Locally Densest Subgraph Discovery

Author: Lu Qin, Rong-Hua Li, Lijun Chang, Chengqi Zhang

KDD 2015

Xiaojia Xu

Problem

- **The studies of dense subgraph mining problem**

- **finding the densest subgraph (the subgraph with the highest density)[6][25]**

- **Identifying an optimal clique-like dense subgraph[37]**

[6] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *J. Algorithms*, 34(2), 2000.

[25] A. V. Goldberg. Finding a maximum density subgraph. Technical report, University of California at Berkeley, 1984.

[37] C. E. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. A. Tsiarli. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *Proc. of KDD'13*, 2013.

Applications

- the network science domain

[14] J. Chen and Y. Saad. Dense subgraph extraction with application to community detection. *TKDE*, 24(7), 2012.

[20] Y. Dourisboure, F. Geraci, and M. Pellegrini. Extraction and classification of dense communities in the web. In *WWW'07*, 2007.

[21] E. Fratkin, B. T. Naughton, D. L. Brutlag, and S. Batzoglou. Motifcut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics*, 22(14), 2006.

[31] B. Saha, A. Hoch, S. Khuller, L. Raschid, and X. Zhang. Dense subgraphs with restrictions and applications to gene annotation graphs. In *Proc. of RECOMB'10*, 2010.

- the biology domain

- The graph database domain

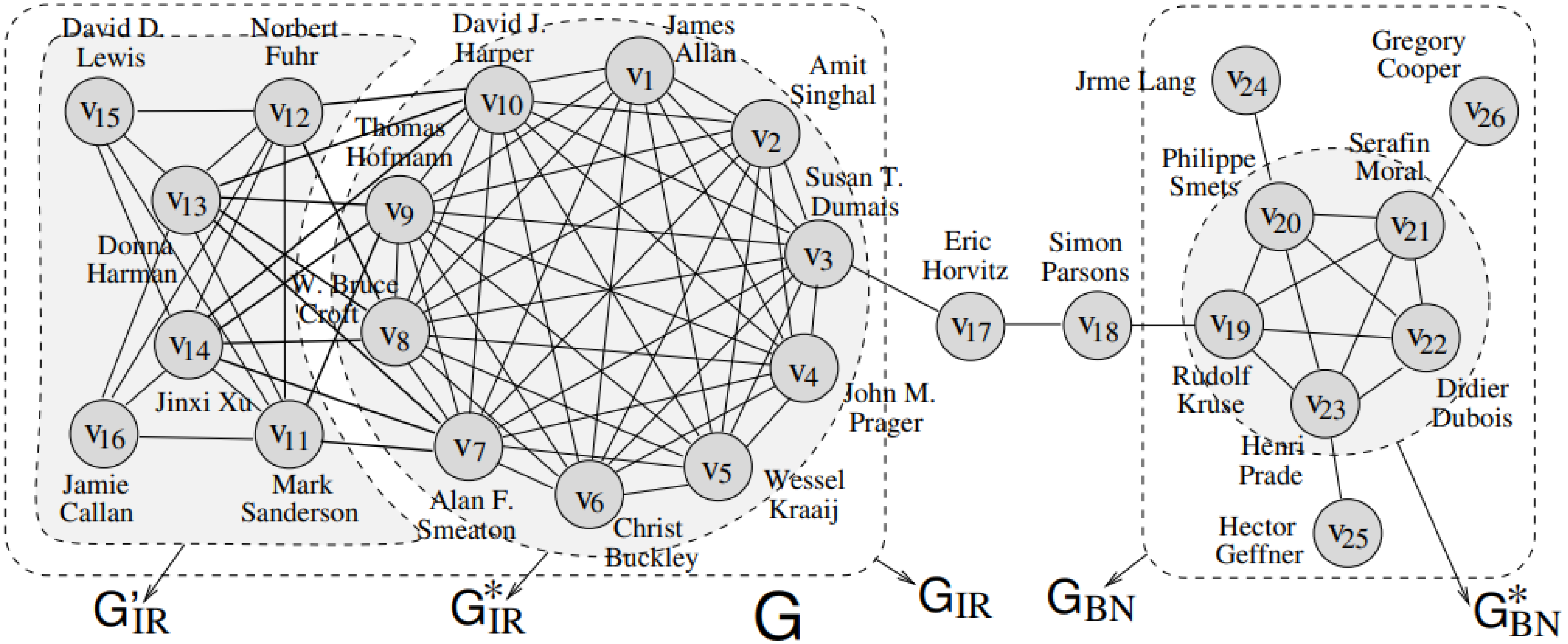
[17] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick. Reachability and distance queries via 2-hop labels. In *Proc. of SODA'02*, 2002.

[26] R. Jin, Y. Xiang, N. Ruan, and D. Fuhry. 3-hop: a high-compression indexing scheme for reachability query. In *SIGMOD'09*, 2009.

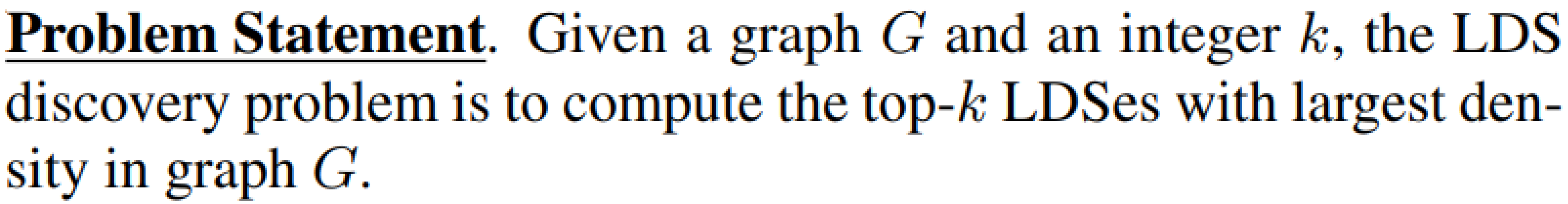
[23] D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *Proc. of VLDB'05*, 2005.

- the web mining domain

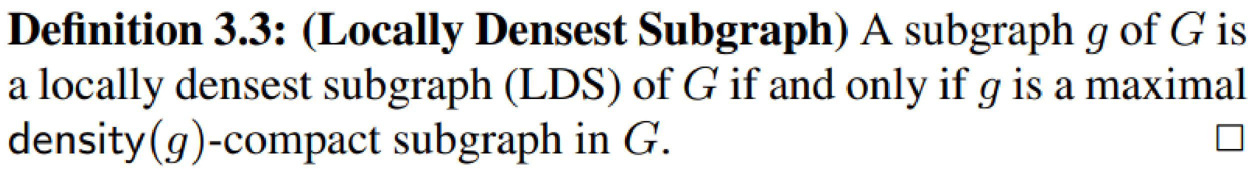
Background


David D. Lewis
Norbert Fuhr
David J. Harper
James Allan
Amit Singhal
Thomas Hofmann
Susan T. Dumais
Donna Harman
W. Bruce Croft
Jinxi Xu
Jamie Callan
Mark Sanderson
Alan F. Smeaton
Christ Buckley
Wessel Kraaij
John M. Prager
Eric Horvitz
Simon Parsons
Jrme Lang
Gregory Cooper
Philippe Smets
Serafin Moral
Rudolf Kruse
Henri Prade
Didier Dubois
Hector Geffner
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
v_{10}
v_{11}
v_{12}
v_{13}
v_{14}
v_{15}
v_{16}
v_{17}
v_{18}
v_{19}
v_{20}
v_{21}
v_{22}
v_{23}
v_{24}
v_{25}
v_{26}
G'_{IR}
G^*_{IR}
G
G_{IR}
G_{BN}
G^*_{BN}

Problem

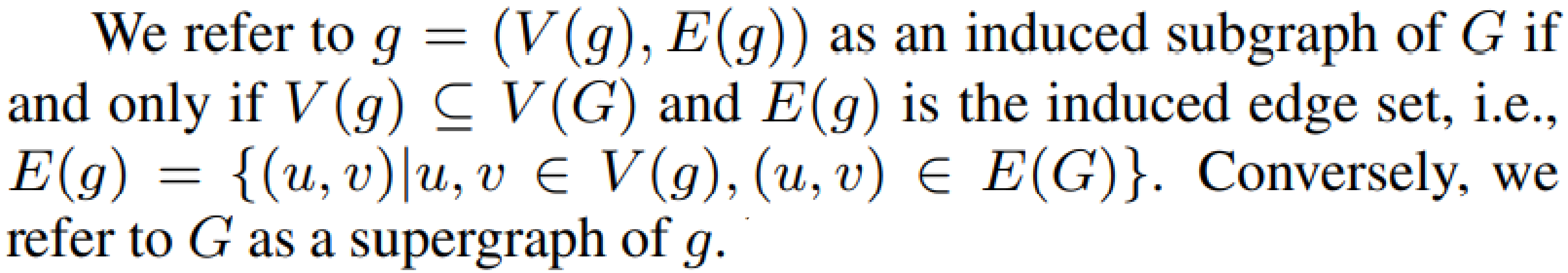


Problem Statement. Given a graph G and an integer k, the LDS discovery problem is to compute the top-k LDSes with largest density in graph G.



Definition 3.3: (Locally Densest Subgraph) A subgraph g of G is a locally densest subgraph (LDS) of G if and only if g is a maximal density(g)-compact subgraph in G. □

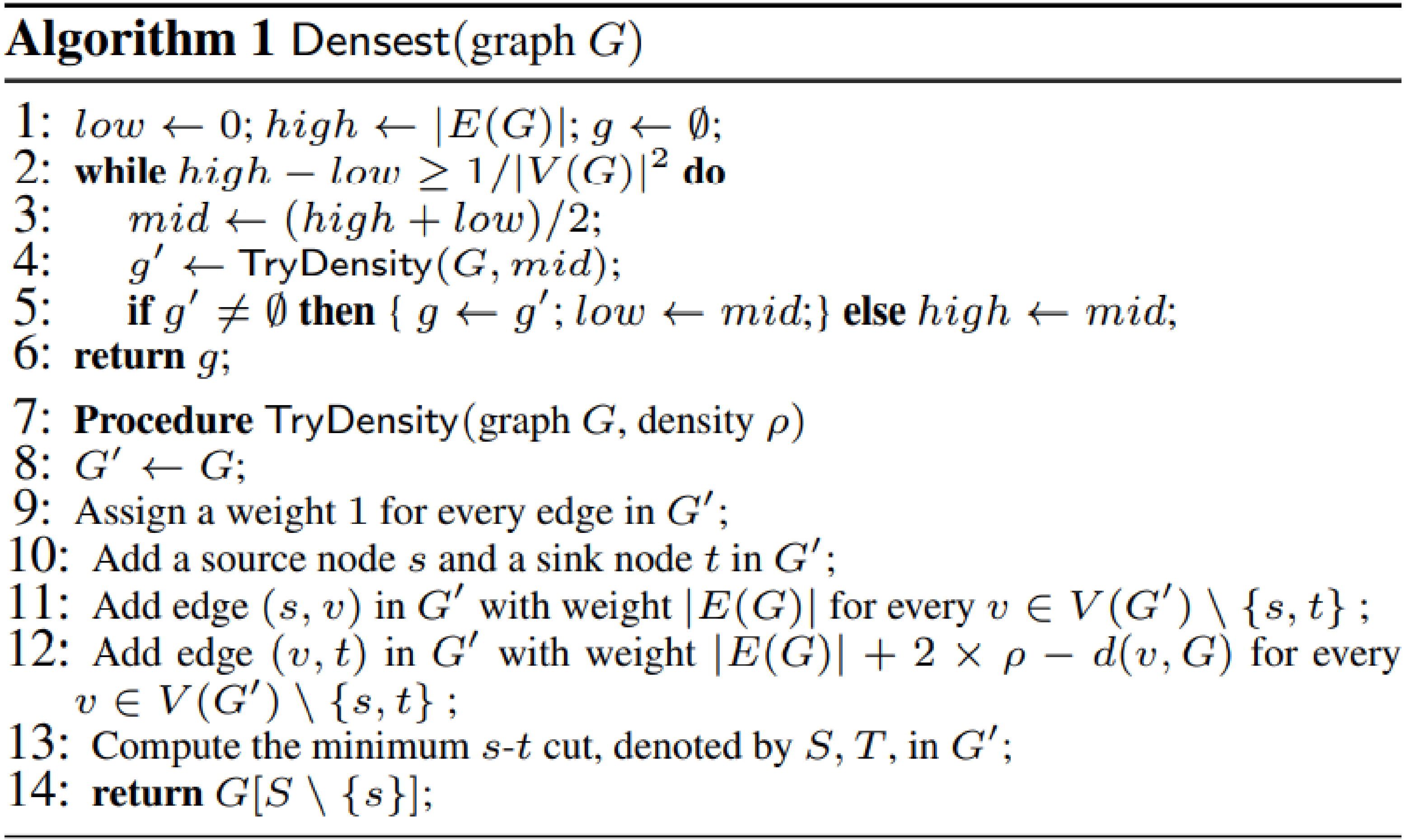
Fundamentals



We refer to $g = (V(g), E(g))$ as an induced subgraph of G if and only if $V(g) \subseteq V(G)$ and $E(g)$ is the induced edge set, i.e., $E(g) = \{(u, v)|u, v \in V(g), (u, v) \in E(G)\}$. Conversely, we refer to G as a supergraph of g.

$$\text{density}(G) = \frac{|E(G)|}{|V(G)|} \tag{1}$$

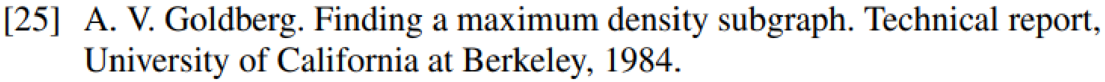
Fundamentals



Algorithm 1 Densest(graph G)

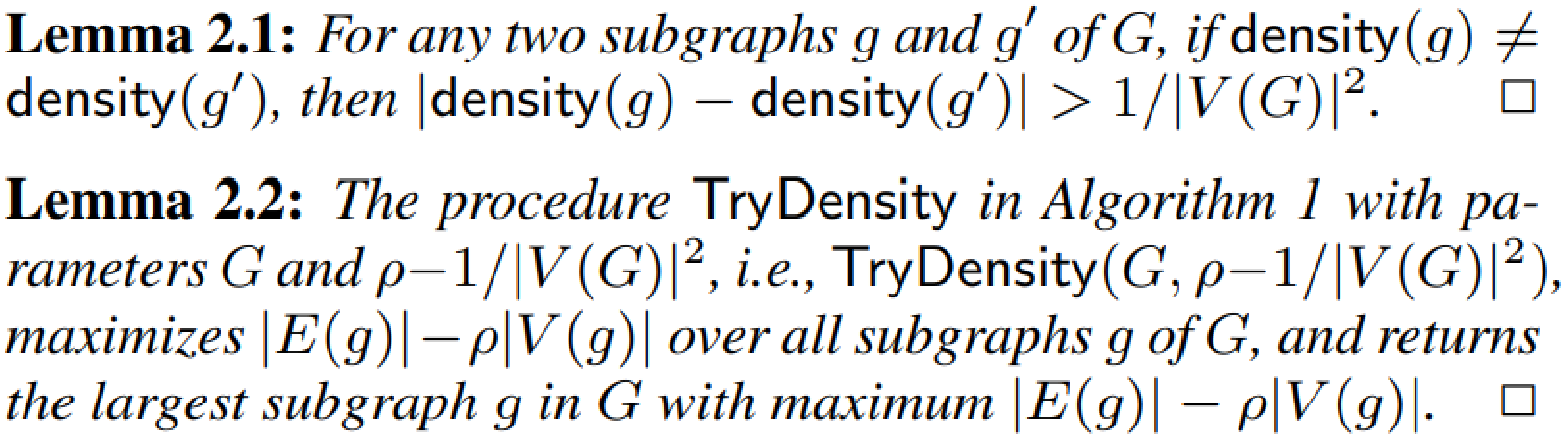
```
$low \leftarrow 0$; $high \leftarrow |E(G)|$; $g \leftarrow \emptyset$;
while $high - low \geq 1/|V(G)|^2$ do
    $mid \leftarrow (high + low)/2$;
    $g' \leftarrow$ TryDensity($G, mid$);
    if $g' \neq \emptyset$ then { $g \leftarrow g'$; $low \leftarrow mid$;} else $high \leftarrow mid$;
return $g$;

Procedure TryDensity(graph $G$, density $\rho$)
$G' \leftarrow G$;
Assign a weight 1 for every edge in $G'$;
Add a source node $s$ and a sink node $t$ in $G'$;
Add edge $(s, v)$ in $G'$ with weight $|E(G)|$ for every $v \in V(G') \setminus \{s, t\}$ ;
Add edge $(v, t)$ in $G'$ with weight $|E(G)| + 2 \times \rho - d(v, G)$ for every $v \in V(G') \setminus \{s, t\}$ ;
Compute the minimum $s$-$t$ cut, denoted by $S, T$, in $G'$;
return $G[S \setminus \{s\}]$;
```



[25] A. V. Goldberg. Finding a maximum density subgraph. Technical report, University of California at Berkeley, 1984.

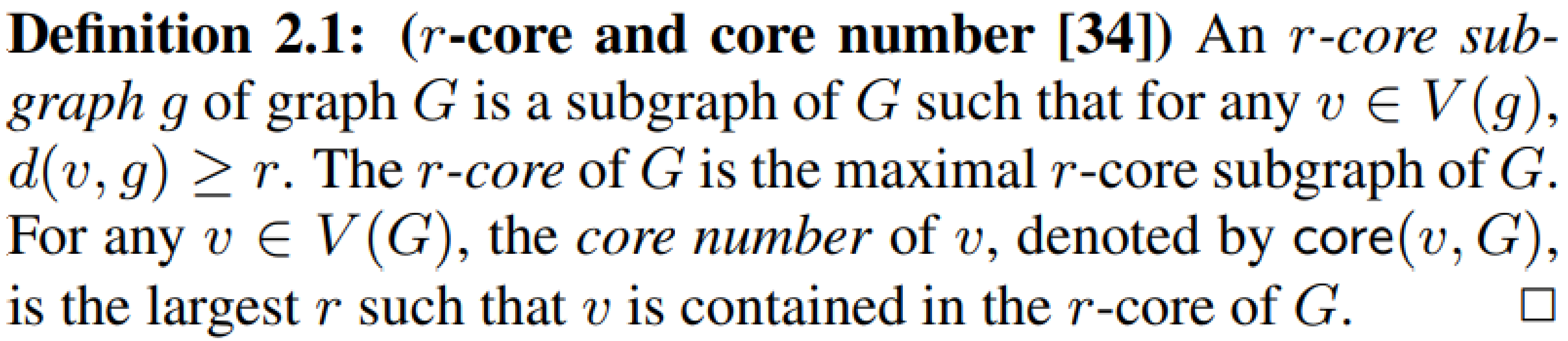
Fundamentals



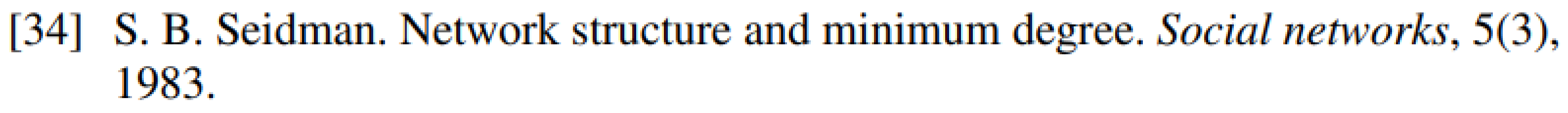
Lemma 2.1: *For any two subgraphs g and g' of G, if* $\mathsf{density}(g) \neq \mathsf{density}(g')$, *then* $|\mathsf{density}(g) - \mathsf{density}(g')| > 1/|V(G)|^2$. □

Lemma 2.2: *The procedure* TryDensity *in Algorithm 1 with parameters G and $\rho{-}1/|V(G)|^2$, i.e.,* $\mathsf{TryDensity}(G, \rho{-}1/|V(G)|^2)$, *maximizes $|E(g)|-\rho|V(g)|$ over all subgraphs g of G, and returns the largest subgraph g in G with maximum $|E(g)| - \rho|V(g)|$.* □

Fundamentals



Definition 2.1: (**r-core and core number [34]**) An *r-core subgraph* g of graph G is a subgraph of G such that for any $v \in V(g)$, $d(v, g) \geq r$. The *r-core* of G is the maximal r-core subgraph of G. For any $v \in V(G)$, the *core number* of v, denoted by $\mathsf{core}(v, G)$, is the largest r such that v is contained in the r-core of G. $\square$



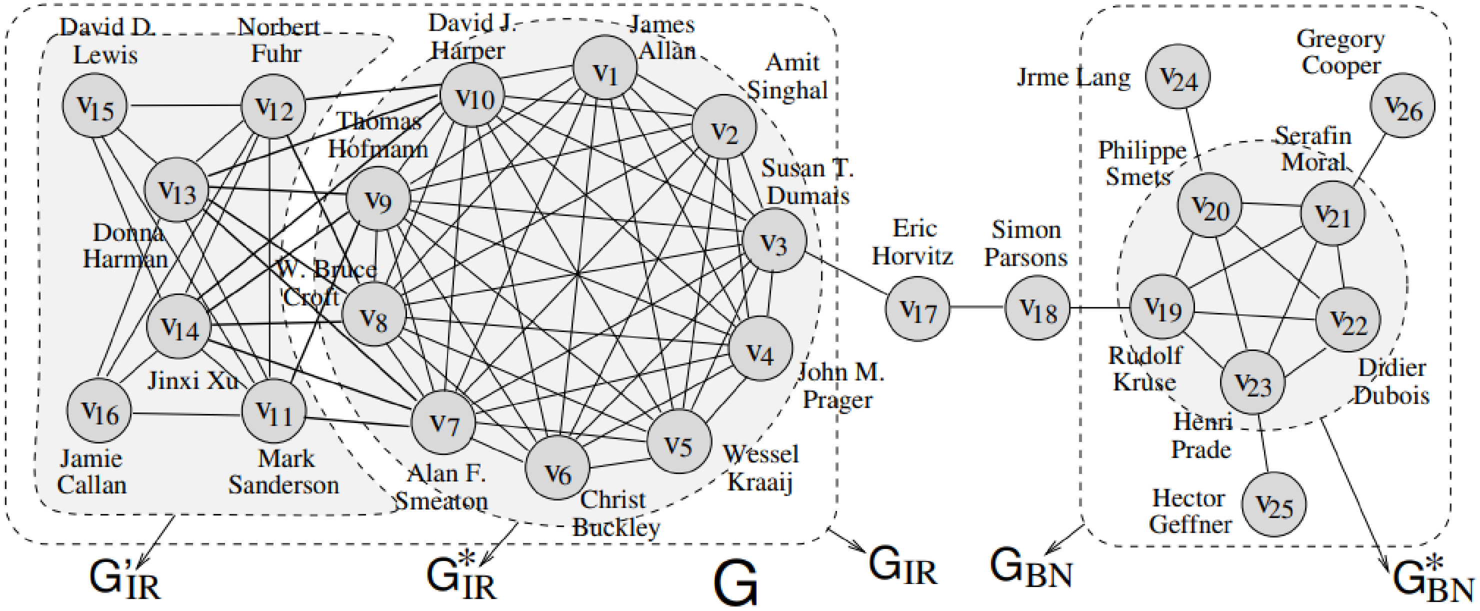
[34] S. B. Seidman. Network structure and minimum degree. *Social networks*, 5(3), 1983.

Greed is Not Good

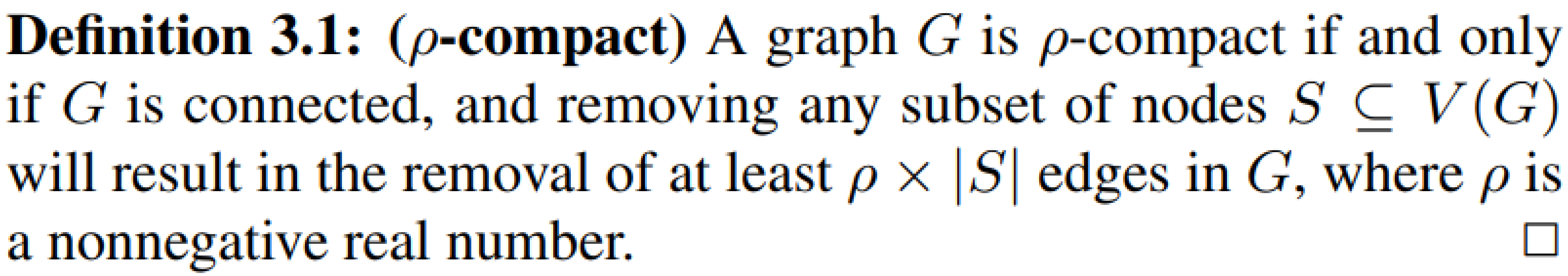
- (1) The top-k results may not fully reflect the top-k densest regions of a graph.
- (2) A subgraph returned by the greedy approach can be partial and subsumed by a better subgraph.
- (3) Such a greedy approach does not provide a formal definition of a result.

Greed is Not Good

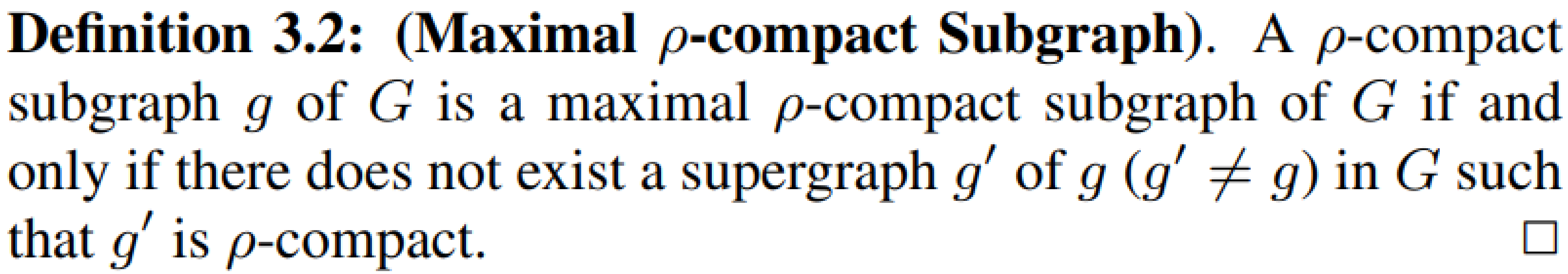
- (1) The top-k results may not fully reflect the top-k densest regions of a graph.
- (2) A subgraph returned by the greedy approach can be partial and subsumed by a better subgraph.
- (3) Such a greedy approach does not provide a formal definition of a result.



Dense or Compact?



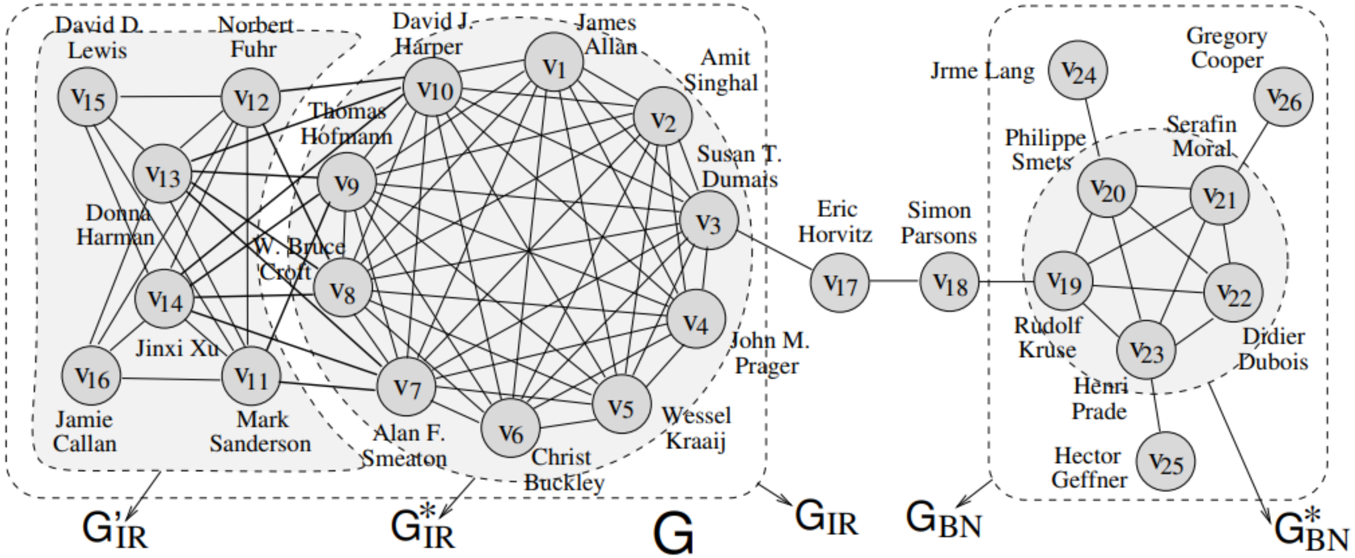
Definition 3.1: (ρ-compact) A graph G is ρ-compact if and only if G is connected, and removing any subset of nodes $S \subseteq V(G)$ will result in the removal of at least $\rho \times |S|$ edges in G, where ρ is a nonnegative real number. $\square$



Definition 3.2: (Maximal ρ-compact Subgraph). A ρ-compact subgraph g of G is a maximal ρ-compact subgraph of G if and only if there does not exist a supergraph g' of g ($g' \neq g$) in G such that g' is ρ-compact. $\square$

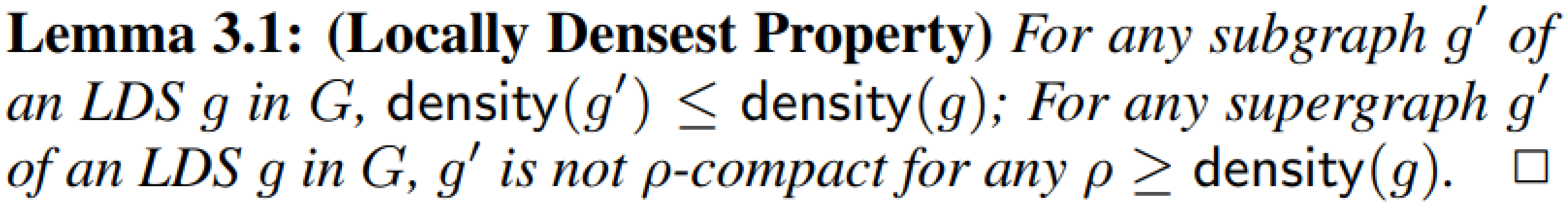
Algorithms

Definition 3.3: (Locally Densest Subgraph) A subgraph g of G is a locally densest subgraph (LDS) of G if and only if g is a maximal density(g)-compact subgraph in G. □

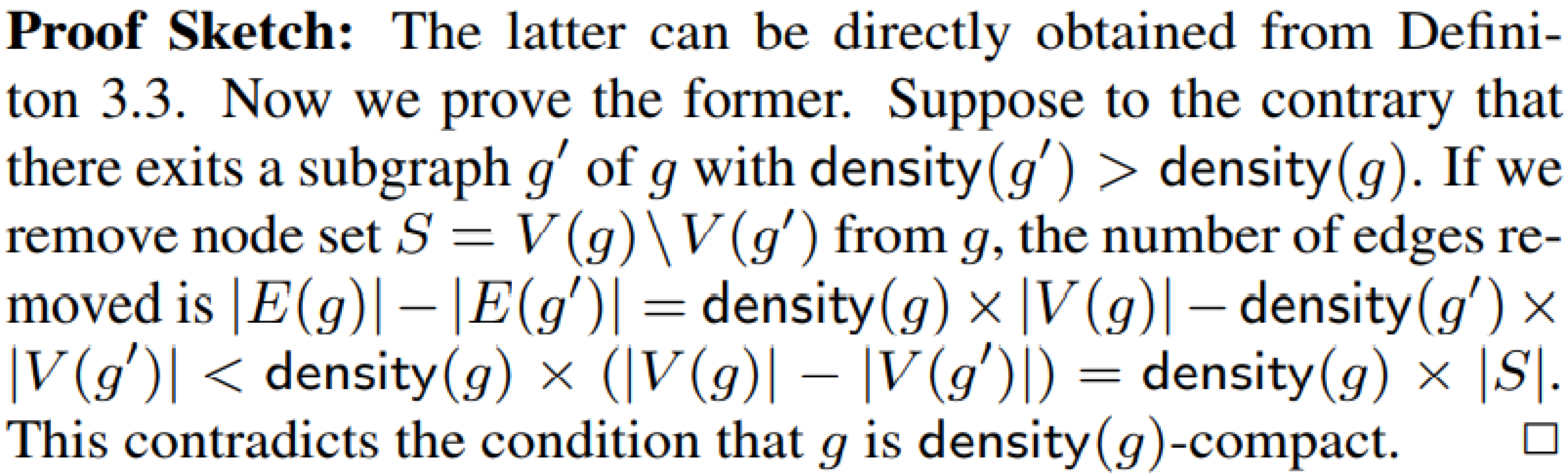


- G_{IR} $density: \frac{35}{8}$ $compact: 4$ $not\ LDS$
- G'_{IR} $density: \frac{13}{6}$ $compact: \frac{13}{6}$ $not\ LDS$
- G^*_{IR} $density: 4.5$ $compact: 4.5$ $is\ LDS$
- G^*_{BN} $density: 2$ $compact: 2$ $is\ LDS$

Algorithms

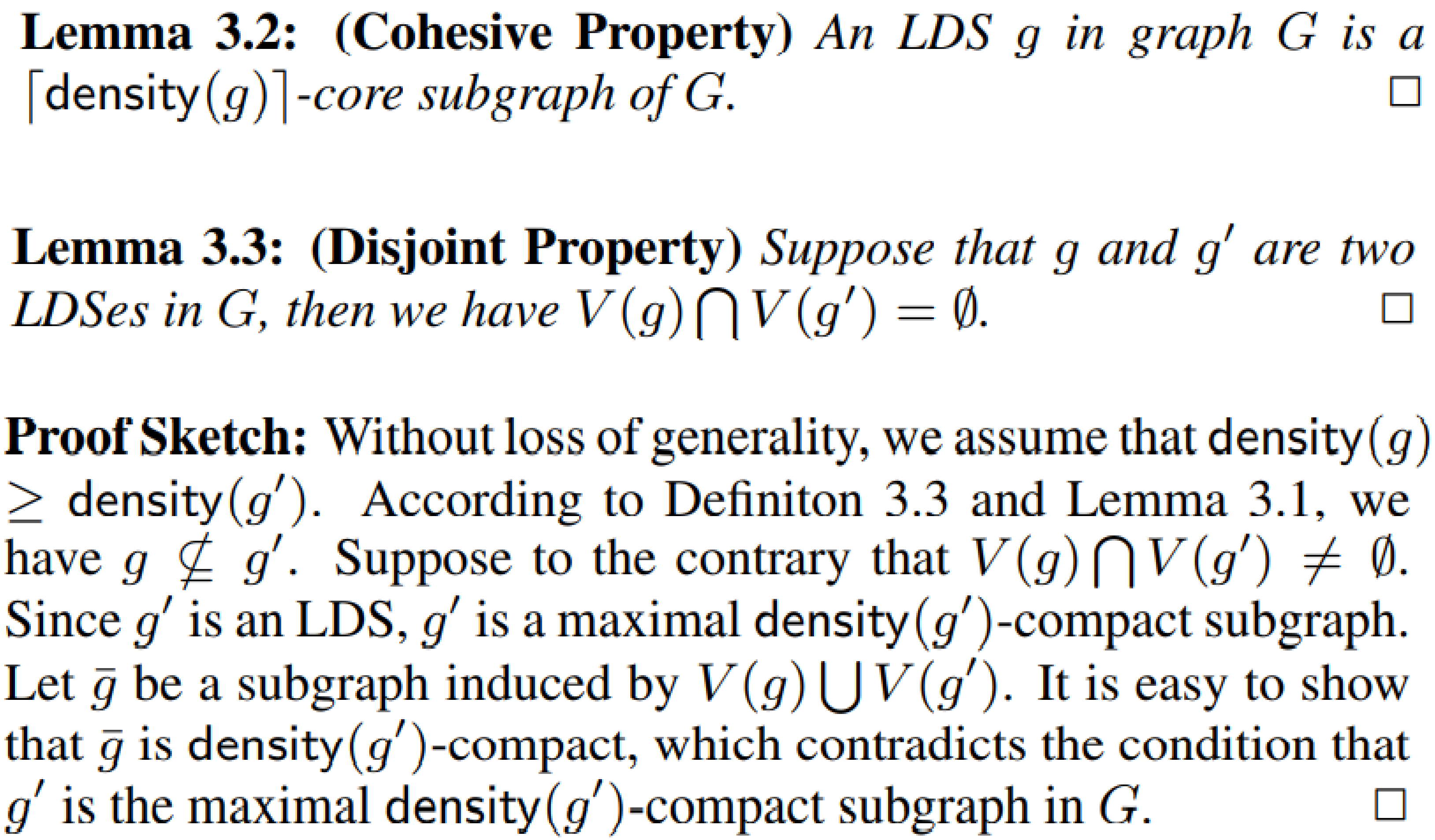


Lemma 3.1: (Locally Densest Property) *For any subgraph g' of an LDS g in G,* $\text{density}(g') \leq \text{density}(g)$*; For any supergraph g' of an LDS g in G, g' is not ρ-compact for any $\rho \geq$* $\text{density}(g)$. $\square$



Proof Sketch: The latter can be directly obtained from Definiton 3.3. Now we prove the former. Suppose to the contrary that there exits a subgraph g' of g with $\text{density}(g') > \text{density}(g)$. If we remove node set $S = V(g) \setminus V(g')$ from g, the number of edges removed is $|E(g)| - |E(g')| = \text{density}(g) \times |V(g)| - \text{density}(g') \times |V(g')| < \text{density}(g) \times (|V(g)| - |V(g')|) = \text{density}(g) \times |S|$. This contradicts the condition that g is $\text{density}(g)$-compact. $\square$

Algorithms



Lemma 3.2: (Cohesive Property) *An LDS g in graph G is a $\lceil \text{density}(g) \rceil$-core subgraph of G.* $\square$

Lemma 3.3: (Disjoint Property) *Suppose that g and g' are two LDSes in G, then we have $V(g) \bigcap V(g') = \emptyset$.* $\square$

Proof Sketch: Without loss of generality, we assume that $\text{density}(g) \geq \text{density}(g')$. According to Definiton 3.3 and Lemma 3.1, we have $g \not\subseteq g'$. Suppose to the contrary that $V(g) \bigcap V(g') \neq \emptyset$. Since g' is an LDS, g' is a maximal $\text{density}(g')$-compact subgraph. Let $\bar{g}$ be a subgraph induced by $V(g) \bigcup V(g')$. It is easy to show that $\bar{g}$ is $\text{density}(g')$-compact, which contradicts the condition that g' is the maximal $\text{density}(g')$-compact subgraph in G. $\square$

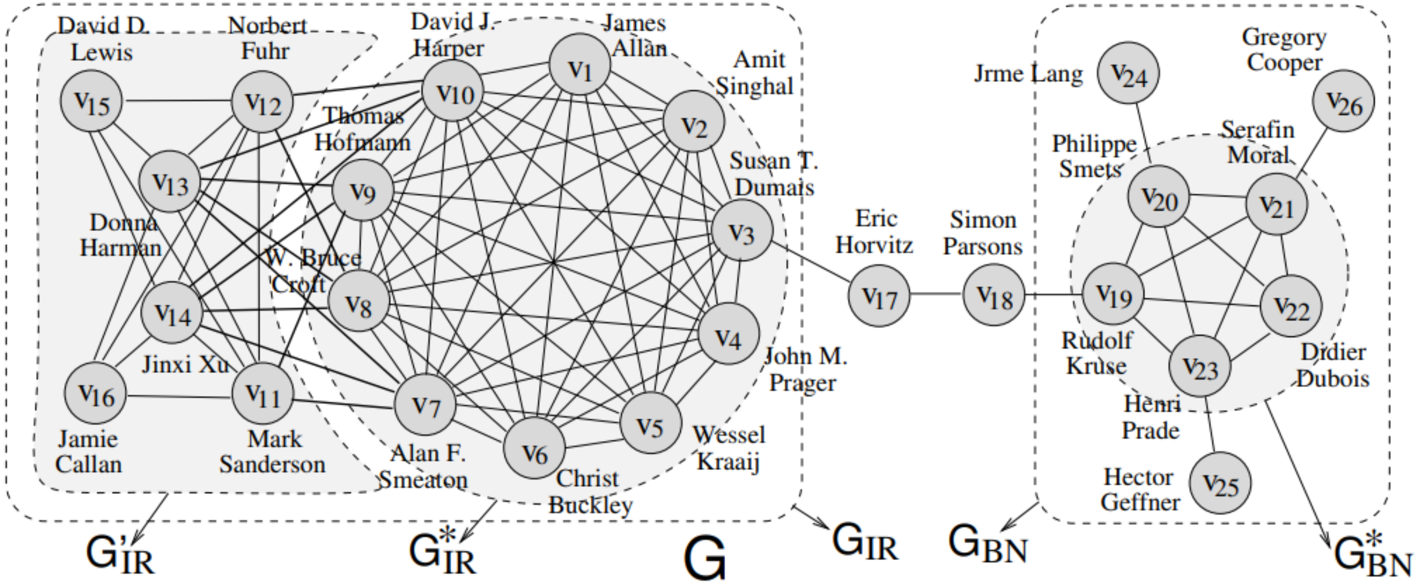
A Polynomial Algorithm

Lemma 4.1: *Any densest subgraph component of graph G is an LDS in G.* $\square$

Lemma 4.2: *Let g be an LDS of G , then any LDS g' $(g' \neq g)$ in G is still an LDS in G', where G' is the residual graph of G after removing g.* $\square$

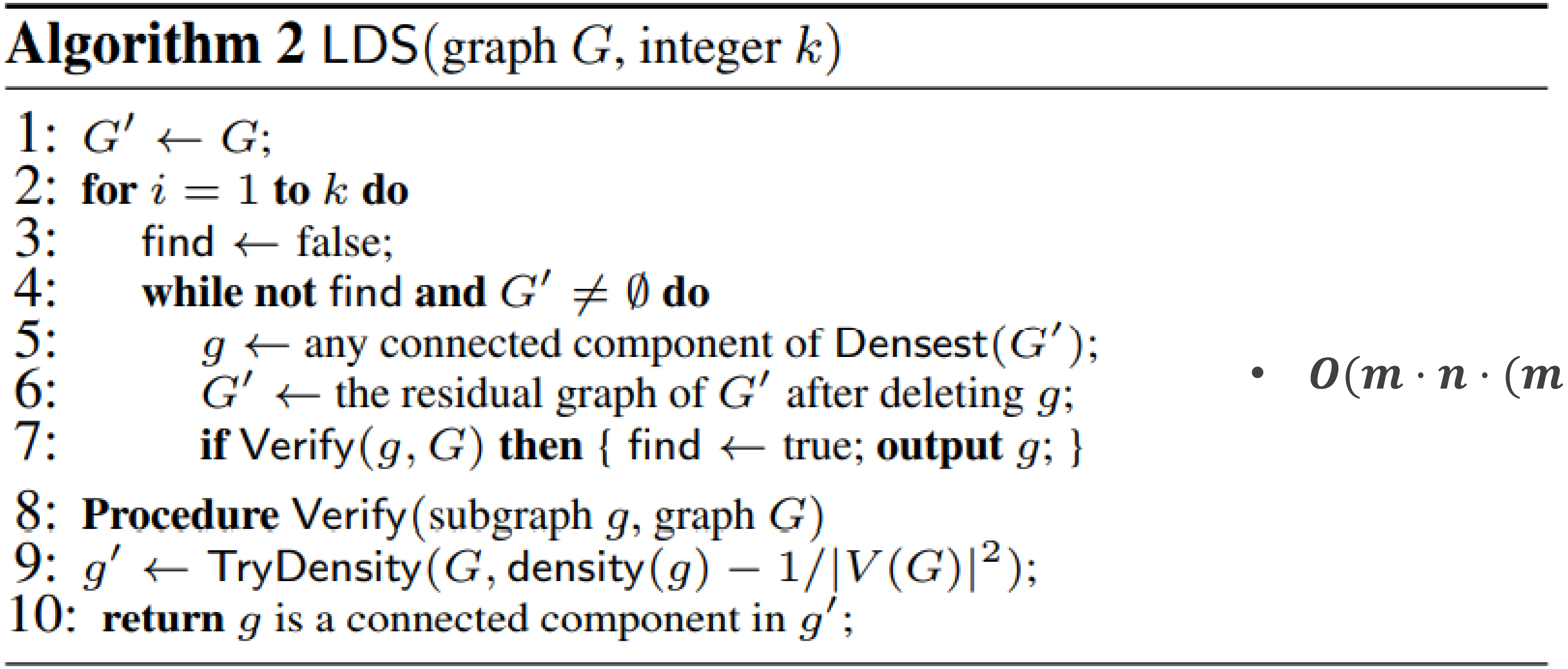
Lemma 3.3: **(Disjoint Property)** *Suppose that g and g' are two LDSes in G, then we have $V(g) \bigcap V(g') = \emptyset$.* $\square$

A Polynomial Algorithm



Lemma 4.3: *If* G *contains maximal* ρ*-compact subgraphs, then the result returned by the procedure* TryDensity$(G, \rho - 1/|V(G)|^2)$ *is the set of all maximal* ρ*-compact subgraphs in* G. □

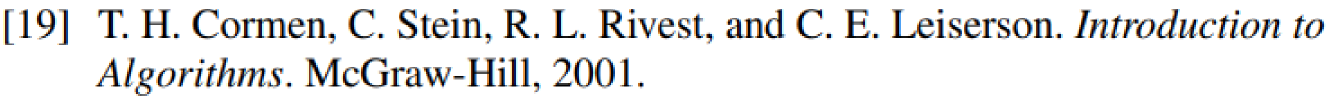
A Polynomial Algorithm



Algorithm 2 LDS(graph G, integer k)

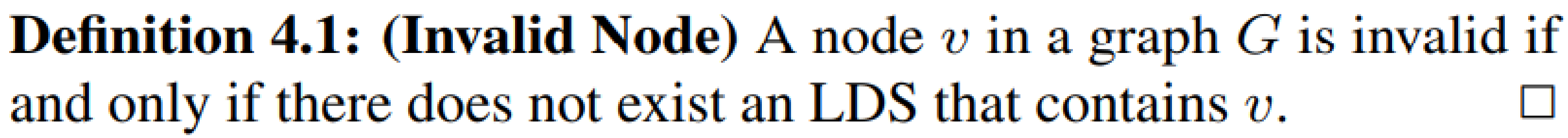
```
$G' \leftarrow G$;
for $i = 1$ to $k$ do
    find $\leftarrow$ false;
    while not find and $G' \neq \emptyset$ do
        $g \leftarrow$ any connected component of Densest($G'$);
        $G' \leftarrow$ the residual graph of $G'$ after deleting $g$;
        if Verify($g, G$) then { find $\leftarrow$ true; output $g$; }
Procedure Verify(subgraph $g$, graph $G$)
$g' \leftarrow$ TryDensity($G$, density($g$) $- 1/|V(G)|^2$);
return $g$ is a connected component in $g'$;
```

- $O(m \cdot n \cdot (m+n) \cdot log^2 n)$

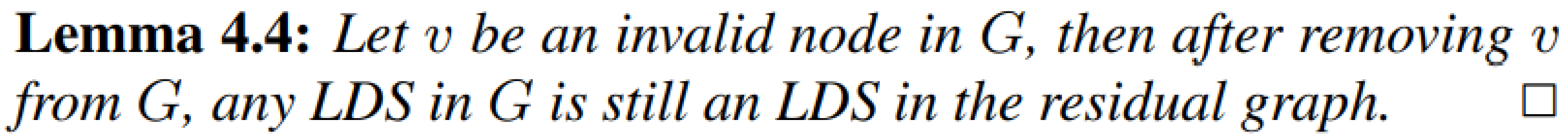


[19] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill, 2001.

The Optimized Algorithm

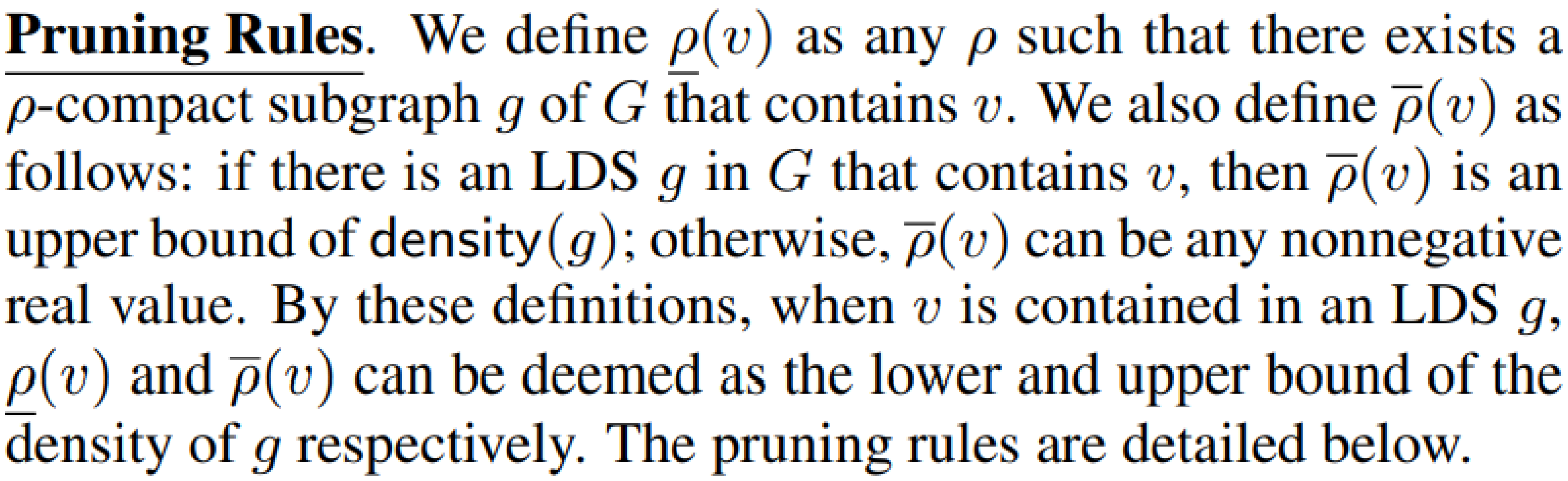


Definition 4.1: (Invalid Node) A node v in a graph G is invalid if and only if there does not exist an LDS that contains v. ☐

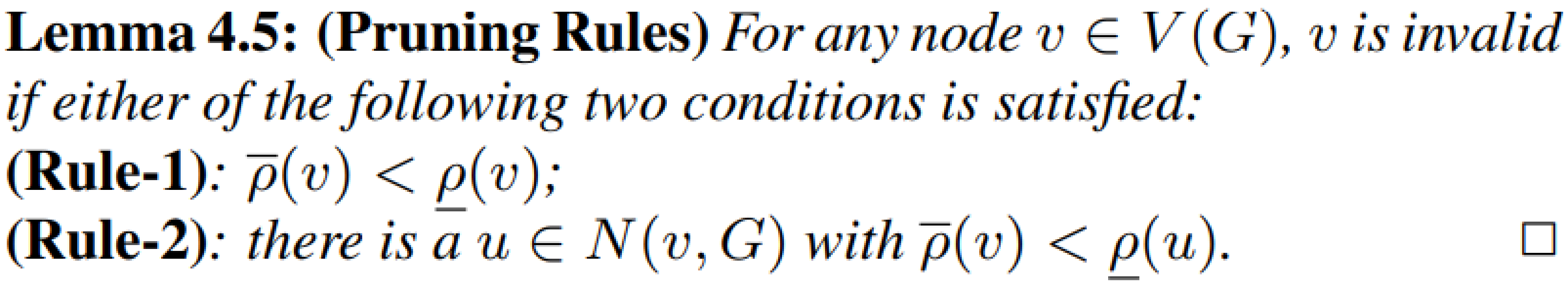


Lemma 4.4: *Let v be an invalid node in G, then after removing v from G, any LDS in G is still an LDS in the residual graph.* ☐

The Optimized Algorithm

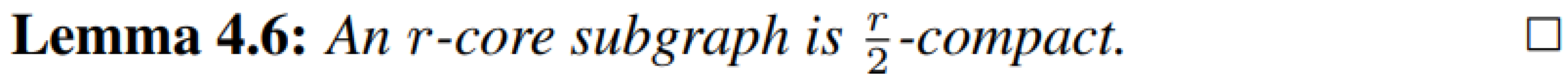


Pruning Rules. We define $\underline{\rho}(v)$ as any ρ such that there exists a ρ-compact subgraph g of G that contains v. We also define $\overline{\rho}(v)$ as follows: if there is an LDS g in G that contains v, then $\overline{\rho}(v)$ is an upper bound of density(g); otherwise, $\overline{\rho}(v)$ can be any nonnegative real value. By these definitions, when v is contained in an LDS g, $\underline{\rho}(v)$ and $\overline{\rho}(v)$ can be deemed as the lower and upper bound of the density of g respectively. The pruning rules are detailed below.

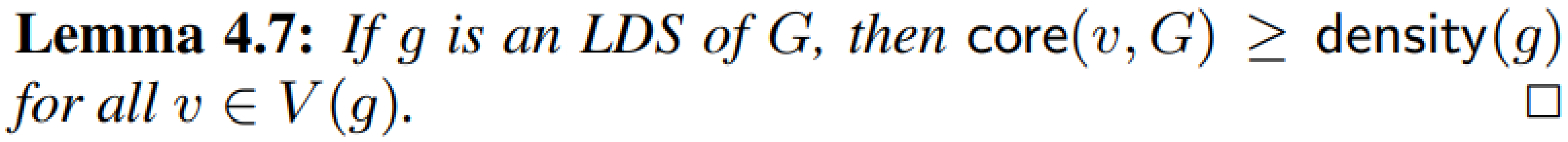


Lemma 4.5: (Pruning Rules) *For any node $v \in V(G)$, v is invalid if either of the following two conditions is satisfied:*
(Rule-1): $\overline{\rho}(v) < \underline{\rho}(v)$;
(Rule-2): *there is a* $u \in N(v, G)$ *with* $\overline{\rho}(v) < \underline{\rho}(u)$. □

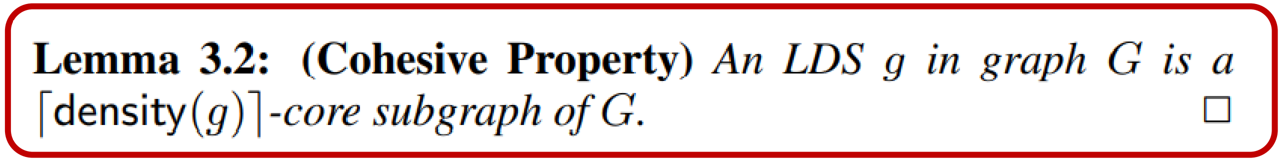
The Optimized Algorithm



Lemma 4.6: *An r-core subgraph is $\frac{r}{2}$-compact.* □

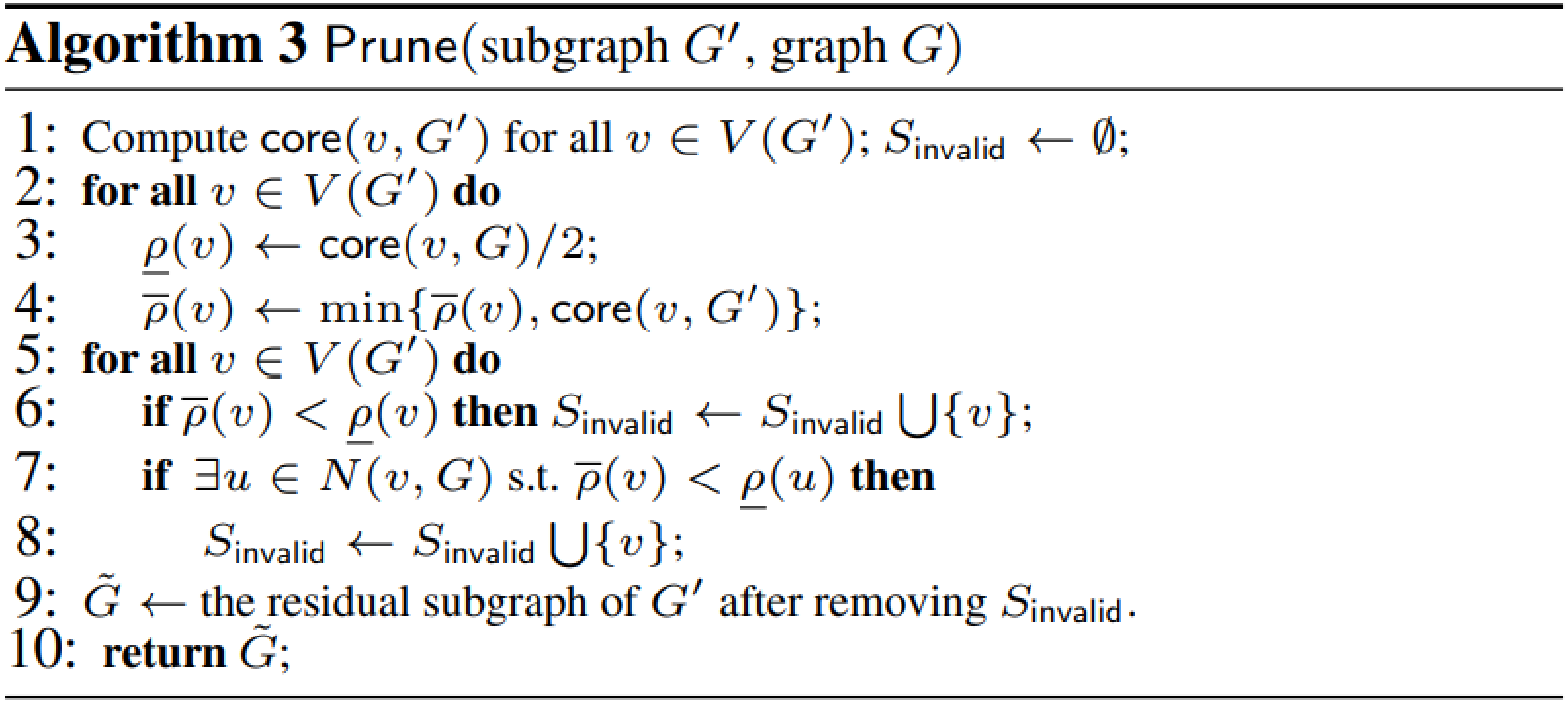


Lemma 4.7: *If g is an LDS of G, then* $\mathsf{core}(v, G) \geq \mathsf{density}(g)$ *for all $v \in V(g)$.* □



Lemma 3.2: (Cohesive Property) *An LDS g in graph G is a $\lceil \mathsf{density}(g) \rceil$-core subgraph of G.* □

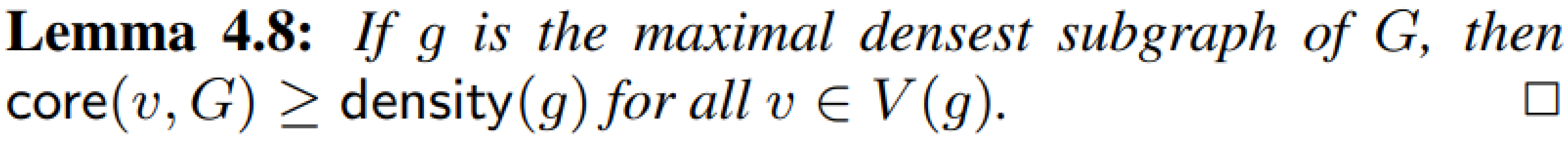
The Pruning Algorithm



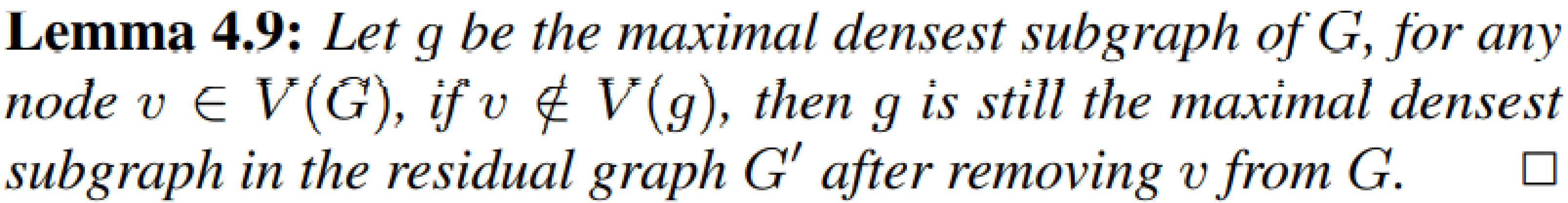
Algorithm 3 Prune(subgraph G', graph G)

```
Compute core$(v, G')$ for all $v \in V(G')$; $S_{\text{invalid}} \leftarrow \emptyset$;
for all $v \in V(G')$ do
    $\underline{\rho}(v) \leftarrow \text{core}(v, G)/2$;
    $\overline{\rho}(v) \leftarrow \min\{\overline{\rho}(v), \text{core}(v, G')\}$;
for all $v \in V(G')$ do
    if $\overline{\rho}(v) < \underline{\rho}(v)$ then $S_{\text{invalid}} \leftarrow S_{\text{invalid}} \bigcup \{v\}$;
    if $\exists u \in N(v, G)$ s.t. $\overline{\rho}(v) < \underline{\rho}(u)$ then
        $S_{\text{invalid}} \leftarrow S_{\text{invalid}} \bigcup \{v\}$;
$\tilde{G} \leftarrow$ the residual subgraph of $G'$ after removing $S_{\text{invalid}}$.
return $\tilde{G}$;
```

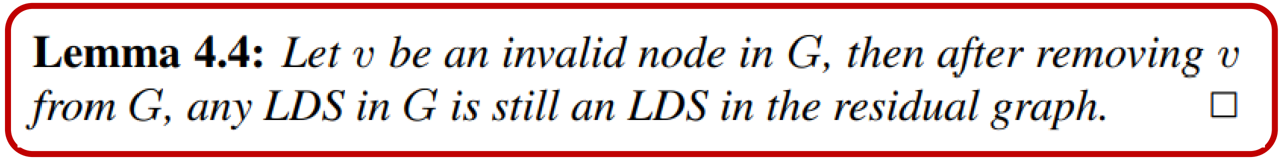
The Optimized Algorithm



Lemma 4.8: *If g is the maximal densest subgraph of G, then* $\mathsf{core}(v, G) \geq \mathsf{density}(g)$ *for all $v \in V(g)$.* □

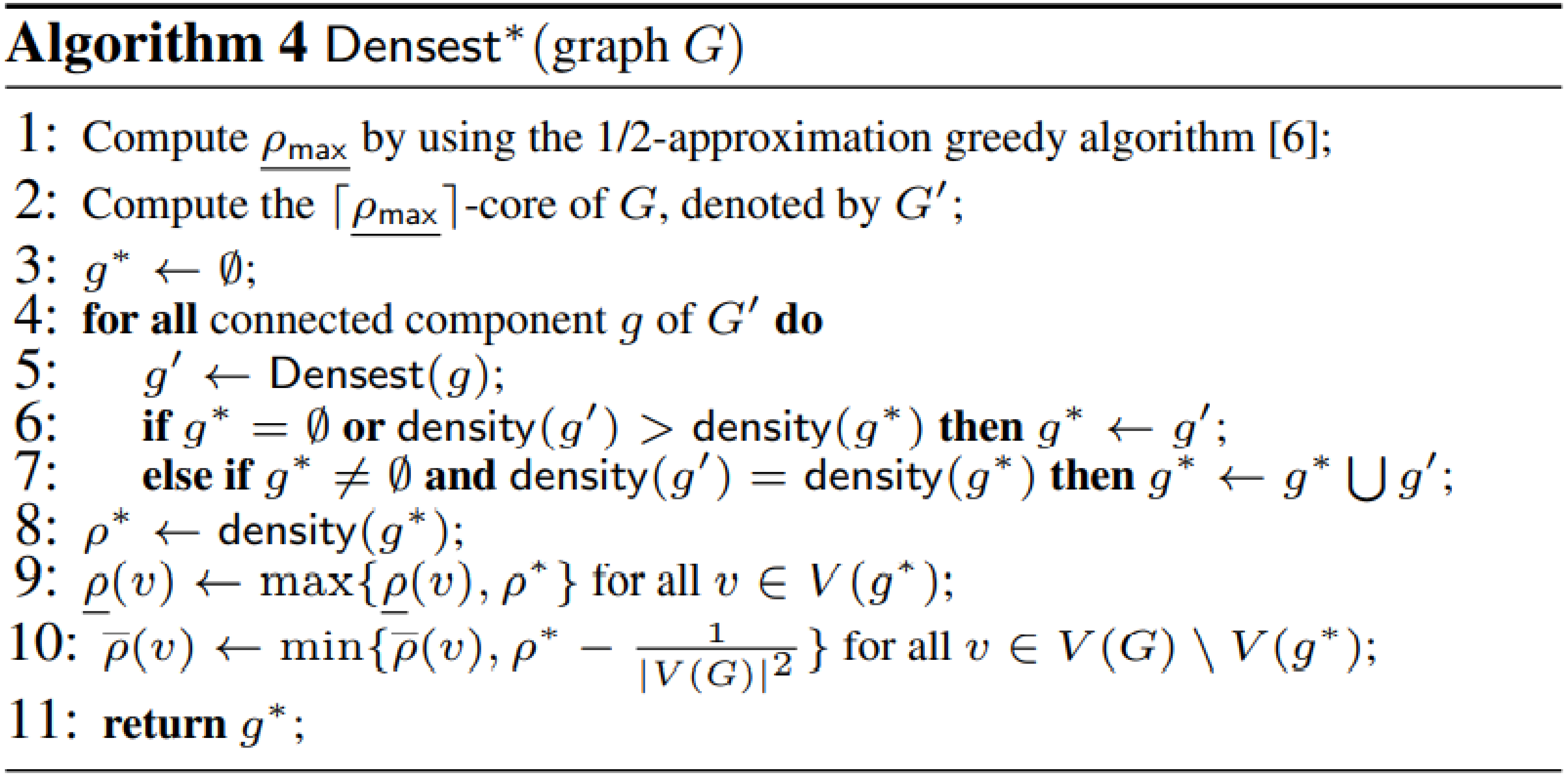


Lemma 4.9: *Let g be the maximal densest subgraph of G, for any node $v \in V(G)$, if $v \notin V(g)$, then g is still the maximal densest subgraph in the residual graph G' after removing v from G.* □



Lemma 4.4: *Let v be an invalid node in G, then after removing v from G, any LDS in G is still an LDS in the residual graph.* □

The Densest Algorithm

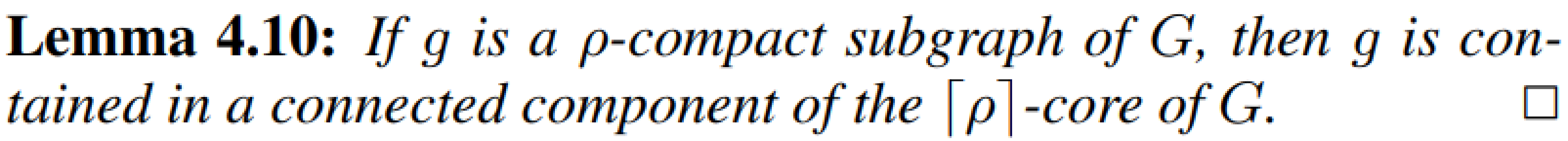


Algorithm 4 Densest* (graph G)

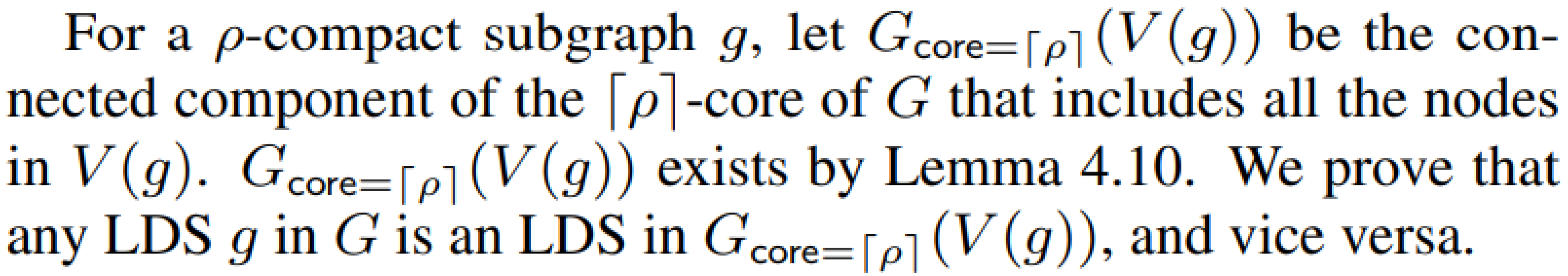
```
Compute $\underline{\rho_{\max}}$ by using the 1/2-approximation greedy algorithm [6];
Compute the $\lceil \underline{\rho_{\max}} \rceil$-core of $G$, denoted by $G'$;
$g^* \leftarrow \emptyset$;
for all connected component $g$ of $G'$ do
    $g' \leftarrow \mathsf{Densest}(g)$;
    if $g^* = \emptyset$ or $\mathsf{density}(g') > \mathsf{density}(g^*)$ then $g^* \leftarrow g'$;
    else if $g^* \neq \emptyset$ and $\mathsf{density}(g') = \mathsf{density}(g^*)$ then $g^* \leftarrow g^* \bigcup g'$;
$\rho^* \leftarrow \mathsf{density}(g^*)$;
$\underline{\rho}(v) \leftarrow \max\{\underline{\rho}(v), \rho^*\}$ for all $v \in V(g^*)$;
$\overline{\rho}(v) \leftarrow \min\{\overline{\rho}(v), \rho^* - \frac{1}{|V(G)|^2}\}$ for all $v \in V(G) \setminus V(g^*)$;
return $g^*$;
```

[6] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *J. Algorithms*, 34(2), 2000.

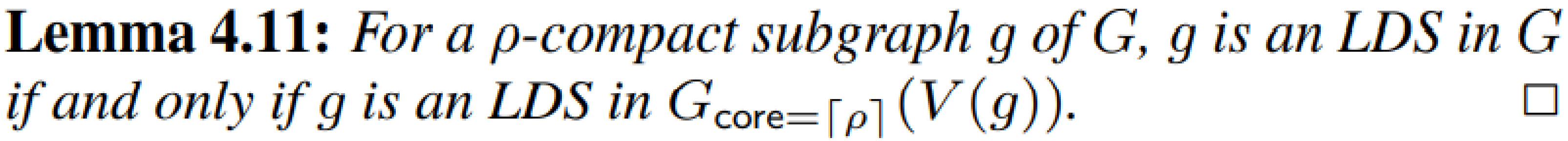
The Optimized Algorithm



Lemma 4.10: *If g is a ρ-compact subgraph of G, then g is contained in a connected component of the $\lceil \rho \rceil$-core of G.* $\square$

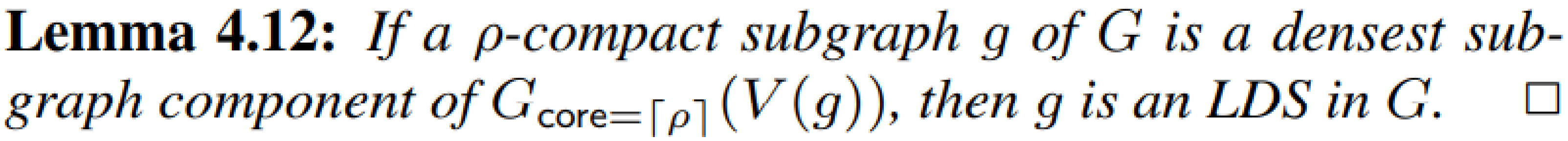


For a ρ-compact subgraph g, let $G_{\text{core}=\lceil \rho \rceil}(V(g))$ be the connected component of the $\lceil \rho \rceil$-core of G that includes all the nodes in $V(g)$. $G_{\text{core}=\lceil \rho \rceil}(V(g))$ exists by Lemma 4.10. We prove that any LDS g in G is an LDS in $G_{\text{core}=\lceil \rho \rceil}(V(g))$, and vice versa.

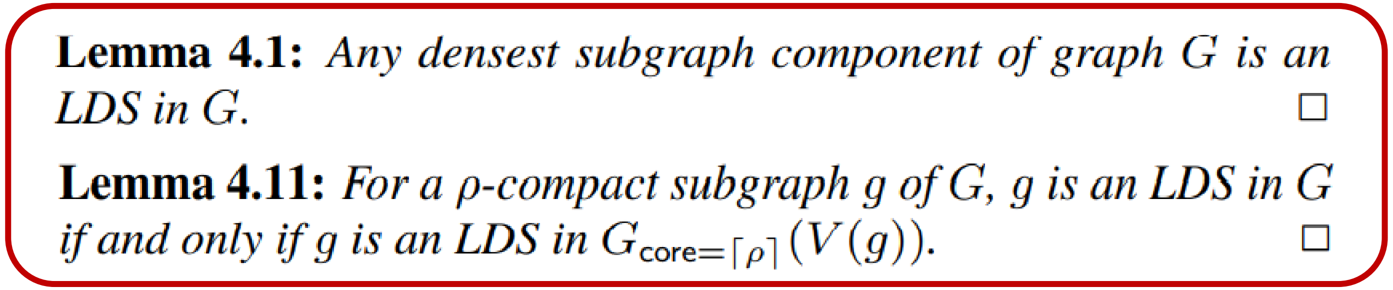


Lemma 4.11: *For a ρ-compact subgraph g of G, g is an LDS in G if and only if g is an LDS in $G_{\text{core}=\lceil \rho \rceil}(V(g))$.* $\square$

The Optimized Algorithm

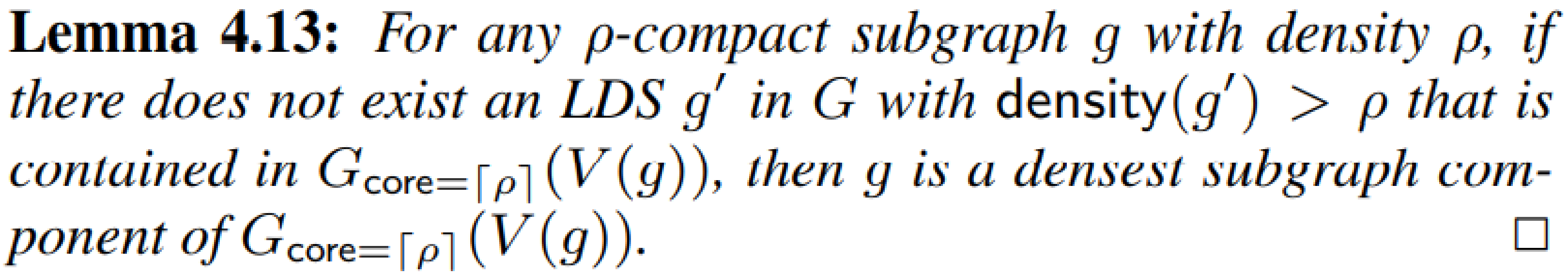


Lemma 4.12: *If a ρ-compact subgraph g of G is a densest subgraph component of $G_{\text{core}=\lceil\rho\rceil}(V(g))$, then g is an LDS in G.* $\square$



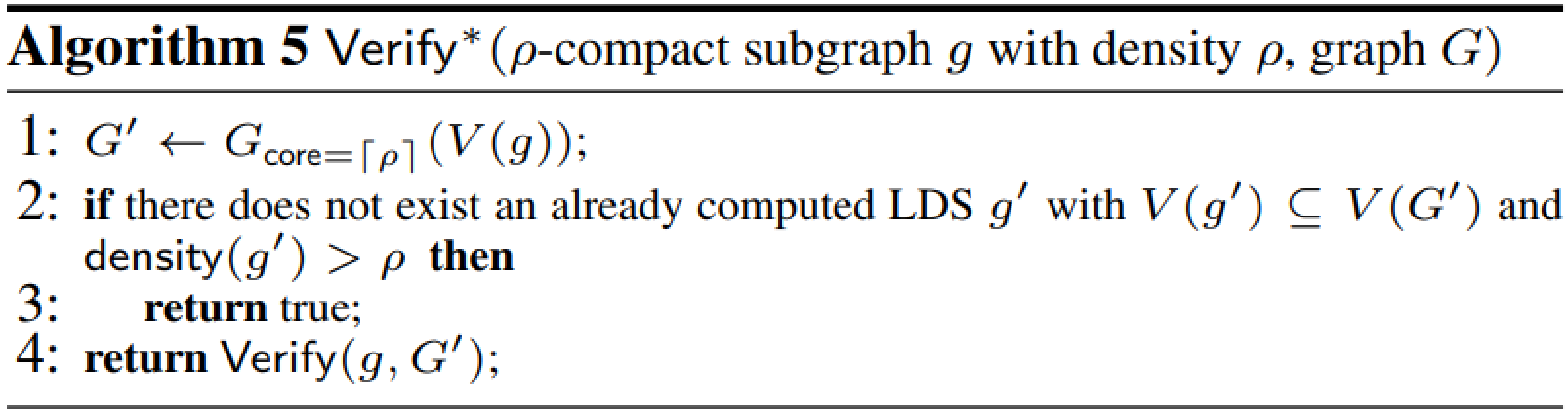
Lemma 4.1: *Any densest subgraph component of graph G is an LDS in G.* $\square$

Lemma 4.11: *For a ρ-compact subgraph g of G, g is an LDS in G if and only if g is an LDS in $G_{\text{core}=\lceil\rho\rceil}(V(g))$.* $\square$



Lemma 4.13: *For any ρ-compact subgraph g with density ρ, if there does not exist an LDS g' in G with $\text{density}(g') > \rho$ that is contained in $G_{\text{core}=\lceil\rho\rceil}(V(g))$, then g is a densest subgraph component of $G_{\text{core}=\lceil\rho\rceil}(V(g))$.* $\square$

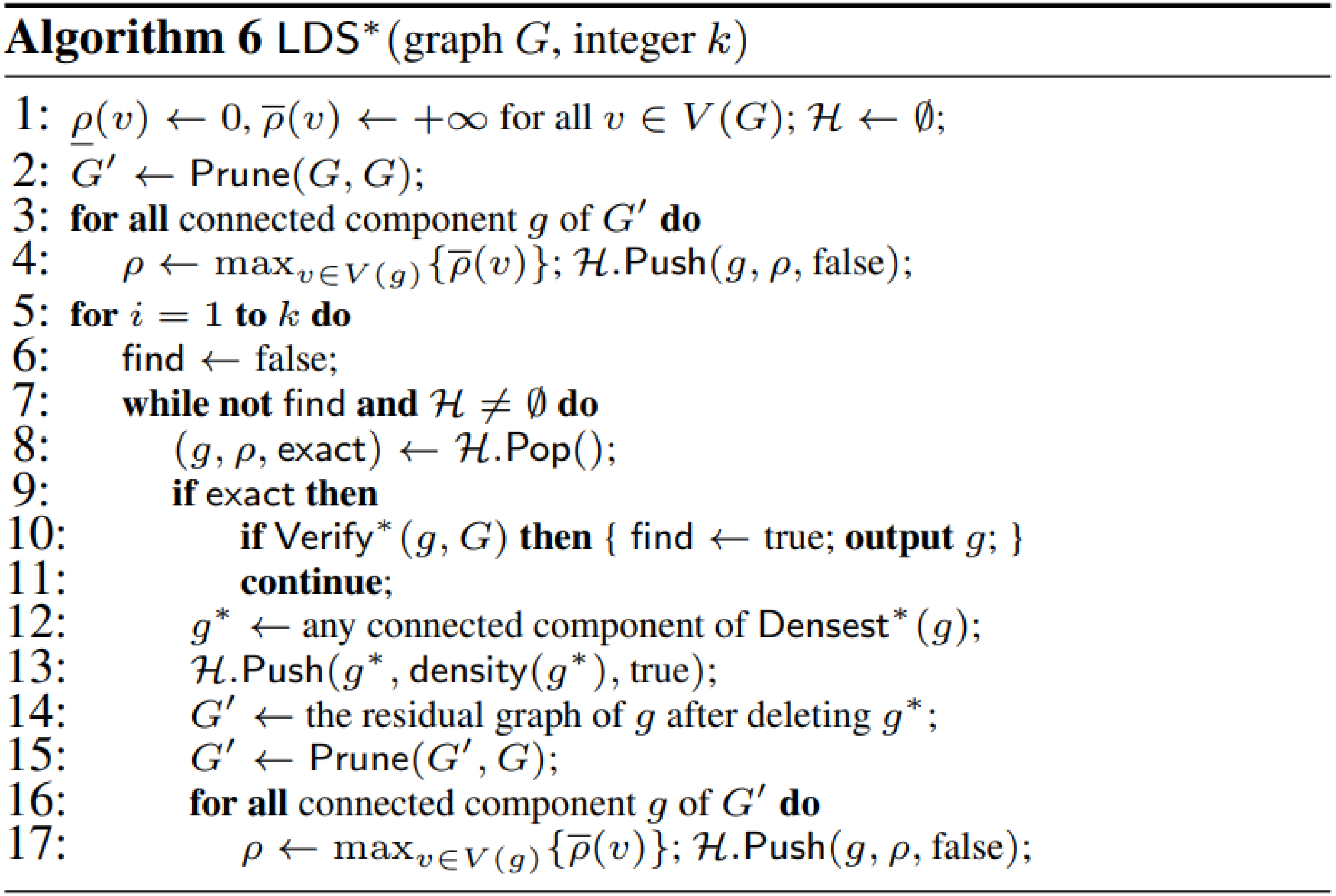
The Verify Algorithm



Algorithm 5 Verify* (ρ-compact subgraph g with density ρ, graph G)

```
$G' \leftarrow G_{\mathsf{core}=\lceil \rho \rceil}(V(g))$;
if there does not exist an already computed LDS $g'$ with $V(g') \subseteq V(G')$ and density$(g') > \rho$ then
    return true;
return Verify$(g, G')$;
```

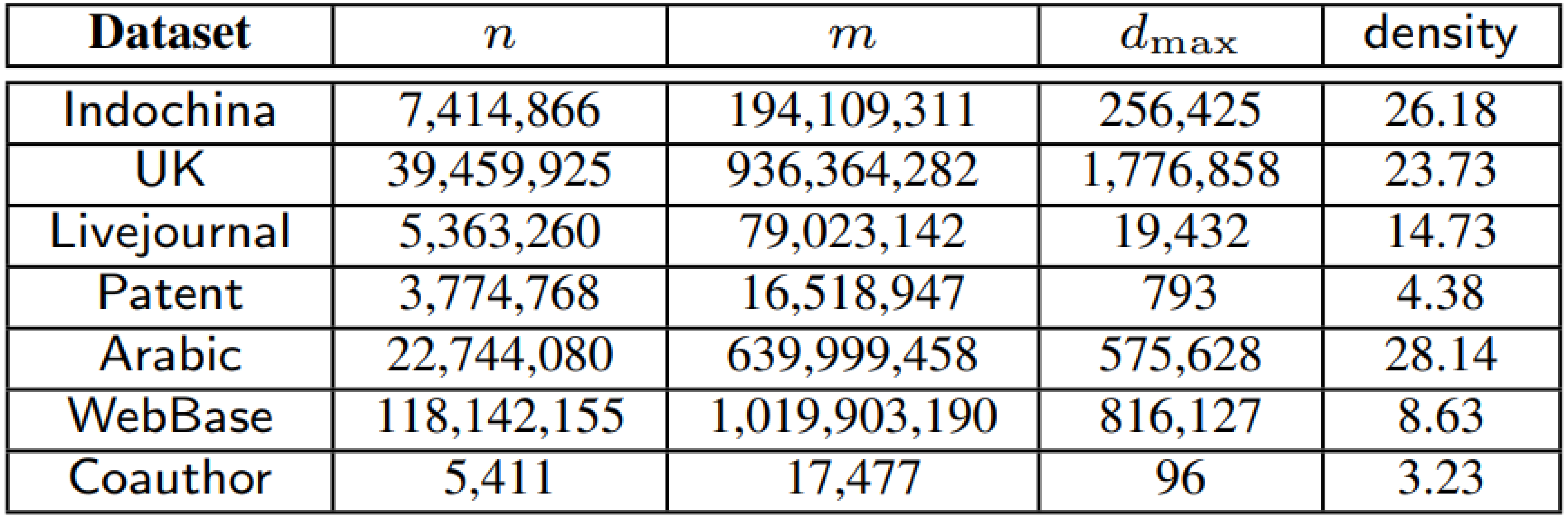
The LDS Algorithm



Algorithm 6 LDS* (graph G, integer k)

```
$\underline{\rho}(v) \leftarrow 0, \overline{\rho}(v) \leftarrow +\infty$ for all $v \in V(G)$; $\mathcal{H} \leftarrow \emptyset$;
$G' \leftarrow$ Prune$(G, G)$;
for all connected component $g$ of $G'$ do
    $\rho \leftarrow \max_{v \in V(g)} \{\overline{\rho}(v)\}$; $\mathcal{H}$.Push$(g, \rho$, false$)$;
for $i = 1$ to $k$ do
    find $\leftarrow$ false;
    while not find and $\mathcal{H} \neq \emptyset$ do
        $(g, \rho$, exact$) \leftarrow \mathcal{H}$.Pop$()$;
        if exact then
            if Verify*$(g, G)$ then { find $\leftarrow$ true; output $g$; }
            continue;
        $g^* \leftarrow$ any connected component of Densest*$(g)$;
        $\mathcal{H}$.Push$(g^*$, density$(g^*)$, true$)$;
        $G' \leftarrow$ the residual graph of $g$ after deleting $g^*$;
        $G' \leftarrow$ Prune$(G', G)$;
        for all connected component $g$ of $G'$ do
            $\rho \leftarrow \max_{v \in V(g)} \{\overline{\rho}(v)\}$; $\mathcal{H}$.Push$(g, \rho$, false$)$;
```

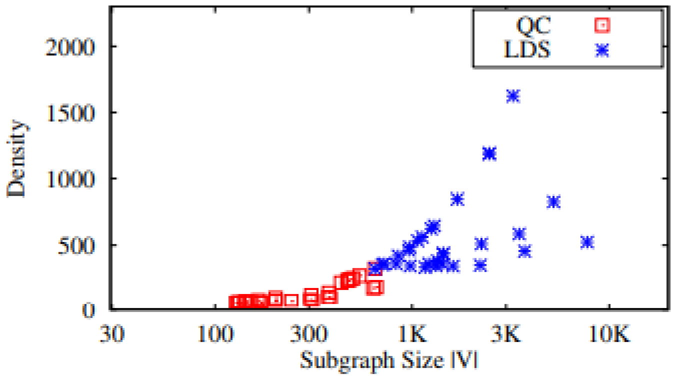
Evaluation



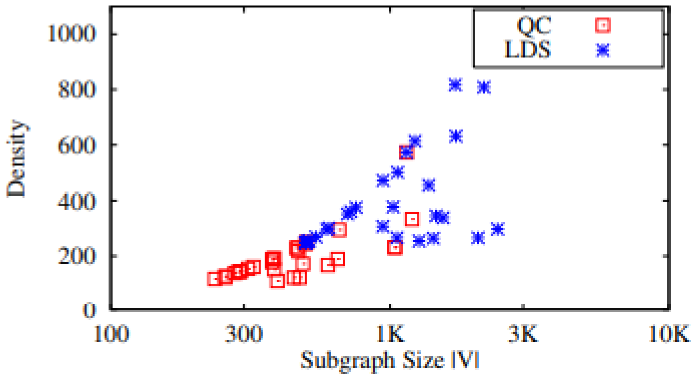
Dataset	n	m	$d_{\max}$	density
Indochina	7,414,866	194,109,311	256,425	26.18
UK	39,459,925	936,364,282	1,776,858	23.73
Livejournal	5,363,260	79,023,142	19,432	14.73
Patent	3,774,768	16,518,947	793	4.38
Arabic	22,744,080	639,999,458	575,628	28.14
WebBase	118,142,155	1,019,903,190	816,127	8.63
Coauthor	5,411	17,477	96	3.23

Table 1: Datasets

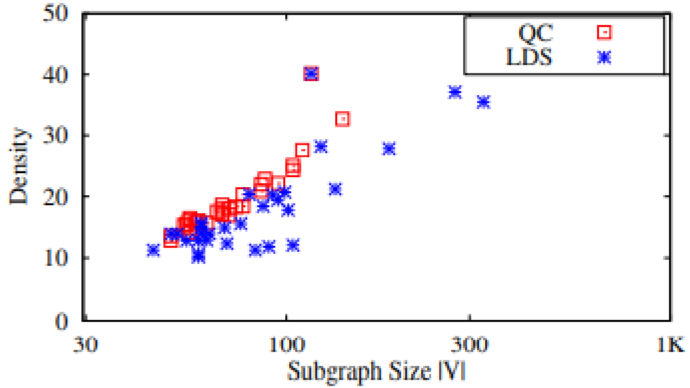
Density Testing



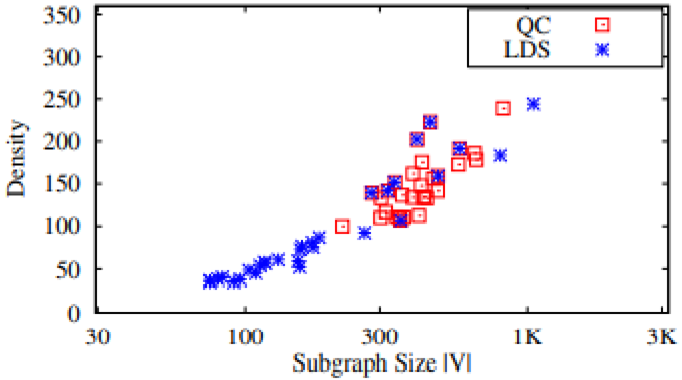
(a) Arabic



(b) WebBase

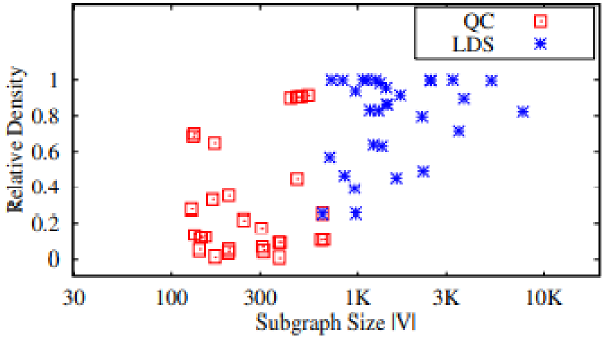


(c) Patent

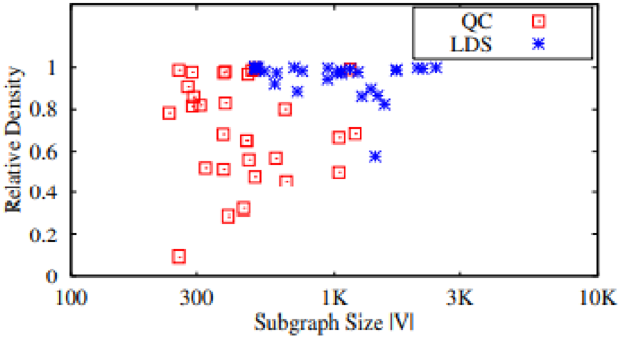


(d) Livejournal

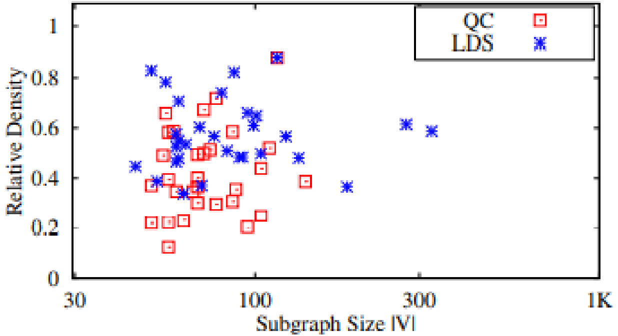
Relative Density Testing



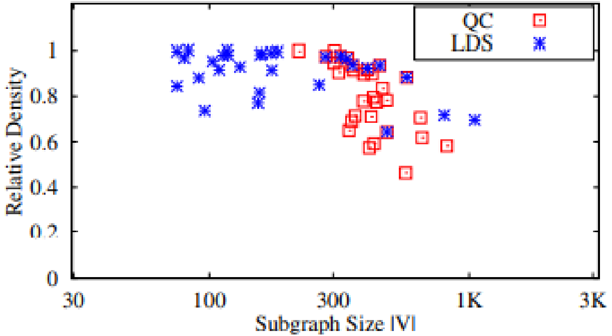
(a) Arabic



(b) WebBase



(c) Patent



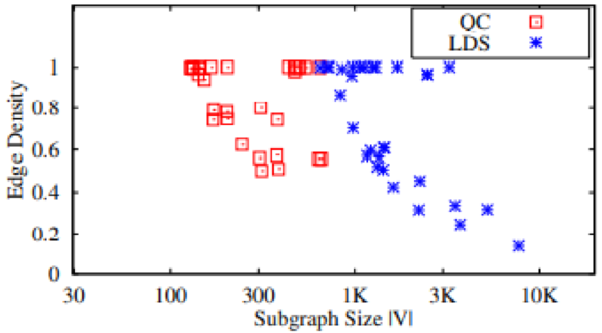
(d) Livejournal

Relative Density ρ_r. Relative density is a popular graph cluster-fitness measure that takes both the inter and intra edges of a subgraph into consideration [33]. Intuitively, a subgraph with high relative density indicates that the density of the subgraph is high and the density of its nearby region is relatively low. The relative density is formally defined as follows:

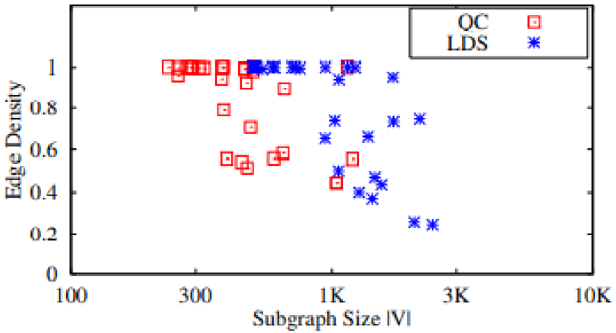
$$\rho_r(g, G) = \frac{|E(g)|}{|E(g)| + |E'(g, G)|} \tag{2}$$

where $E'(g, G) = \{(u, v) | (u, v) \in E(G), u \in V(g), v \notin V(g)\}$ is the set of inter-edges for subgraph g in G.

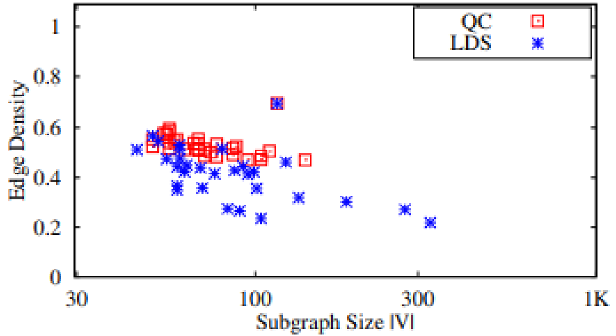
Edge Density Testing



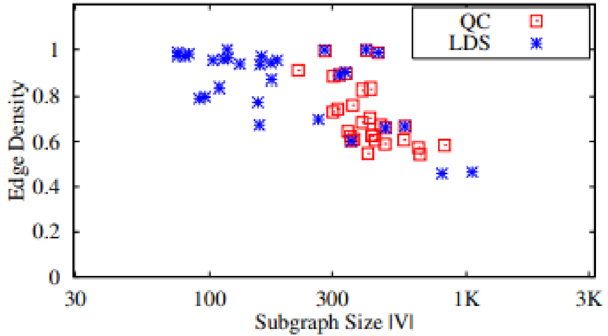
(a) Arabic



(b) WebBase



(c) Patent

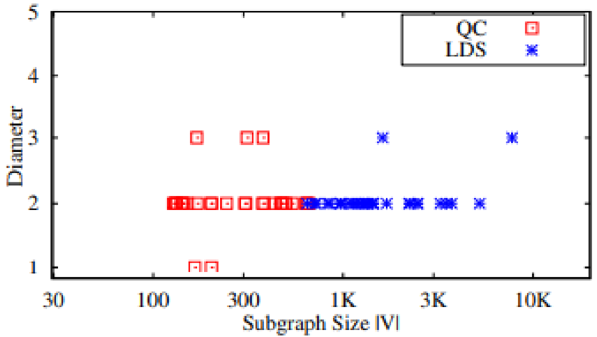


(d) Livejournal

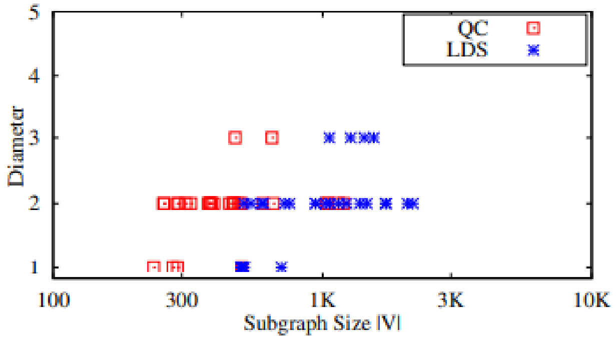
Edge Density ρ_e. Edge density is the ratio of the number of edges in a graph to the number of edges in a complete graph with the same set of nodes, which is defined as:

$$\rho_e(g) = \frac{2 \times |E(g)|}{|E(g)| \times (|E(g)| - 1)} \tag{3}$$

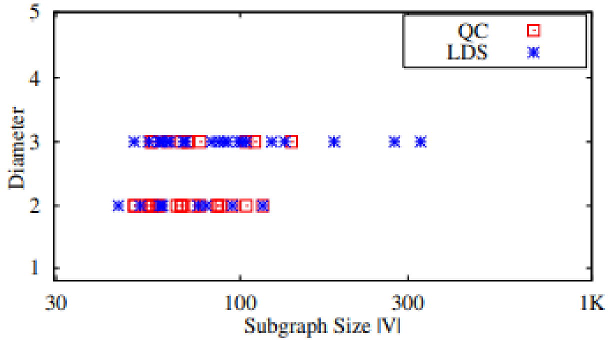
Diameter Testing



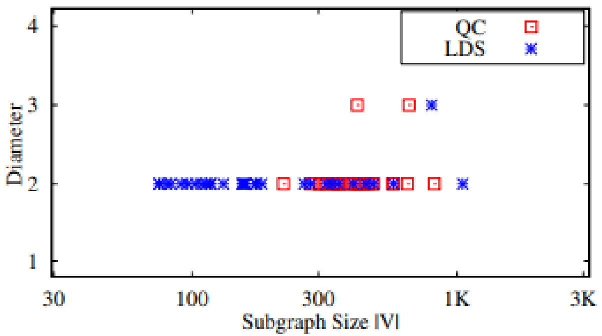
(a) Arabic



(b) WebBase



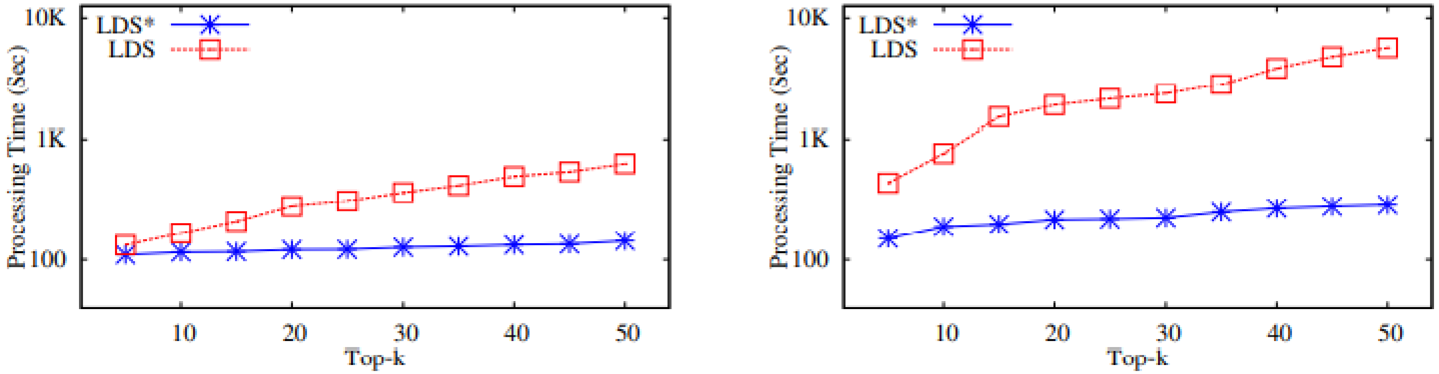
(c) Patent



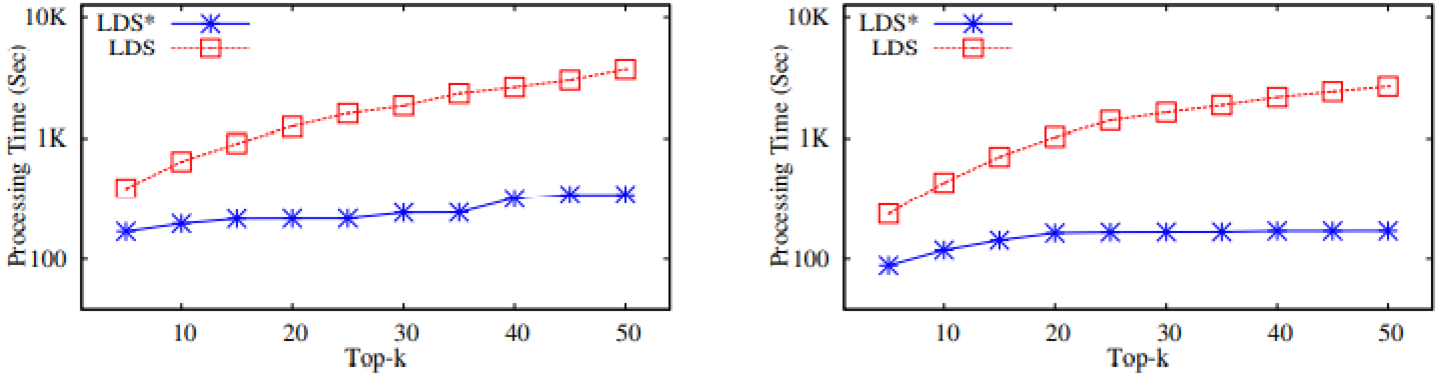
(d) Livejournal

Diameter. The diameter of a graph is the longest distance of all pairs of nodes in the graph, where the distance of two nodes is the minimum number of hops to reach from one node to another.

Efficiency Testing

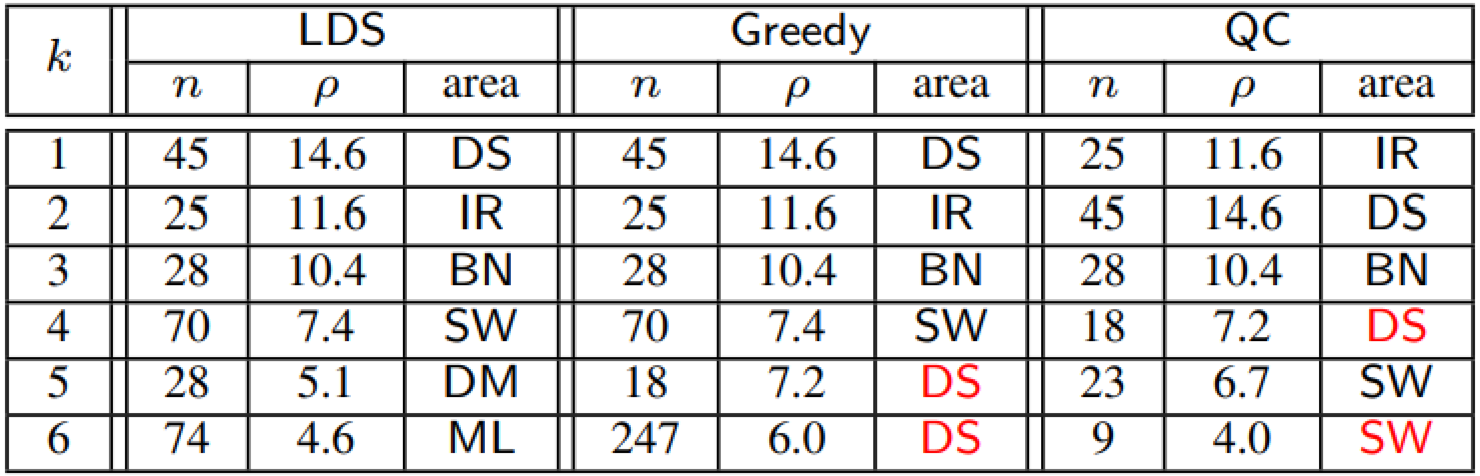


(a) Indochina (b) Arabic



(c) UK (d) WebBase

Case Study



k	LDS			Greedy			QC		
	n	ρ	area	n	ρ	area	n	ρ	area
1	45	14.6	DS	45	14.6	DS	25	11.6	IR
2	25	11.6	IR	25	11.6	IR	45	14.6	DS
3	28	10.4	BN	28	10.4	BN	28	10.4	BN
4	70	7.4	SW	70	7.4	SW	18	7.2	DS
5	28	5.1	DM	18	7.2	DS	23	6.7	SW
6	74	4.6	ML	247	6.0	DS	9	4.0	SW

- **Database Systems (DS), Information Retrieval (IR), Machine Learning (ML), Data Mining (DM), Bayesian Networks(BN), and Semantic Web (SW)**

Conclusion

- In this paper, they study the problem of discovering the top-k locally densest subgraphs (LDSes) in a graph, which can be used to identify the local dense regions of a graph, and can be applied in a variety of application domains.
- They provide a parameter-free definition of an LDS with several useful properties.
- They show that the LDSes of a graph can be computed in polynomial time, and propose three novel optimization strategies to improve the algorithm.
- They conduct extensive experiments using seven real datasets to demonstrate the effectiveness and efficiency of their approach.

THANK YOU

Xiaojia Xu