智能网络与优化实验室

Intelligent Network and Optimization Laboratory, Renmin University

Colorful h-star Core Decomposition

Author: Sen Gao, Rong-Hua Li, Hongchao Qin, Hongzhi Chen, Ye Yuan, Guoren Wang

2022 IEEE 38th International Conference on Data Engineering (ICDE)

Xiaojia Xu

Applications

- community search

- locating influential nodes

- Keyword extraction from text

- real-time story identification

[2] J. Chen and Y. Saad, "Dense subgraph extraction with application to community detection," *IEEE Transactions on knowledge and data engineering*, vol. 24, no. 7, pp. 1216–1230, 2010.

[3] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, "Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees," in *KDD*, 2013.

[4] X. Huang, L. V. Lakshmanan, and J. Xu, "Community search over big graphs: Models, algorithms, and opportunities," in *ICDE*, 2017.

[5] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, "Identification of influential spreaders in complex networks," *Nature Physics*, vol. 6, pp. 888–893, 2010.

[6] F. D. Malliaros, M.-E. G. Rossi, and M. Vazirgiannis, "Locating influential nodes in complex networks," *Scientific reports*, vol. 6, no. 1, pp. 1–10, 2016.

[7] M. Vazirgiannis, "Graph of words: Boosting text mining tasks with graphs," in *WWW Companion*, 2017.

[8] A. Tixier, F. Malliaros, and M. Vazirgiannis, "A graph degeneracy-based approach to keyword extraction," in *EMNLP*, 2016.

[9] A. Angel, N. Koudas, N. Sarkas, and D. Srivastava, "Dense subgraph maintenance under streaming edge weight updates for real-time story identification," *PVLDB*, vol. 5, no. 6, pp. 574–585, 2012.

[10] L. Chang and L. Qin, "Cohesive subgraph computation over large sparse graphs," in *ICDE*, 2019.

Cohesive Subgraph Models

- **k-core**
- **k-plex**
- **k-truss**
- **h-clique**

[11] W. Cui, Y. Xiao, H. Wang, and W. Wang, "Local search of communities in large graphs," in *SIGMOD*, 2014.

[12] R.-H. Li, J. Yu, and R. Mao, "Efficient core maintenance in large dynamic graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2453–2465, 2014.

[13] J. Wang and J. Cheng, "Truss decomposition in massive networks," *PVLDB*, vol. 5, no. 9, pp. 812–823, 2012.

[14] X. Huang, H. Cheng, L. Qin, W. Tian, and J. X. Yu, "Querying k-truss community in large and dynamic graphs," *SIGMOD*, 2014.

[15] A. Conte, D. Firmani, C. Mordente, M. Patrignani, and R. Torlone, "Fast enumeration of large k-plexes," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 115–124.

[16] M. Danisch, O. Balalau, and M. Sozio, "Listing k-cliques in sparse real-world graphs," in *WWW*, P. Champin, F. Gandon, M. Lalmas, and P. G. Ipeirotis, Eds., 2018.

[17] R. Li, S. Gao, L. Qin, G. Wang, W. Yang, and J. X. Yu, "Ordering heuristics for k-clique listing." *Proc. VLDB Endow.*, vol. 13, no. 11, pp. 2536–2548, 2020.

[18] S. Jain and C. Seshadhri, "The power of pivoting for exact clique counting," in *WSDM*, J. Caverlee, X. B. Hu, M. Lalmas, and W. Wang, Eds., 2020.

Cohesive Subgraph Models

- **k-edge connected subgraph**

[19] R. Zhou, C. Liu, J. X. Yu, W. Liang, B. Chen, and J. Li, "Finding maximal k-edge-connected subgraphs from a large graph," in *EDBT*, 2012.

[20] L. Chang, J. X. Yu, L. Qin, X. Lin, C. Liu, and W. Liang, "Efficiently computing k-edge connected components via graph decomposition," in *SIGMOD*, 2013.

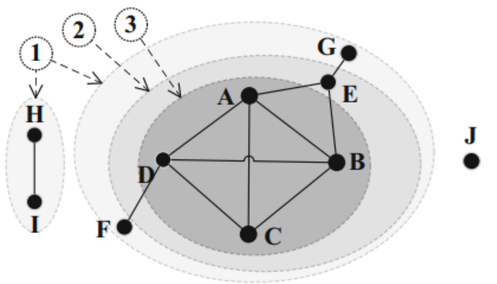
[2] J. Chen and Y. Saad, "Dense subgraph extraction with application to community detection," *IEEE Transactions on knowledge and data engineering*, vol. 24, no. 7, pp. 1216–1230, 2010.

[3] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, "Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees," in *KDD*, 2013.

- **densest subgraph**

Cohesive Subgraph Models

- **k-core**

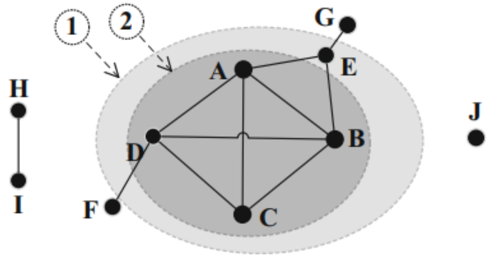


(a) k-cores

Core number	Vertices
0	J
1	F, G, H, I
2	E
3	A, B, C, D

(b) Core numbers

- **k-truss**

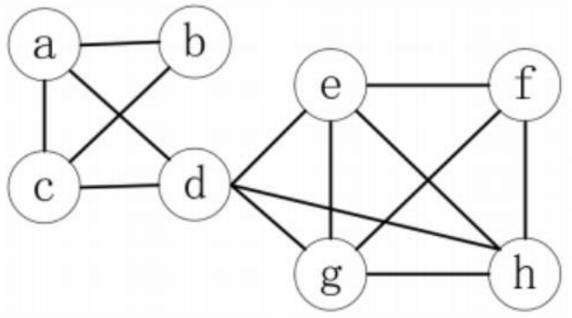


(a) k-trusses

Truss number	Edges
4	(A, B), (A, C), (A, D), (B, C), (B, D), (C, D)
3	(A, E), (B, E)
2	(D, F), (E, G), (H, I)

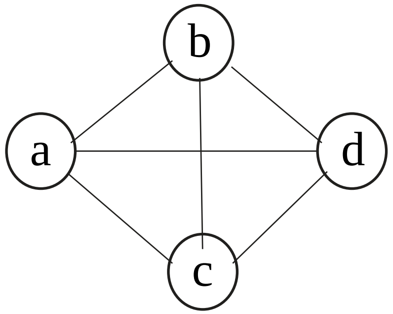
(b) Trusses

- **k-plex**



{a, b, c, d}和{e, f, g, h} 是 2-plex或k-plex(k≥2)

- k-clique
- **k-ECC**

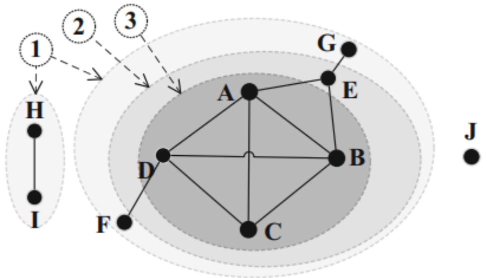


{a, b, c, d} 是 4-clique
{a, b, c, d} 是 3-ECC

[10] L. Chang and L. Qin, "Cohesive subgraph computation over large sparse graphs," in *ICDE*, 2019.
[] Fang Y , Huang X , Qin L , et al. A survey of community search over big graphs[J]. Springer Berlin Heidelberg, 2020.

Cohesive Subgraph Models

- **k-core** $O(m)$

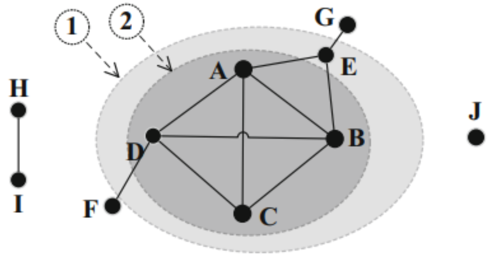


(a) *k*-cores

Core number	Vertices
0	J
1	F, G, H, I
2	E
3	A, B, C, D

(b) Core numbers

- **k-truss** $O(m^{1.5})$



(a) *k*-trusses

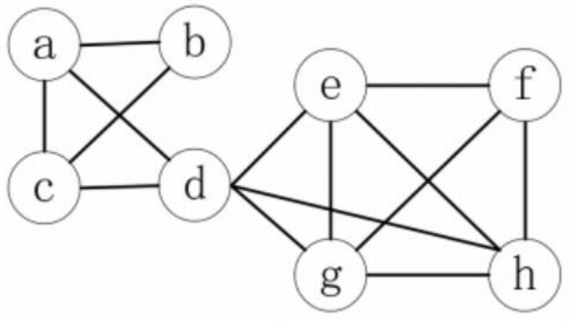
Truss number	Edges
4	(A, B), (A, C), (A, D), (B, C), (B, D), (C, D)
3	(A, E), (B, E)
2	(D, F), (E, G), (H, I)

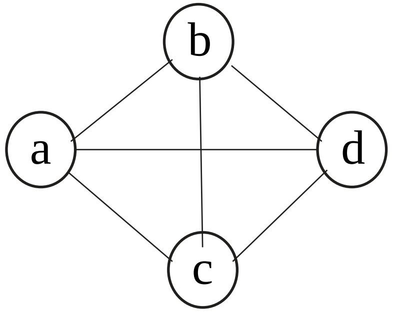
(b) Trusses

- **k-plex** $NP-complete$

- **k-clique** $NP-complete$

- **k-ECC** $O(h*l*m)$





{a, b, c, d}和{e, f, g, h} 是 2-plex或k-plex(k≥2)

{a, b, c, d} 是 4-clique {a, b, c, d} 是 3-ECC

Cohesive Subgraph Models

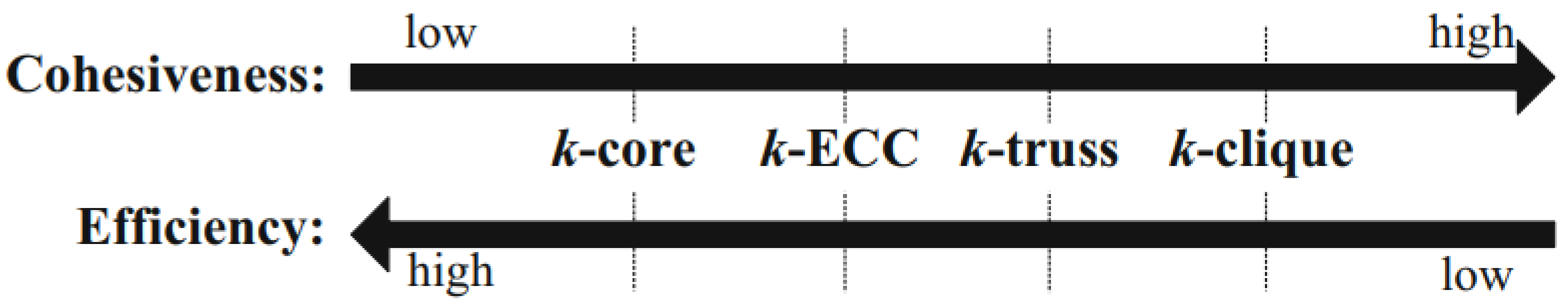


Table 2 Efficiency comparison for different metrics

Datasets	k-core (s)	k-ECC (s)	k-truss	k-clique
Email-Enron	0.2	0.8	5	201 s
Google	8.9	40.8	65	> 24 h
Livejournal	85	854	1726	> 24 h
Wise	553 s	5764	32,221	> 24 h

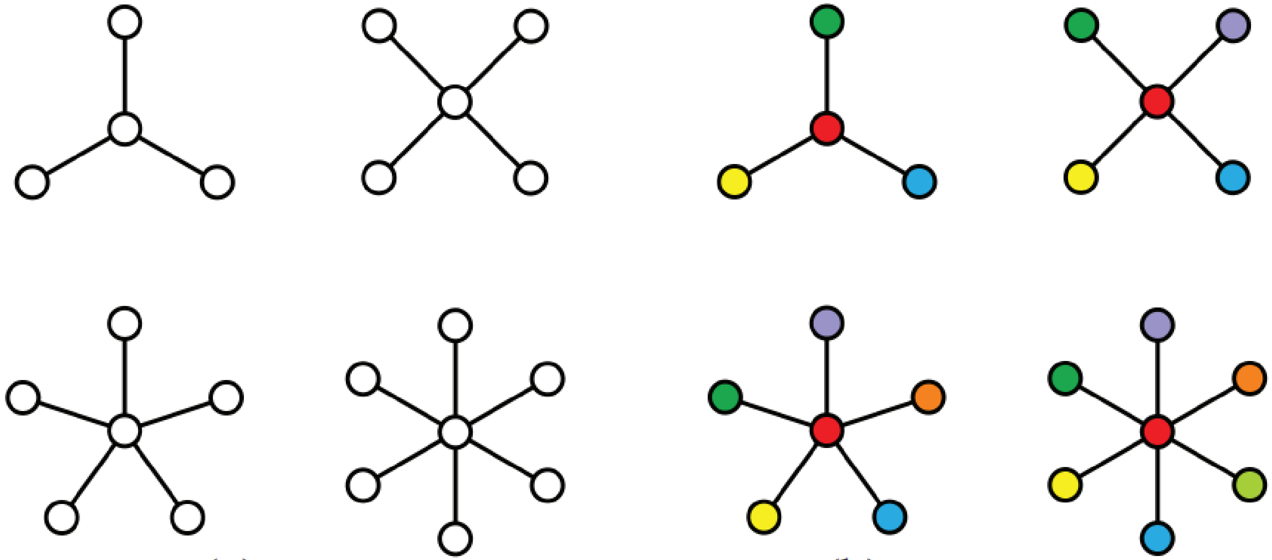
Related Work

- **h-clique k core, is defined as a maximal subgraph, in which each node participates in at least k h-cliques** [16][22]
  (h-clique is a subgraph with h nodes such that each pair of nodes is connected with an edge)

- **traditional densest subgraph** $\xrightarrow{\text{a higher-order variant}}$ - **h-clique densest subgraph**[23][25]

[16] M. Danisch, O. Balalau, and M. Sozio, "Listing k-cliques in sparse real-world graphs," in *WWW*, P. Champin, F. Gandon, M. Lalmas, and P. G. Ipeirotis, Eds., 2018.
[22] Y. Fang, K. Yu, R. Cheng, L. V. Lakshmanan, and X. Lin, "Efficient algorithms for densest subgraph discovery," *Proc. VLDB Endow.*, vol. 12, no. 11, pp. 1719–1732, 2019.
[23] C. E. Tsourakakis, "The k-clique densest subgraph problem," in *WWW*, 2015.
[25] B. Sun, M. Danisch, T. Chan, and M. Sozio, "Kclist++: A simple algorithm for finding k-clique densest subgraphs in large graphs," *Proceedings of the VLDB Endowment (PVLDB)*, 2020.

Background



(a) Stars (b) Colorful stars

Fundamentals

$G = (V, E)$: **an undirected graph**

$N_u(G)$: **the set of neighbor nodes of** u **in** G

$d_u(G)$: **the degree of** u **in** G **,** $d_u(G) = |N_u(G)|$

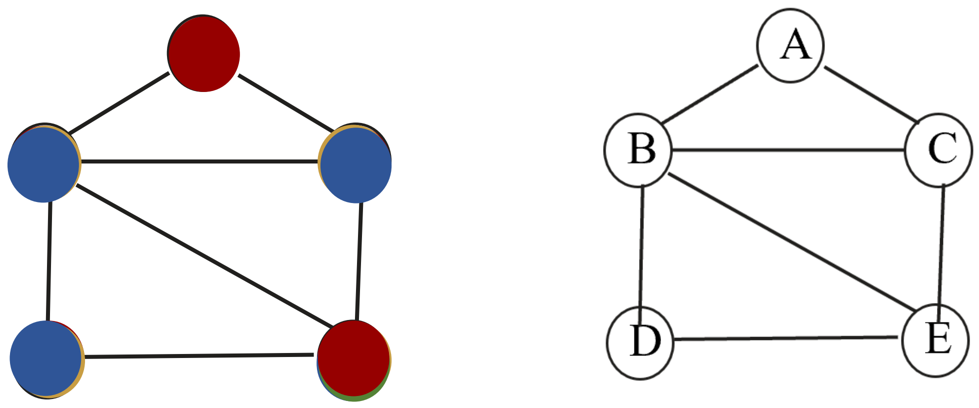
$H = (V_H, E_H)$: **an induced subgraph of** G

$$V_H \subset V, E_H = \{(u,v)|(u,v) \subset E, u \in V_H, v \in V_H\}$$

h-star R : **a tree, with one internal or central node having degree** $h-1$ **and the other** $h-1$ **nodes having degree** 1

a node u participates in $\binom{d_u(G)}{h-1}$ h-stars which are centered on u if $d_u(G) \geq h-1$.

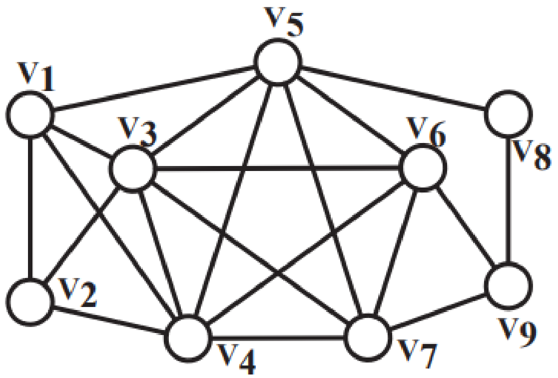
Graph coloring



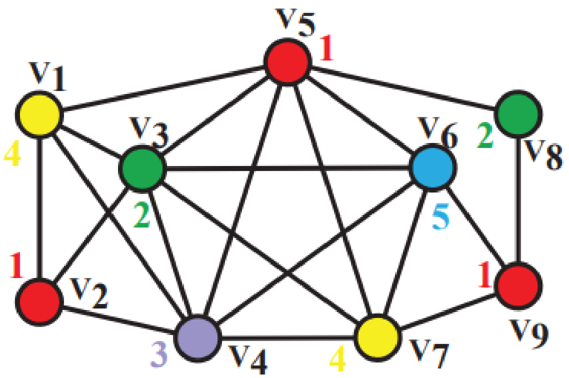
3 colors

[26] W. Hasenplaugh, T. Kaler, T. B. Schardl, and C. E. Leiserson, "Ordering heuristics for parallel graph coloring," in *SPAA*, 2014.
[27] L. Yuan, L. Qin, X. Lin, L. Chang, and W. Zhang, "Effective and efficient dynamic graph coloring," *PVLDB*, vol. 11, no. 3, pp. 338–351, 2017.

Fundamentals



(a) An undirected graph



(b) The colored graph

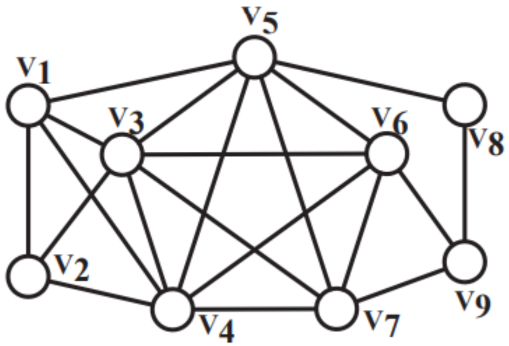
Fundamentals

Definition 1 (Colorful h-star). *Given a colored graph $G = (V, E)$ and an integer $h \geq 2$, an h-star $\mathcal{S}$ in G is colorful if any pair of nodes $u, v \in \mathcal{S}$ have different color values.*

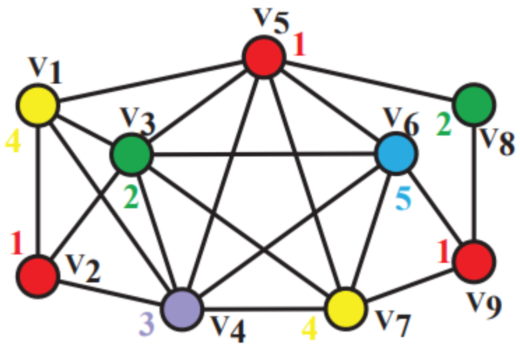
Definition 2 (Pattern degree). *Given a colored graph $G = (V, E)$ and an integer h. Let $\mathcal{R}$, $\mathcal{S}$, Ψ be an h-star, a colorful h-star and an h-clique respectively. The h-star degree of a node u in G, denoted by $d_u(G, \mathcal{R})$, is the number of h-stars centered on u; the colorful h-star degree of u, denoted by $d_u(G, \mathcal{S})$, is the number of colorful h-stars centered on u; and the h-clique degree of u, denoted by $d_u(G, \Psi)$, is the number of h-cliques that u participates in.*

Fundamentals

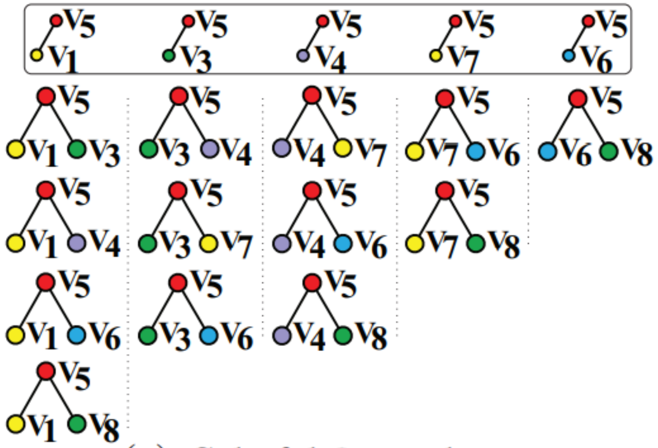
Example 2. *Reconsider the graph G in Fig. 2(a). Suppose $h = 3$. For node v_5, we can easily derive that its h-star degree is $d_{v_5}(G, \mathcal{R}) = \binom{d_{v_5}(G)}{h-1} = 15$. Also, v_5 participates in 8 3-cliques listed in Fig.3(b), thus its h-clique degree $d_{v_5}(G, \Psi)$ is equal to 8. After coloring G in Fig. 2(b), the colorful h-star degree of v_5, denoted by $d_{v_5}(G, \mathcal{S})$, is 13, because there exist 13 colorful 3-stars which are centered on v_5 as enumerated in Fig. 3(a).*



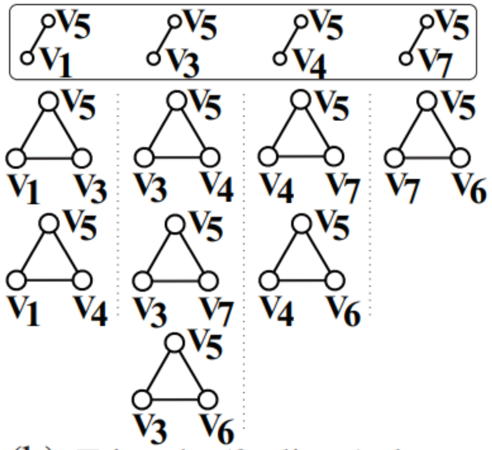
(a) An undirected graph



(b) The colored graph



(a) Colorful 3-star degree



(b) Triangle (3-clique) degree

COLORFUL h-STAR COUNTING AND UPDATING

Algorithm 1: The DP-based Counting Algorithm

```
Input: A graph G and a node u
Output: The colorful h-star degree d_u(G, S)
color[1, ··· , n] ← GreedyColoring(G);
for i = 1 to χ do
    Group(i) ← {v | v ∈ N_u(G), color(v) = i};
    cnt(i) ← |Group(i)|;
d_u(G, S) ← DP(χ, h − 1);
return d_u(G, S);
Procedure GreedyColoring(G)
Let π′ be any ordering on nodes;
flag(i) ← −1 for i = 1, ··· , χ;
for each node v ∈ π′ in order do
    for u ∈ N_v(G) do
        flag(color(u)) ← v;
    c ← min{i | i > 0, flag(i) ≠ v};
    color(v) ← c;
return color(v) for all v ∈ G;
Procedure DP(c, h̄)
for i = 0 to c do
    for j = 0 to h̄ do
        if j = 0 then dp(i, j) ← 1;
        else if i < j then dp(i, j) ← 0;
        else dp(i, j) ← dp(i − 1, j − 1) × cnt(i) + dp(i − 1, j);
return dp(c, h̄);
```

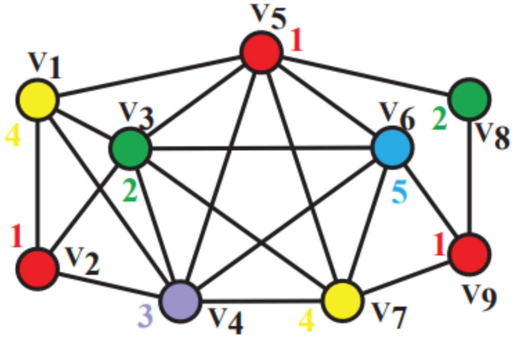
The DP algorithm

$$\mathsf{DP}(i,j) = \mathsf{DP}(i-1, j-1) \times \mathsf{cnt}(i) + \mathsf{D}$$

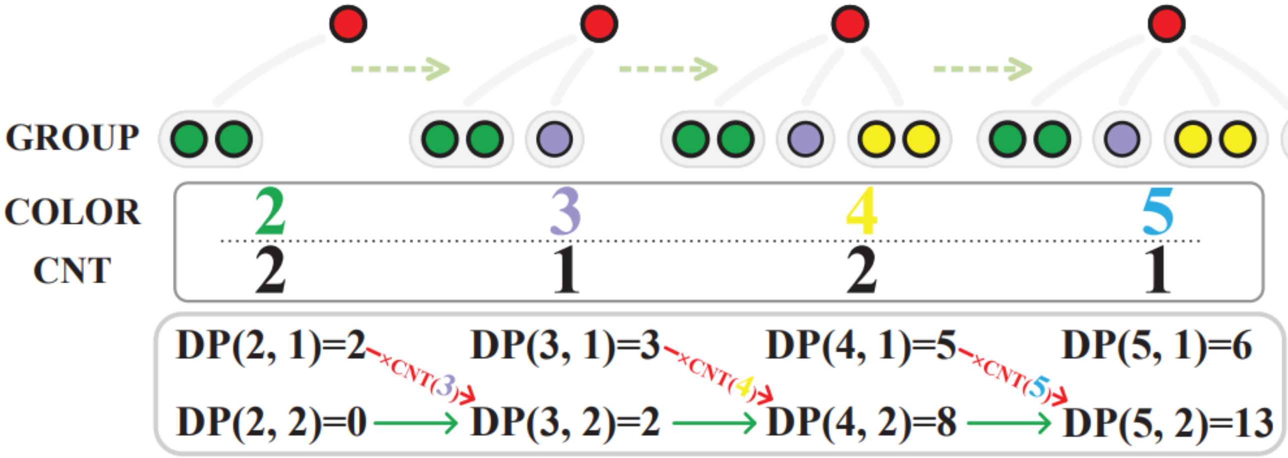
for all $i \in [1, \cdots, \chi], j \in [1, \cdots, h], i \geq j$.

The base cases can be set as follows:

$$\begin{cases} \mathsf{DP}(i,0) = 1, & \text{for all } i \in [0, \cdots, \chi] \\ \mathsf{DP}(i,j) = 0, & \text{for all } i \in [0, \cdots, \chi], j \in \end{cases}$$



(b) The colored graph



The DP algorithm

Theorem 1. *Given a graph G, a node u and an integer h, Algorithm 1 computes the colorful h-star degree of u in $O(h \times \min\{\chi, d_u(G)\})$ time using $O(h)$ space, where χ is the maximum color value of all nodes in G.*

$$\mathsf{DP}(j) = \mathsf{DP}(j-1) \times \mathsf{cnt}(i) + \mathsf{DP}(j), \tag{3}$$

for all $i \in [0..\chi], j \in [0..h], i \geq j$.

The updating algorithm

$$
\begin{aligned}
P(A) &: \forall\{v,w\}(\{v,w\} \subseteq A \rightarrow \mathsf{color}(v) \neq \mathsf{color}(w)) \\
Q(A) &: \forall v(v \in A \rightarrow \mathsf{color}(v) \neq \chi') \\
\bar{Q}(A) &: \exists v(v \in A \wedge \mathsf{color}(v) = \chi') \\
\mathcal{F}(i) &= |\{A|A \subseteq N_u(G), |A| = i, P(A), Q(A)\}| \\
\mathcal{G}(i) &= |\{A|A \subseteq N_u(G), |A| = i, P(A), \bar{Q}(A)\}| \\
\mathcal{DP}(i) &= |\{A|A \subseteq N_u(G), |A| = i, P(A)\}|.
\end{aligned}
$$

The updating algorithm

Theorem 2. *After removing a neighbor v, the colorful h-star degree of u can be updated by the following DP equation*

$$\begin{cases} \mathcal{G}(i) \leftarrow \mathcal{F}(i-1) \times \mathsf{cnt}(\mathsf{color}(v)) \\ \mathcal{DP}(i) \leftarrow \mathcal{F}(i) + \mathcal{G}(i) \end{cases} \tag{4}$$

for all $i \in [1, \cdots, h]$. The updated $d_u(G \backslash v, \mathcal{S})$ is equal to $\mathcal{DP}(h-1)$.

The updating algorithm

Algorithm 2: The Counting and Updating Algorithm

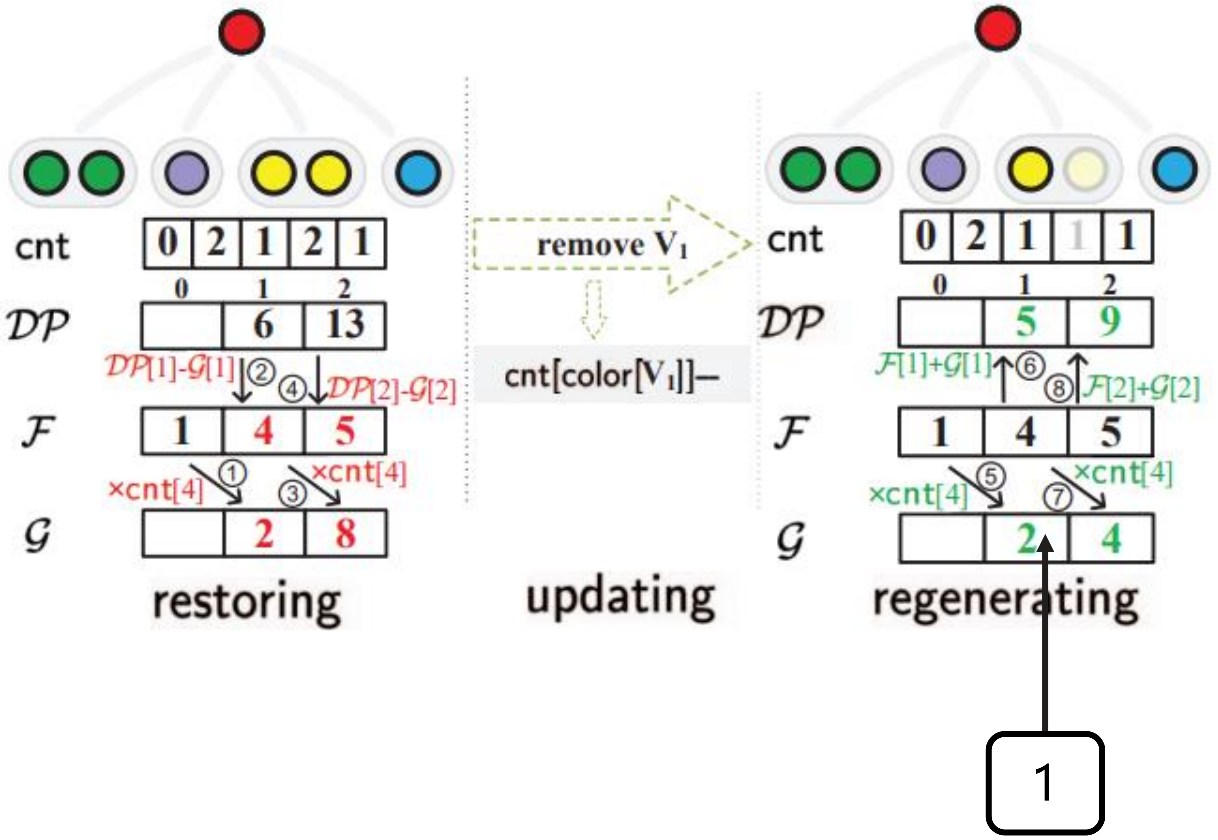
Input: A graph G, a node u and its neighbor v
Output: The updated colorful h-star degree $d_u(G\backslash v, \mathcal{S})$

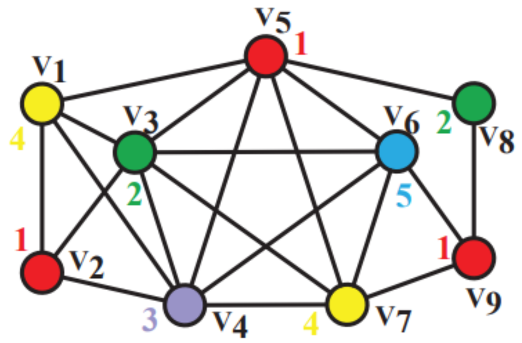
$d_u(G, \mathcal{S}) \leftarrow \mathsf{Counting}(u)$;
Delete v from G;
$d_u(G\backslash v, \mathcal{S}) \leftarrow \mathsf{Updating}(\mathcal{DP}, v)$;

Procedure $\mathsf{Counting}(u)$
$\mathcal{F}(\cdot) \leftarrow 0; \mathcal{F}(0) \leftarrow 1$;
$\chi' \leftarrow 1$;
for $i = 2$ **to** χ **do**
  for $j = h$ **to** 1 **do**
    $\mathcal{F}(j) \leftarrow \mathcal{F}(j-1) \times \mathsf{cnt}(i) + \mathcal{F}(j)$;
for $j = 1$ **to** h **do**
  $\mathcal{G}(j) \leftarrow \mathcal{F}(j-1) \times \mathsf{cnt}(1)$;
  $\mathcal{DP}(j) \leftarrow \mathcal{F}(j) + \mathcal{G}(j)$;
return $\mathcal{DP}(h-1)$;

Procedure $\mathsf{Updating}(\mathcal{DP}, v)$
$\chi' \leftarrow \mathsf{color}(v)$, $\mathcal{F}(0) \leftarrow 1$;
for $i = 1$ **to** h **do**
  $\mathcal{G}(i) \leftarrow \mathcal{F}(i-1) \times \mathsf{cnt}(\mathsf{color}(v))$;
  $\mathcal{F}(i) \leftarrow \mathcal{DP}(i) - \mathcal{G}(i)$;
$\mathsf{cnt}(\mathsf{color}(v)) \leftarrow \mathsf{cnt}(\mathsf{color}(v)) - 1$;
for $i = 1$ **to** h **do**
  $\mathcal{G}(i) \leftarrow \mathcal{F}(i-1) \times \mathsf{cnt}(\mathsf{color}(v))$;
  $\mathcal{DP}(i) \leftarrow \mathcal{F}(i) + \mathcal{G}(i)$;
return $\mathcal{DP}(h-1)$;

The updating algorithm



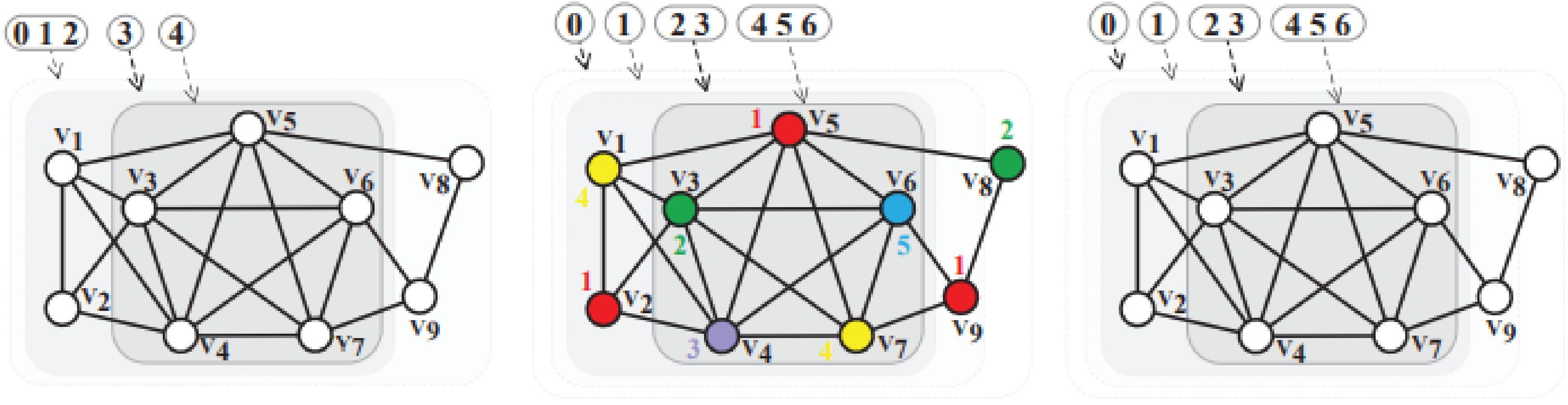


(b) The colored graph

The updating algorithm

Theorem 3. *Given a graph G, an integer h and a node u, after removing any neighbor of u, the updating procedure of Algorithm 2 updates its colorful h-star degree in $O(h)$ time using $O(\min\{\chi, d_u(G)\} + h)$ space, where χ is the maximum color value of all nodes in G.*

COLORFUL h-STAR CORE DECOMPOSITION



(a) k-cores (b) $(k, \mathcal{S})$-cores (c) (k, Ψ)-cores

Definition 3 (Colorful h-star k core). *Given a colored graph G, an integer k and the size h of a colorful star $\mathcal{S}$, a colorful h-star k core, or $(k, \mathcal{S})$-core of G, denoted by C_k^s, is a maximal subgraph H such that $\forall u \in V_H$, $d_u(H, \mathcal{S}) \geq k$.*

COLORFUL h-STAR CORE DECOMPOSITION

Problem 1 (COLORFUL h-STAR CORE DECOMPOSITION). *Given a graph G and an integer h, the colorful h-star core decomposition of G is a problem of computing the colorful h-star core numbers for all nodes in G.*

COLORFUL h-STAR CORE DECOMPOSITION

Algorithm 3: The Colorful h-Star Core Decomposition

Input: A graph G and an integer h

Output: $c_u(G, \mathcal{S})$ for each node $u \in V$

for $u = 1$ **to** $|V|$ **do**
  $d_u(G, \mathcal{S}) \leftarrow \mathsf{Counting}(u)$; // Algorithm 2

Sort nodes of G in a non-decreasing order of their colorful h-star degrees;
$H \leftarrow G$; max_core $\leftarrow 0$;
for $i = 1$ **to** $|V|$ **do**
  $u \leftarrow \arg\min_{v \in V_H} d_v(H, \mathcal{S})$;
  if $d_u(H, \mathcal{S}) >$ max_core **then**
    max_core $\leftarrow d_u(H, \mathcal{S})$;
  $c_u(G, \mathcal{S}) \leftarrow$ max_core;
  for *each* $w \in N_u(H)$ **do**
    $d_w(H \backslash u, \mathcal{S}) \leftarrow \mathsf{Updating}(\mathcal{DP}, w)$; // Algorithm 2
  Delete u from H;
  Resort the nodes of H;

return $c_u(G, \mathcal{S})$ for each node $u \in V$;

COLORFUL h-STAR CORE DECOMPOSITION

Theorem 4. *Given a graph G and an integer h, Algorithm 3 computes the colorful h-star core decomposition in $O(h \times m)$ time using $O(hn + m)$ space.*

SPEED UP h-CLIQUE DENSEST SUBGRAPH MINING

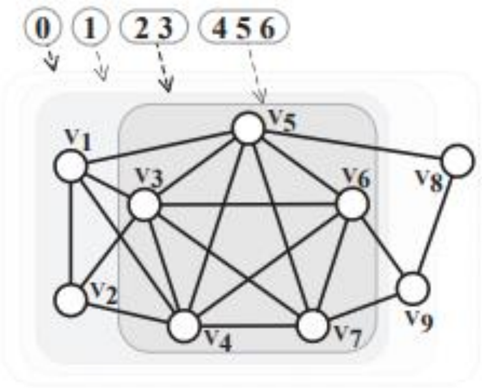
Definition 4 (h-CLIQUE DENSITY). *Given a graph G and an integer h, for any induced subgraph H, $V_H \subseteq V_G$, its h-clique density is defined as $\sigma_h(H) = \frac{c_h(H)}{|V_H|}$.*

Problem 2 (H-CLIQUE-DS-PROBLEM). *Given a graph G and an integer h, find a subgraph H^* that achieves the largest h-clique density among all subgraphs of G, and let $\sigma_h^* = \sigma_h(H^*)$.*

[22] Y. Fang, K. Yu, R. Cheng, L. V. Lakshmanan, and X. Lin, "Efficient algorithms for densest subgraph discovery," *Proc. VLDB Endow.*, vol. 12, no. 11, pp. 1719–1732, 2019.

SPEED UP h-CLIQUE DENSEST SUBGRAPH MINING

Definition 5 (h-CLIQUE CORE). *([22]) Given a graph G, an integer k, and an h-clique Ψ, the h-clique k core, or (k, Ψ)-core of G, denoted by C_k^{Ψ}, is a maximal subgraph such that $\forall u \in C_k^{\Psi}$, $d_u(C_k^{\Psi}, \Psi) \geq k$.*



(c) (k, Ψ)-cores

SPEED UP h-CLIQUE DENSEST SUBGRAPH MINING

Theorem 5. *([22]) Given a graph G and an h-clique Ψ, the (δ^{Ψ}, Ψ)-core $C^{\Psi}_{\delta^{\Psi}}$ is a $\frac{1}{h}$-approximation solution to h-clique densest subgraph problem, such that $\sigma_h(C^{\Psi}_{\delta^{\Psi}}) \geq \frac{1}{h} \times \sigma^*_h$.*

We denote the h-clique core number of a node $u \in V$ by $c_u(G, \Psi)$, which is the largest k such that there exists an h-clique k core containing u. The maximum h-clique core number of a graph G, denoted by δ^{Ψ}, is the maximum value of h-core numbers among all nodes.

Existing approximation algorithms

- Core-based approximation algorithm.

  CoreApp algorithm[22]

- Sampling-based approximation algorithm.

  SeqSamp++ algorithm[25]

[22] Y. Fang, K. Yu, R. Cheng, L. V. Lakshmanan, and X. Lin, "Efficient algorithms for densest subgraph discovery," *Proc. VLDB Endow.*, vol. 12, no. 11, pp. 1719–1732, 2019.

[25] B. Sun, M. Danisch, T. Chan, and M. Sozio, "Kclist++: A simple algorithm for finding k-clique densest subgraphs in large graphs," *Proceedings of the VLDB Endowment (PVLDB)*, 2020.

The colorful h-star core based algorithm

Theorem 6. *Given a graph G, and an h-clique Ψ, the h-clique θ core C_θ^Ψ is contained in the $(w-1)$-core, where w is the size of a large clique of G and $\theta = \binom{w-1}{h-1}$.*

Theorem 7. *Given a graph G, and an integer h, the colorful h-star θ core $C_\theta^\mathcal{S}$ is contained in the $(w-1)$-core of G, where w is the size of a large clique of G, and $\theta = \binom{w-1}{h-1}$.*

$$\binom{w-1}{h-1} = c_v(C_\theta^\mathcal{S}, \mathcal{S}) \leq d_v(C_\theta^\mathcal{S}, \mathcal{S}) \leq d_v(C_\theta^\mathcal{S}, \mathcal{R}) = \binom{d_v(C_\theta^\mathcal{S})}{h-1}.$$

$$G \supseteq C_{w-1} \supseteq C_\theta^\mathcal{S} \supseteq C_{\delta^\Psi}^\Psi.$$

[31] R. A. Rossi, D. F. Gleich, and A. H. Gebremedhin, "Parallel maximum clique algorithms with applications to network analysis," *SIAM Journal on Scientific Computing*, vol. 37, no. 5, pp. C589–C616, 2015.

The colorful h-star core based algorithm

Algorithm 4: The Colorful h-Star Core Reduction

Input: A graph G and an integer h

Output: The $(\theta, \mathcal{S})$-core

$\Psi \leftarrow$ compute a large clique using a greedy algorithm proposed in [31];
$\omega \leftarrow |V_\Psi|, \theta \leftarrow c_h(\Psi) = \binom{\omega-1}{h-1}$;
$C_{w-1} \leftarrow$ compute the $(\omega - 1)$-core using the peeling algorithm [30];
$(\theta, \mathcal{S})$-core $\leftarrow$ ColorfulStarCore(C_{w-1}, h, θ);
return $(\theta, \mathcal{S})$-core;

Procedure ColorfulStarCore (H, h, k)
for $u = 1$ **to** $|V_H|$ **do**
    $d_u(H, \mathcal{S}) \leftarrow$ Counting(u);
Let Q be an empty queue;
for *each* $v \in H$ **do**
    if $d_u(H, \mathcal{S}) < k$ **then**
        Push v to Q;
while $Q \neq \emptyset$ **do**
    Pop a node u from Q;
    for *each* $v \in N_u(H)$ **do**
        $d_v(H \backslash u, \mathcal{S}) \leftarrow$ Updating$(\mathcal{DP}, v)$;
        if $d_v(H \backslash u, \mathcal{S}) < k$ **then**
            Push v to Q;
    Delete u from H;
return H;

Experimental Setup

TABLE I
DATASETS

Dataset	$n = \|V\|$	$m = \|E\|$	χ	$d_{\max}$
Nasasrb	54,870	1,311,227	38	275
Pkustk	87,804	2,565,054	54	131
Buzznet	101,163	2,763,066	62	64,289
Pwtk	217,891	5,653,221	42	179
DBLP	317,080	1,049,866	114	343
MsDoor	404,785	9,378,650	42	76
Digg	770,799	5,907,132	66	17,643
LDoor	909,537	20,770,807	42	76
Skitter	1,694,616	11,094,209	71	35,455
Orkut	2,997,166	106,349,209	79	27,466
LiveJournal	4,847,572	42,851,237	324	20,333

Results of the colorful h-star core decomposition

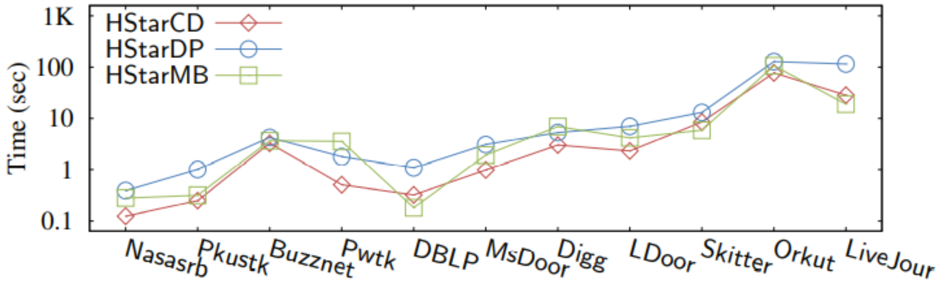


Fig. 7. Running time of HStarCD, HStarDP and HStarMB ($h = 6$)

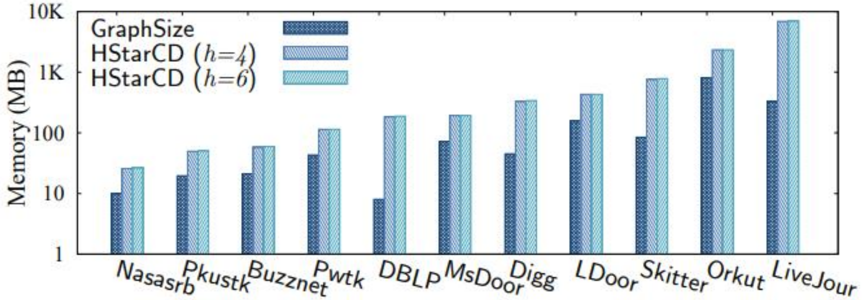
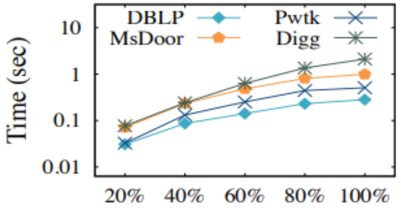
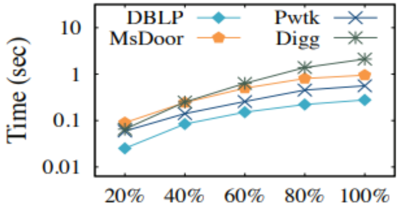


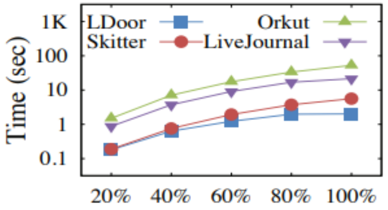
Fig. 9. Memory overhead of HStarCD on different datasets ($h = 4, 6$)



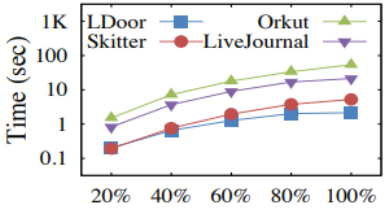
(a) Medium-sized graphs ($h = 4$)



(b) Medium-sized graphs ($h = 6$)



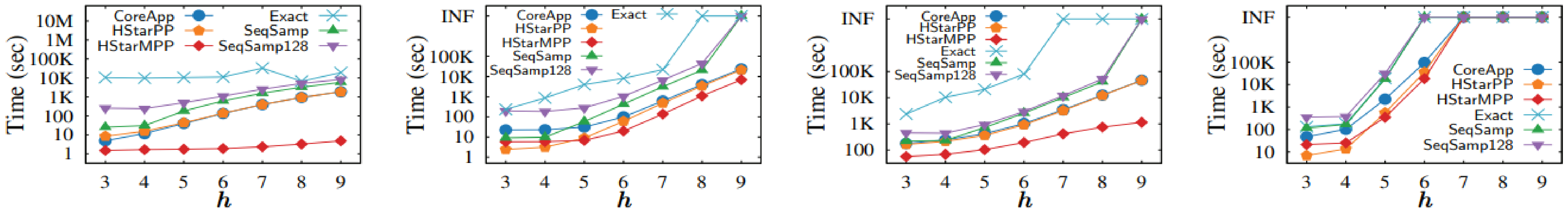
(c) Massive graphs ($h = 4$)



(d) Massive graphs graphs ($h = 6$)

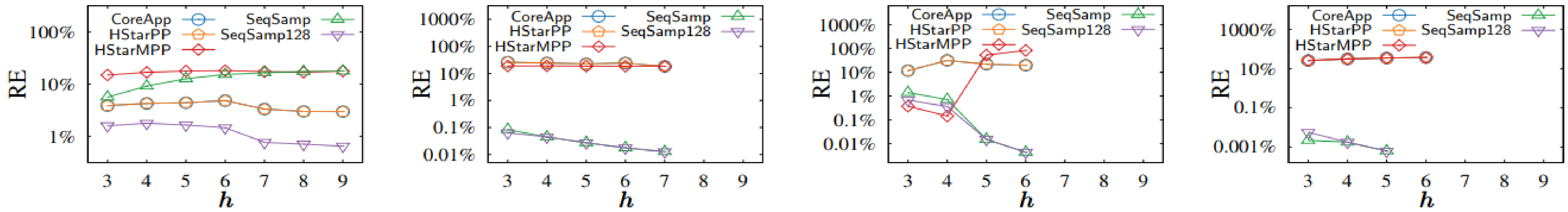
Fig. 8. Scalability of HStarCD

Results of the H-CLIQUE-DS-PROBLEM



(a) LDoor (b) Skitter (c) Orkut (d) LiveJournal

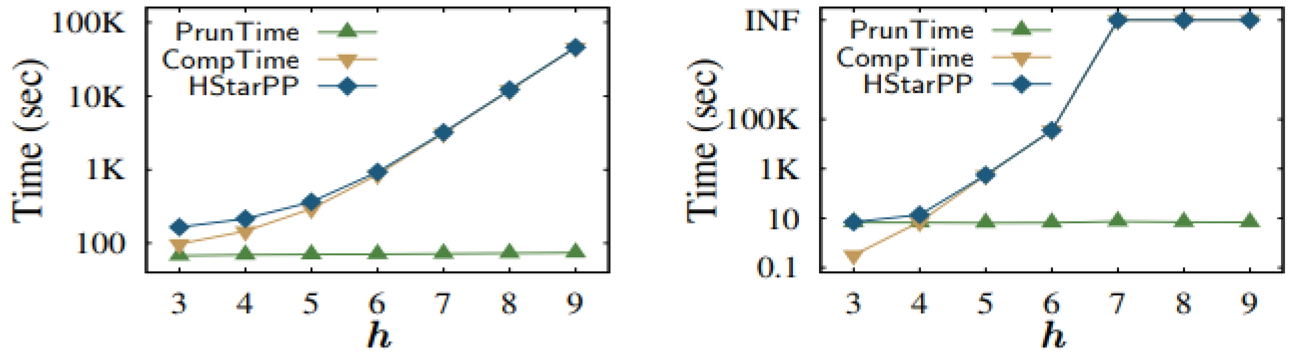
Fig. 10. Running time of different algorithms with varying h from 3 to 9



(a) LDoor (b) Skitter (c) Orkut (d) LiveJournal

Fig. 11. Relative error (RE) of different algorithms with varying h from 3 to 9 (Points are missing where Exact runs out of 24 hours.)

Results of the H-CLIQUE-DS-PROBLEM



(a) Orkut (b) LiveJournal

Fig. 12. The running time of HStarPP with varying h

TABLE II

THE POWER OF PRUNING TECHNIQUES USED IN HStarPP ($h = 6$, 1K=1,000, 1M=1,000,000, 1G=1,000,000,000)

Dataset	$n = \|V\|$			$m = \|E\|$			#Density=m/n			Δ	σ_6: 6-clique density		
	G	C^S_θ	$C^\Psi_{\delta\Psi}$	G	C^S_θ	$C^\Psi_{\delta\Psi}$	G	C^S_θ	$C^\Psi_{\delta\Psi}$	C^S_θ	G	C^S_θ	$C^\Psi_{\delta\Psi}$
Nasasrb	54.9K	52.1K	1.6K	1.3M	1.3M	28.4K	23.90	24.13	17.50	5.13%	12.36K	12.99K	11.14K
Pkustk	87.8K	41.3K	396	2.6M	1.4M	9.1K	29.21	32.75	23.05	53.01%	25.90K	29.71K	95.19K
Buzznet	101K	33.8K	275	2.8M	2.2M	21.5K	27.31	65.04	78.23	66.63%	63.68K	191K	4.17M
DBLP	317K	114	114	1.0M	6.4K	6.4K	3.31	56.50	56.50	99.96%	13.31K	23.39M	23.39M
Digg	771K	23.4K	153	5.9M	2.9M	9.5K	7.66	125.26	62.29	96.96%	20.01K	658K	10.25M
Skitter	1.7M	3.0K	180	11.1M	222K	11.9K	6.55	73.87	66.24	99.82%	5.76K	2.66M	9.53M
Orkut	3.0M	693K	132	106M	50.4M	7.5K	35.48	72.70	56.77	76.87%	15.75K	64.80K	7.57M
LiveJournal	4.8M	483	385	42.9M	108K	73.7K	8.84	224.41	191.31	99.99%	1.70M	16.86G	10.72G

The effects of graph colorings

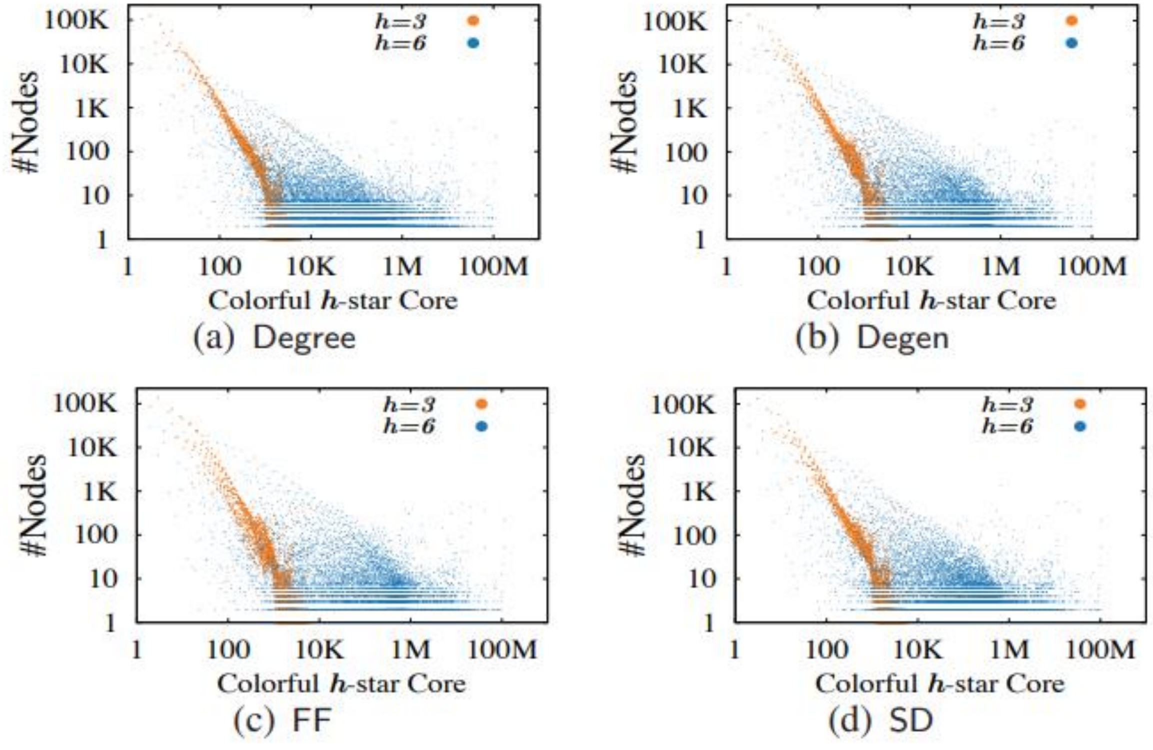


Fig. 13. Nodes' distributions of colorful h-star cores based on different graph colorings (Skitter)

TABLE III
PERFORMANCE OF DIFFERENT COLOR ALGORITHMS
(Skitter, $h = 6$, H IS THE $(\delta^{\mathcal{S}}, \mathcal{S})$-CORE)

	$n = \|V_H\|$	$m = \|E_H\|$	χ	$\sigma_6(H)$
Degree	212	15,503	71	10.27M
Degen	213	15,609	75	10.28M
FF	233	15,145	101	10.08M
SD	213	15,600	68	10.31M

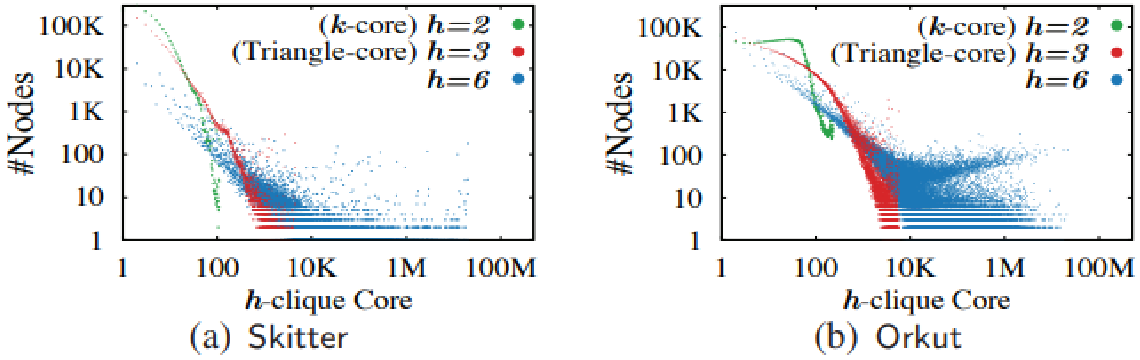
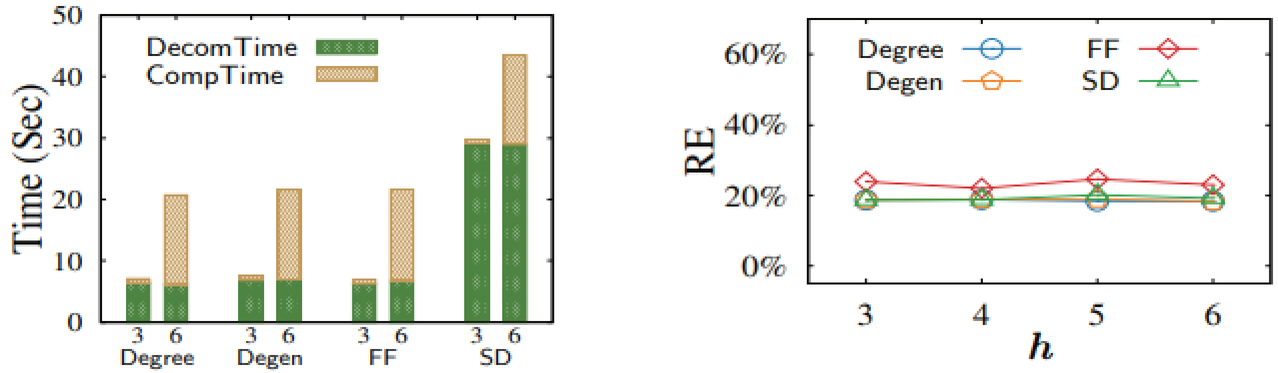


Fig. 14. Nodes' distributions of h-clique cores

The effects of graph colorings

TABLE III
PERFORMANCE OF DIFFERENT COLOR ALGORITHMS
(Skitter, $h = 6$, H IS THE $(\delta^{\mathcal{S}}, \mathcal{S})$-CORE)

	$n = \|V_H\|$	$m = \|E_H\|$	χ	$\sigma_6(H)$
Degree	212	15,503	71	10.27M
Degen	213	15,609	75	10.28M
FF	233	15,145	101	10.08M
SD	213	15,600	68	10.31M

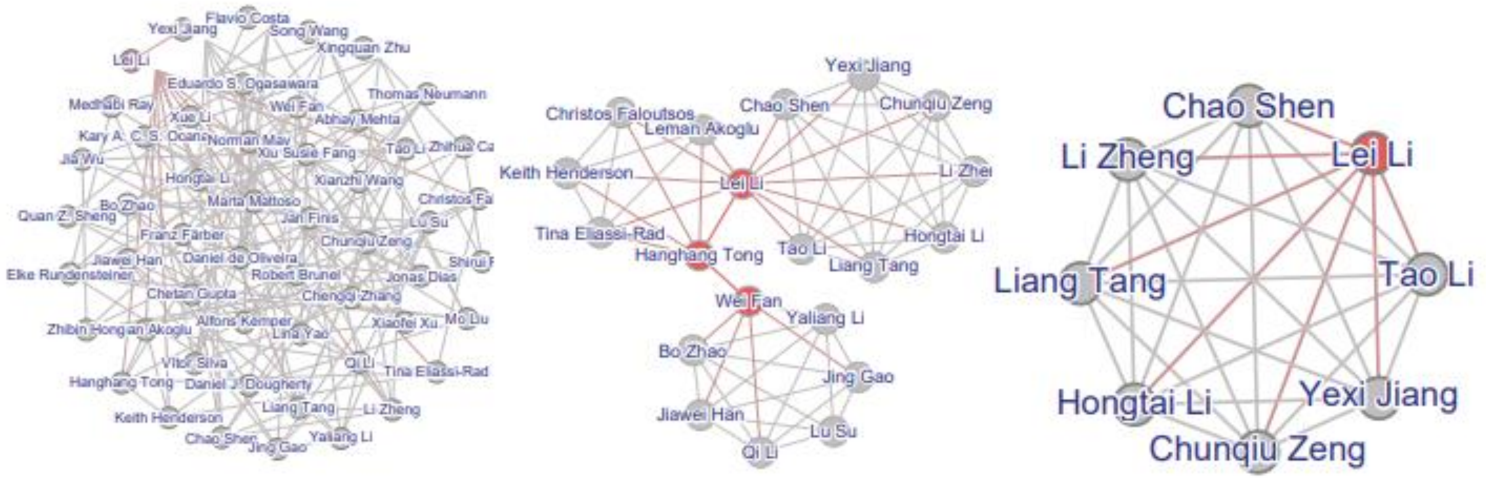


(a) Running Time ($h = 3, 6$) (b) Relative Error

Fig. 15. Performance of HStarMPP with various graph coloring tehcniques

a good approximation of h-clique core.

Case Studies



(a) k-core (b) $(k, \mathcal{S})$-cores (c) triangle densest subgraph

Fig. 16. The cohesive subgraphs found in DBLPs, based on k-core, $(k, \mathcal{S})$-cores and triangle densest subgraph ($\mathcal{S}$: colorful 3-star).

Conclusion

- In this paper, they propose a new colorful h-star core model to identify cohesive subgraphs in large real-world graphs.
- To efficiently compute the colorful h-star cores, they develop a novel DP based colorful h-star degree counting and updating algorithms.
- Based on colorful h-star core, they propose a graph reduction technique to speed up an approximate k-clique densest subgraph mining algorithm.
- They conduct extensive experiments on 11 large real-world graphs, and the results demonstrate the efficiency, scalability and effectiveness of the proposed solutions.

THANK YOU

Xiaojia Xu