**	**8.6**	**8.6**	**8.5**	**8.5**	**8.5**	**8.5**	**8.5**	**8.5**	**8.5**	**8.5**	**8.4**	**8.4**	**8.4**	**8.4**
WG	PSCTL	2.8	2.8	6.9	18.3	1.4	3.2	3.2	4.8	1.3	1.3	1.3	2.5	0.9	0.9	0.9	2.0
	KCCA	7.0	8.2	11.5	16.0	8.3	8.3	8.3	8.6	6.9	6.9	6.9	6.9	6.1	6.1	6.1	6.1
	CCAS	0.2	0.2	0.3	0.3	0.2	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
HW	PSCTL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	KCCA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	CCAS	**52.3**	**52.3**	**52.3**	**52.3**	**52.2**	**52.2**	**52.2**	**52.2**	**53.9**	**53.9**	**53.9**	**53.9**	**51.1**	**51.1**	**51.1**	**51.1**
ZB	PSCTL	89.4	123.5	182.1	507.6	100.4	143.9	143.9	634.7	79.4	112.6	112.6	466.0	65.1	94.9	94.9	404.9
	KCCA	185.5	261.5	261.5	339.2	150.2	176.9	176.9	405.3	105.7	127.4	127.4	323.2	122.6	147.9	147.9	370.6
	CCAS	**11.5**	**11.5**	**11.5**	**12.1**	**9.3**	**9.3**	**9.3**	**10.0**	**7.0**	**7.0**	**7.2**	**8.4**	**5.4**	**5.4**	**5.5**	**6.7**
UK	PSCTL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	KCCA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	CCAS	**78.6**	**78.6**	**78.6**	**78.6**	**20.0**	**20.0**	**20.0**	**20.0**	**18.9**	**18.9**	**18.9**	**18.9**	**17.8**	**17.8**	**17.8**	**17.8**
AC	PSCTL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	KCCA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	CCAS	**5,853**	**5,853**	**5,853**	**5,853**	**6,703**	**6,703**	**6,703**	**6,703**	**6,967**	**6,967**	**6,967**	**6,967**	**7,056**	**7,056**	**7,056**	**7,056**

Table 11: Ruining time of all k values. (Processing time (in seconds); Best performers are highlighted in bold.)

Dataset	Method	$\epsilon=1$				$\epsilon=0.1$				$\epsilon=0.05$				$\epsilon=0.01$			
		25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
DP	PSCTL	5.3	9.3	12.9	16.0	6.2	10.2	13.8	16.9	6.4	10.4	14.0	17.1	9.1	13.2	16.7	19.8
	KCCA	5.0	5.7	6.1	6.3	6.4	7.1	7.5	7.7	7.3	8.0	8.4	9.6	8.1	8.8	9.3	9.5
	CCAS-A	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**	**0.7**
HT	PSCTL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	KCCA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	CCAS-A	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**	**322.4**
WP	PSCTL	3,219	3,855	4,127	4,211	4,798	5,434	5,707	5,797	7,080	7,719	7,995	8,085	21,103	21,789	22,080	22,172
	KCCA	4,557	6,071	6,556	6,702	5,713	7,338	7,869	8,056	6,597	8,254	8,840	9,050	13,970	15,767	16,622	16,853
	CCAS-A	**28.5**	**30.8**	**32.7**	**33.9**	**39.0**	**44.2**	**46.8**	**48.0**	**48.6**	**55.1**	**60.7**	**62.2**	**243.1**	**255.7**	**268.0**	**274.5**
ZB	PSCTL	2,438	2,983	3,441	3,707	3,509	4,053	4,511	4,777	3,788	4,390	4,884	5,150	12,181	12,968	13,569	13,835
	KCCA	3,960	4,206	4,334	4,396	4,781	5,033	5,163	5,230	4,996	5,258	5,393	5,468	9,949	10,273	10,453	10,581
	CCAS-A	**66.0**	**103.0**	**136.9**	**161.7**	**66.0**	**103.0**	**136.7**	**161.7**	**72.8**	**114.1**	**151.1**	**176.7**	**106.0**	**166.6**	**217.3**	**247.5**
UK	PSCTL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	KCCA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	CCAS-A	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**	**456.6**
FS	PSCTL	119,740	185,679	239,801	280,842	125,544	191,483	245,604	286,646	135,619	201,558	255,679	296,721	201,621	267,560	321,681	—
	KCCA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	CCAS-A	**2,167**	**2,168**	**2,169**	**2,169**	**2,167**	**2,168**	**2,169**	**2,169**	**2,273**	**2,274**	**2,275**	**2,276**	**3,090**	**3,091**	**3,092**	**3,092**