On Directed Densest Subgraph Detection

Kai Yao, Xin Yang, and Lijun Chang

The University of Sydney, Sydney, Australia
kai.yao@sydney.edu.au, xyan5144@uni.sydney.edu.au,
lijun.chang@sydney.edu.au

Abstract. The well-studied directed densest subgraph problem aims to find two (possibly overlapping) vertex subsets S^* and T^* in a given directed graph $G = (V, E)$ such that $\rho(S,T) = \frac{|E(S,T)|}{\sqrt{|S||T|}}$ is maximized; here $E(S,T)$ denotes the set of edges from vertices of S to T in G. This problem is polynomial-time solvable, and both exact algorithms and approximation algorithms have been proposed in the literature. However, the existing exact algorithms are time-consuming, while the existing approximation algorithms often yield trivial solutions that consist of the highest-degree vertex and its in-neighbors or out-neighbors. Moreover, there is nothing special about geometric mean that is adopted in the existing density measure for combining $\frac{|E(S,T)|}{|S|}$ and $\frac{|E(S,T)|}{|T|}$. In this paper, we explore alternative density measures and propose corresponding algorithms, for directed densest subgraph identification. Specifically, we introduce three density measures that combine $\frac{|E(S,T)|}{|S|}$ and $\frac{|E(S,T)|}{|T|}$ by harmonic mean, arithmetic mean, and minimum mean, respectively. Based on these density measures, we formulate the harmonic mean-based directed densest subgraph (HDDS) problem, the arithmetic mean-based directed densest subgraph (ADDS) problem, and the minimum mean-based directed densest subgraph (MDDS) problem. We then propose a 2-approximation algorithm for HDDS, a 2-approximation algorithm for ADDS, and a heuristic algorithm for MDDS; our HDDS and MDDS algorithms run in linear time to the input graph size. Extensive empirical studies on large real-world directed graphs show that our ADDS algorithm produces similar trivial results as the existing approximation algorithm, and our HDDS and MDDS algorithms generate nontrivial and much better solutions and scale to large graphs.

Keywords: Directed densest subgraph · Directed graphs

1 Introduction

Directed graphs are essential for modeling relationships in real-world contexts like social networks, information networks, and biological networks [1]. Recently, identifying the densest subgraph in a directed graph has gained substantial attention [2–6]. This problem finds applications in areas like fake follower detection [6], community detection [4], and graph compression [7]. Instead of returning a vertex subset (or vertex-induced subgraph) as done in densest subgraph identification over undirected graphs, densest subgraph identification over directed graphs typically returns two (possibly overlapping) vertex subsets S and T such that there are a lot of edges from S to T. Specifically, Kannan and Vinay [2] defined the density of $S, T \subseteq V$ as $\rho(S,T) = \frac{|E(S,T)|}{\sqrt{|S||T|}}$, where $E(S,T)$ is the set of edges from S to T in the input graph. The pair of vertex subsets S^* and T^* that maximizes $\rho(S^*,T^*)$ is referred to as the directed densest subgraph, which is polynomial-time computable. This density measure has been followed and used by all subsequent work on directed densest subgraph identification [2, 6, 8, 9]. Both exact algorithms and approximation algorithms have been proposed in the literature. However, the existing exact algorithms are time-consuming due to lots of maximum flow computations and thus are not suitable for handling large graphs. Among the existing approximation algorithms, Core-Approx [6] is a state-of-the-art algorithm that achieves a good balance between the time complexity and the approximation ratio; specifically, Core-Approx computes a 2-approximate result in $O(\sqrt{m}(n+m))$ time, for a graph with n vertices and m edges. However, our empirical studies show that Core-Approx usually reports trivial solutions that consist of the highest-degree vertex and its in-neighbors or out-neighbors, due to the power-law degree distribution prevailing in real-world graphs. Such skewed and trivial solutions severely curtail the meaningful insights attainable through directed densest graph extraction.

In this paper, we aim to explore alternative density measures and propose corresponding algorithms, for directed densest subgraph identification. We observe that the density formulated by Kannan and Vinay [2] can be regarded as the *geometric mean* of $\frac{|E(S,T)|}{|S|}$ and $\frac{|E(S,T)|}{|T|}$ that measure the density for the S part and for the T part, respectively. There is nothing special about geometric mean, and we can use other mean measures instead. Thus, we introduce three novel density measures, namely the *harmonic mean-based density*, *arithmetic mean-based density*, and *minimum mean-based density* that, respectively, apply the harmonic mean, arithmetic mean, and minimum mean to $\frac{|E(S,T)|}{|S|}$ and $\frac{|E(S,T)|}{|T|}$. For each of these density measures, we formulate a distinct directed densest subgraph problem, whose objective is to identify S and T that maximize the corresponding density measure. This endeavor yields three specific problem instances: the harmonic mean-based directed densest subgraph (HDDS) problem, the arithmetic mean-based directed densest subgraph (ADDS) problem, and the minimum mean-based directed densest subgraph (MDDS) problem. To solve the HDDS problem, we establish the equivalence between the problem over directed graphs and the extensively studied densest subgraph problem over undirected graphs. Subsequently, existing exact and approximation algorithms for undirected densest subgraph identification can be directly applied. Regarding the ADDS problem, we propose a 2-approximation algorithm A-Approx that runs in $O(\sqrt{m}(n+m))$ time. For the MDDS problem, we propose a heuristic algorithm M-Approx that runs in $O(n+m)$ time.

Our main contributions are summarized as follows.

- We formulate three alternative density measures for directed densest subgraph identification (Section 2).

– We establish the equivalence between the HDDS problem and the extensively studied densest subgraph problem over undirected graphs (Section 3).
– We propose a 2-approximation algorithm A-Approx for the ADDS problem that runs in $O(\sqrt{m}(n+m))$ time (Section 4).
– We propose a heuristic algorithm M-Approx for the MDDS problem that runs in $O(n+m)$ time (Section 5).

We conduct extensive experiments on large real-world directed graphs to demonstrate the effectiveness and efficiency of our algorithms in Section 6. The results show that A-Approx produces similar trivial solutions as the existing algorithm Core-Approx, and our H-Approx and M-Approx algorithms generate nontrivial and much better solutions and scale to large graphs. Related works are discussed in Section 7. Proofs of all theorems and lemmas are omitted due to limit of space.

2 Preliminaries

Let $G=(V,E)$ represent a directed graph, where $n=|V|$ denotes the number of vertices and $m=|E|$ denotes the number of edges in the graph G. Given two (possibly overlapping) vertex subsets S and T, we use $E(S,T)$ to denote the set of all edges that connect vertices from set S to set T, formally defined as $E(S,T)=E\cap(S\times T)$. We use the term (S,T)-induced subgraph to refer to the subgraph of G that is induced by vertices S and T and edges $E(S,T)$. This subgraph is denoted as $G[S,T]$. We use $d_G^+(v)$ and $d_G^-(v)$ to denote v's out-degree and in-degree in G, respectively. Next, we present possible density measures for (S,T)-induced subgraphs of a directed graph by leveraging the combination of two components $\frac{|E(S,T)|}{|S|}$ and $\frac{|E(S,T)|}{|T|}$.

Definition 1 (Geometric Mean-based Density [2]). *Given a directed graph $G=(V,E)$ and two vertex sets $S,T\subseteq V$, the geometric mean-based density of the (S,T)-induced subgraph $G[S,T]$ is defined as:*

$$\rho_g(S,T)=\sqrt{\frac{|E(S,T)|}{|S|}\cdot\frac{|E(S,T)|}{|T|}}=\frac{|E(S,T)|}{\sqrt{|S||T|}}$$

Definition 2 (Harmonic Mean-based Density). *Given a directed graph $G=(V,E)$ and two vertex sets $S,T\subseteq V$, the harmonic mean-based density of the (S,T)-induced subgraph $G[S,T]$ is defined as:*

$$\rho_h(S,T)=\left(\frac{\left(\frac{|E(S,T)|}{|S|}\right)^{-1}+\left(\frac{|E(S,T)|}{|T|}\right)^{-1}}{2}\right)^{-1}=\frac{2|E(S,T)|}{|S|+|T|}$$

Definition 3 (Arithmetic Mean-based Density). *Given a directed graph $G=(V,E)$ and two vertex sets $S,T\subseteq V$, the arithmetic mean-based density of the (S,T)-induced subgraph $G[S,T]$ is defined as:*

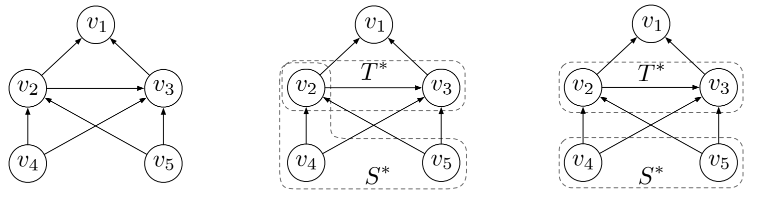
$$\rho_a(S,T)=\frac{\frac{|E(S,T)|}{|S|}+\frac{|E(S,T)|}{|T|}}{2}=\frac{(|S|+|T|)|E(S,T)|}{2|S||T|}$$

Definition 4 (Minimum Mean-based Density). *Given a directed graph $G = (V, E)$ and two vertex sets $S, T \subseteq V$, the minimum mean-based density of the (S, T)-induced subgraph $G[S, T]$ is defined as:*

$$\rho_m(S,T) = \min\left\{\frac{|E(S,T)|}{|S|}, \frac{|E(S,T)|}{|T|}\right\} = \frac{|E(S,T)|}{\max\{|S|,|T|\}}$$

Problem Definition: Given a directed graph $G = (V, E)$ and one of the above density measures, we aim to find the (S^*, T^*)-induced subgraph whose density is the highest among all the possible (S, T)-induced subgraphs of G.

Adopting different density measures leads to four specific problems: the geometric mean-based directed densest subgraph (GDDS) problem, the harmonic mean-based directed densest subgraph (HDDS) problem, the arithmetic mean-based directed densest subgraph (ADDS) problem, and the minimum mean-based directed densest subgraph (MDDS) problem. We use ρ_g^*, ρ_h^*, ρ_a^* and ρ_m^* to represent the density of the densest subgraph for the four problems, respectively.



(a) Directed graph G (b) GDDS & ADDS (c) MDDS & HDDS

Fig. 1. Illustration of different density measures

Example 1. Consider the directed graph G in Figure 1(a). Both the GDDS and ADDS are the subgraph induced by $S^* = \{v_2, v_4, v_5\}$ and $T^* = \{v_2, v_3\}$, and $\rho_g^* = \frac{5}{\sqrt{2\times3}} = 2.04$ and $\rho_a^* = \frac{(2+3)\times5}{2\times2\times3} = 2.08$ (see Figure 1(b)). Both the MDDS and HDDS are the subgraph induced by $S^* = \{v_4, v_5\}$ and $T^* = \{v_2, v_3\}$, and $\rho_m^* = \frac{4}{\max\{2,2\}} = 2$ and $\rho_h^* = \frac{2\times4}{2+2} = 2$ (see Figure 1(c)). Note that the subgraph induced by $S^* = \{v_2, v_4, v_5\}$ and $T^* = \{v_2, v_3\}$ also has the harmonic mean-based density of 2 and thus is also an HDDS.

Presently, the mainstream approach to capturing the directed densest subgraph is through the use of geometric mean-based density, with both exact and approximation algorithms being designed. In this paper, we initiate the study of directed densest subgraph identification for other alternative density measures. Approximation/heuristic algorithms are proposed in the following, while designing efficient exact algorithms is left as our future work.

3 Harmonic Mean-based Directed Densest Subgraph

In this section, we demonstrate the equivalence between the HDDS problem over directed graphs and the extensively studied densest subgraph problem over undirected graphs [10]. Consequently, the existing exact and approximation algorithms for the densest subgraph problem over undirected graphs can be directly employed to tackle the HDDS problem.

Let us begin by discussing how to transform a directed graph $G = (V, E)$ into an undirected bipartite graph $G' = (V_1, V_2, E')$. We create two sets of vertices $V_1 = V_2 = V$. Note that, to distinguish the same vertex in V_1 and V_2, for each $v \in V_1$ we use v' to denote its counterpart in V_2. Then, for each directed edge (u, v) in G, we add an undirected edge to E' by linking $u \in V_1$ and $v' \in V_2$. This implies that for each vertex $u \in V$, its out-degree in G is equal to the degree of vertex u in G', and its in-degree in G is equal to the degree of vertex u' in G'. Furthermore, in the bipartite graph $G' = (V_1, V_2, E')$, the subgraph induced by $S \subseteq V_1$ and $T \subseteq V_2$ is denoted as $G'[S \cup T]$. With this transformation, we can now establish the following lemma, describing the connection between the HDDS problem over G and the densest subgraph problem over G'; note that, we will treat the undirected bipartite graph G' simply as an undirected graph, by ignoring its bipartiteness.

Lemma 1. *For a given directed graph $G = (V, E)$ and any two sets $S, T \subseteq V$, the harmonic mean-based density of $D = G[S, T]$ in the directed graph G is exactly twice the density of $H = G'[S \cup T]$ in the undirected graph G'.*

Algorithm 1: H-Approx

```
   Input: A directed graph $G = (V, E)$
   Output: A 2-approximate HDDS $G[S, T]$
1  Transform $G$ to an undirected graph $G' = (V_1, V_2, E')$;
2  $(S, T) \leftarrow$ Approx$(G')$;
3  return $G[S, T]$;
   Procedure Approx$(G = (V_1, V_2, E))$
4  $S \leftarrow V_1$, $T \leftarrow V_2$, $\rho^* \leftarrow \frac{|E(V_1,V_2)|}{|V_1|+|V_2|}$;
5  while $G \neq \emptyset$ do
6      $u \leftarrow$ the vertex with the smallest degree in $G$;
7      if $u \in V_1$ then $V_1 \leftarrow V_1 \setminus \{u\}$ ;
8      else $V_2 \leftarrow V_2 \setminus \{u\}$ ;
9      if $\frac{|E(V_1,V_2)|}{|V_1|+|V_2|} > \rho^*$ then
10         $S \leftarrow V_1$, $T \leftarrow V_2$;
11         $\rho^* \leftarrow \frac{|E(V_1,V_2)|}{|V_1|+|V_2|}$;
12 return $(S, T)$;
```

We propose to first compute the dense subgraph $H = G'[S \cup T]$ in G' using the state-of-the-art 2-approximation algorithm Approx [10], and then construct the directed dense subgraph $D = G[S, T]$ in G. As per Lemma 1, it is established that subgraph $D = G[S, T]$ achieves a 2-approximation ratio to the HDDS in G. The pseudo-code of our algorithm H-Approx is shown in Algorithm 1. Firstly, it transforms the directed graph $G = (V, E)$ to the undirected graph $G' =$

(V_1, V_2, E') (Line 1). Then, it invokes Approx to obtain the dense subgraph $G'[S \cup T]$ in G' (Line 2). Based on the returned S and T, it returns the subgraph $D = G[S, T]$ (Line 3). The procedure Approx uses S and T to record the best solution found so far, and ρ^* to record its density. Firstly, the best solution is initialized as the original graph (Line 4). Then, it iteratively deletes the smallest-degree vertex and its associated edges from the remaining graph (Lines 6-8). After deleting a vertex, it computes the density of the remaining graph and updates S, T, and ρ^* if necessary (Lines 9-11). Finally, it returns (S, T) such that $G[S \cup T]$ has the highest density among all the intermediate subgraphs (Line 12). The time complexity of H-Approx is $\mathcal{O}(n + m)$ by using the bin-sort technique [11].

4 Arithmetic Mean-based Directed Densest Subgraph

In this section, we present a 2-approximation algorithm for the ADDS problem. Before delving into the details of the algorithm, we first introduce some concepts and prove some important properties that form the basis for our approach.

Definition 5 ((x, y)-core [6]). *Given a directed graph $G = (V, E)$, an (S, T)-induced subgraph $H = G[S, T]$ is called an (x, y)-core if it satisfies:*

(1) $\forall u \in S, d^+_H(u) \geq x$ and $\forall v \in T, d^-_H(v) \geq y$;
(2) $\nexists H'$ s.t. $H \subset H'$ and H' satisfies (1).

We call (x, y) the *core number pair* of the (x, y)-core, abbreviated as *cn-pair*. We call a cn-pair (x, x) as the maximum equal cn-pair, if x has the maximum value among all possible (x, x)-cores.

Definition 6 (Key cn-pair [6]). *Given a directed graph $G = (V, E)$ and its maximum equal cn-pair (γ, γ), the cn-pair of an (x, y)-core is a key cn-pair, if one of the following conditions is satisfied:*

(1) if $x \leq \gamma$, there is no (x, y')-core in G s.t. $y' > y$;
(2) if $y \leq \gamma$, there is no (x', y)-core in G s.t. $x' > x$.

Example 2. Consider the directed graph G in Figure 1(a). The subgraph induced by $S = \{v_4, v_5\}$ and $T = \{v_2, v_3\}$ is a $(2, 2)$-core, and $(2, 2)$ is the maximum equal cn-pair in G. The subgraph induced by $S = \{v_2, v_4, v_5\}$ and $T = \{v_3\}$ is a $(1, 3)$-core, which is a key cn-pair.

Lemma 2. *Given a directed graph $G = (V, E)$ and its ADDS $D = G[S^*, T^*]$ with density ρ^*, let $\frac{|S^*|}{|T^*|} = a$, it satisfies the following properties:*

(1) For any subset $U_S \subset S^*$, removing U_S from S^* will result in the removal of at least $\frac{2\rho^*|U_S|}{(1+a)^2}$ edges from D;
(2) For any subset $U_T \subset T^*$, removing U_T from T^* will result in the removal of at least $\frac{2\rho^*|U_T|}{(1+\frac{1}{a})^2}$ edges from D.

Corollary 1. *Given a directed graph $G = (V, E)$ and its ADDS $D = G[S^*, T^*]$ with density ρ^*, let $\frac{|S^*|}{|T^*|} = a$, then D is contained in the $(\lceil \frac{2\rho^*}{(1+a)^2} \rceil, \lceil \frac{2\rho^*}{(1+\frac{1}{a})^2} \rceil)$-core.*

Lemma 3. *Given a directed graph $G = (V, E)$ and an (x, y)-core $H = G[S, T]$, we have that $\rho_a(S, T) \geq \frac{x+y}{2}$.*

Definition 7 (Max-sum cn-pair). *Given a directed graph $G = (V, E)$, a cn-pair (x, y) is called max-sum cn-pair, if $x+y$ achieves the maximum value among all the possible (x, y)-cores. We denote the max-sum cn-pair by (x^*, y^*).*

Lemma 4. *(x^*, y^*)-core is a 2-approximate solution to the arithmetic mean-based densest subgraph problem.*

Algorithm 2: A-Approx

```
   Input: A directed graph $G = (V, E)$
   Output: A 2-approximate ADDS $G[S, T]$, i.e., $(x^*, y^*)$-core
1  $x^* \leftarrow 0$, $y^* \leftarrow 0$;
2  Compute the maximum equal cn-pair $(\gamma, \gamma)$ by the peeling algorithm [3];
3  for $x \leftarrow 1$ to $\gamma$ do
4      $y \leftarrow \mathsf{MaxY}(G, x)$;
5      if $x + y > x^* + y^*$ then $x^* \leftarrow x$, $y^* \leftarrow y$;
6  for $y \leftarrow 1$ to $\gamma$ do
7      $x \leftarrow \mathsf{MaxX}(G, y)$;
8      if $x + y > x^* + y^*$ then $x^* \leftarrow x$, $y^* \leftarrow y$;
9  return $(x^*, y^*)$-core;
   Procedure MaxY$(G, x)$
10 $S \leftarrow V$, $T \leftarrow V$, $y_{max} \leftarrow 0$, $y \leftarrow x^* + y^* - x + 1$;
11 if $y > \arg\max_{u \in T} d^-_G(u)$ then return $y_{max}$;
12 while $|E| > 0$ do
13     while $\exists u \in T$ s.t. $d^-_G(u) < y$ do
14         $E \leftarrow E \setminus \{(v, u) \mid v \in S\}$, $T \leftarrow T \setminus \{u\}$;
15         while $\exists v \in S$ s.t. $d^+_G(v) < x$ do
16             $E \leftarrow E \setminus \{(v, u) \mid u \in T\}$, $S \leftarrow S \setminus \{v\}$;
17     if $|E| > 0$ then $y_{max} \leftarrow y$;
18     $y \leftarrow y + 1$;
19 return $y_{max}$;
   Procedure MaxX$(G, y)$
20 Reuse lines 10-19 by interchanging $u$ with $v$, $S$ with $T$, $x$ with $y$, and changing
     $y_{max}$ to $x_{max}$;
```

Next, we will introduce how to compute (x^*, y^*)-core efficiently. One straightforward approach is to compute all (x, y)-cores in G and return the one with the maximum $x + y$ value. The time complexity of this approach is $\mathcal{O}(n(n + m))$, since there are n possible values of x and it takes $\mathcal{O}(n + m)$ to compute all

(x, y)-cores for each specific x. Based on the concepts of maximum equal cn-pair and key cn-pair, we propose a more efficient approach to compute (x^*, y^*)-core. As shown in our algorithm A-Approx (i.e., Algorithm 2), it firstly computes the maximum equal cn-pair (γ, γ) by using the peeling algorithm in [3], which iteratively peels vertices that have the minimum in-degrees or out-degrees (Line 2). Then, it enumerates x and y from 1 to γ to search all the key cn-pairs (Lines 3-8). Finally, the (x^*, y^*)-core is returned (Line 9).

Given an input x, the procedure MaxY computes the key cn-pair whose first element is x. In MaxY, it first initializes S, T, y_{max}, and y, where y is set to $x^* + y^* - x + 1$ (Line). Then, in the while loop, if there is vertex $u \in T$ with in-degree less than y, the algorithm deletes it (Lines 13-14). The deletion of u may make some vertices' out-degrees become less than x, so the algorithm deletes these vertices as well (Lines 15-16). After that, we update y_{max} and increase y by 1 (Lines 17-18). Finally, we get the maximum value of y. Similarly, we have a procedure MaxX to get the key cn-pair whose second element is a given y.

Theorem 1. *The time complexity of* A-Approx *(Algorithm 2) is* $\mathcal{O}(\sqrt{m}(n+m))$.

5 Minimum Mean-based Directed Densest Subgraph

This section is dedicated to a theoretical analysis of the MDDS problem. Subsequently, leveraging the insights from this analysis, we proceed to devise an approximation algorithm tailored specifically for addressing this problem. Firstly, we have the following lemma.

Lemma 5. *Given a directed graph $G = (V, E)$, there exists an (S^*, T^*)-induced subgraph $G[S^*, T^*]$ with the highest minimum mean-based density and $|S^*| = |T^*|$.*

The above lemma highlights that the optimal solution for the MDDS problem exhibits an equal size of vertex sets S and T. Motivated by this insight, we endeavor to find a high-quality approximate solution by enforcing $|S| = |T|$. To achieve this, we introduce an approximation algorithm named M-Approx. The main idea is to maintain two vertex sets S and T, both of equal size, and iteratively remove the minimum out-degree vertex from S and the minimum in-degree vertex from T, until all vertices are removed. Throughout this process, we keep track of the minimum mean-based density of each intermediate (S, T)-induced subgraph and select the highest value as our final result.

The pseudo-code for M-Approx is presented in Algorithm 3. In this algorithm, S^* and T^* are used to store the current optimal solution found, and ρ^* records the corresponding density. Firstly, it initializes both S and T as V (Line 1) and regards the entire graph as the current optimal solution (Line 2). Then, it removes from S the vertex u that has the minimum out-degree and from T the vertex v that has the minimum in-degree (Lines 4-6). After removing u and v, it recomputes the minimum mean-based density of the remained (S, T)-induced subgraph and updates ρ^*, S^*, and T^* if necessary (Lines 7-8). It repeats

Algorithm 3: M-Approx

```
Input: A directed graph $G = (V, E)$
Output: A subgraph $G[S, T]$ with high minimum mean-based density
1 $S \leftarrow V$, $T \leftarrow V$;
2 $S^* \leftarrow S$, $T^* \leftarrow T$, $\rho^* \leftarrow \rho_m(S, T)$;
3 while $S \neq \emptyset$ and $T \neq \emptyset$ do
4     $u \leftarrow \arg\min_{u \in S} d^+_{G[S,T]}(u)$;
5     $v \leftarrow \arg\min_{v \in T} d^-_{G[S,T]}(v)$;
6     $S \leftarrow S \setminus \{u\}$, $T \leftarrow T \setminus \{v\}$;
7     if $\rho_m(S, T) > \rho^*$ then
8         $S^* \leftarrow S$, $T^* \leftarrow T$, $\rho^* \leftarrow \rho_m(S, T)$ ;
9 return $G[S^*, T^*]$;
```

the above step until S and T become empty. Finally, the algorithm returns the intermediate subgraph $G[S^*, T^*]$ with the highest density, as the desired approximation result (Line 9).

Theorem 2. *The time complexity of* M-Approx *(Algorithm 3) is* $\mathcal{O}(n + m)$.

6 Experiments

In this section, we empirically evaluate the efficiency and effectiveness of our algorithms. We compare the following algorithms:

- H-Approx: Our approximation algorithm for the HDDS problem.
- A-Approx: Our approximation algorithm for the ADDS problem.
- M-Approx: Our approximation algorithm for the MDDS problem.
- Core-Approx: The approximation algorithm proposed in [6] for the GDDS problem.

All algorithms are implemented in C++ and compiled with g++ 7.5.0 with the -O3 flag.[1] All experiments are conducted on a machine with an Intel Core-i7 3.20GHz CPU and 64GB RAM running Ubuntu 18.04. The time cost is measured as the amount of wall-clock time elapsed during the program's execution.

Datasets. We evaluate our algorithms on 10 real-world directed graphs, which are publicly available on Konect[2]. The main characteristics of the tested graphs are summarized in Table 1, where d^+_{max}, d^-_{max} and γ represent the maximum out-degree, maximum in-degree and the maximum x value among all possible (x, x)-cores in the graph, respectively.

6.1 Effectiveness Evaluation of Different Models

In this experimental study, we analyze the sizes of sets S and T yielded by each algorithm in Table 2. Our observations reveal that, except on Enron and Baidu,

[1] Our source codes are available at https://github.com/kyaocs/DDS

[2] konect.cc

Table 1. Statistics of datasets

Abbreviation	Dataset	$\lvert V\rvert$	$\lvert E\rvert$	d^+_{max}	d^-_{max}	γ	Category
TW	Twitter	$23,370$	$33,101$	238	57	10	Social
AD	Advogato	$5,155$	$47,135$	785	721	18	Social
EN	Enron	$86,978$	$320,154$	$1,566$	$1,338$	49	Communication
EP	Epinions	$75,879$	$508,837$	$1,801$	$3,035$	50	Social
AU	AskUbuntu	$157,222$	$544,621$	$4,965$	$1,953$	34	Communication
AM	Amazon	$403,394$	$3,387,388$	10	$2,751$	10	Trade
YT	YouTube	$1,138,494$	$4,942,297$	$28,564$	$25,487$	49	Social
WT	wikiTalk	$2,394,385$	$5,021,410$	$100,022$	$3,311$	96	Communication
BA	Baidu	$2,140,198$	$17,632,190$	$2,477$	$97,848$	59	Hyperlink
DB	DBpedia	$18,268,991$	$136,537,566$	$8,105$	$612,308$	132	Hyperlink

both A-Approx and Core-Approx consistently generate identical outcomes across various datasets, and their outcomes frequently manifest as imbalanced in terms of the sizes of S and T. For instance, on DBpedia graph, both A-Approx and Core-Approx identify the $(1, 612308)$-core, i.e., $|S| = 612308$ and $|T| = 1$. This outcome can be attributed to the prevalence of the power-law degree distribution characteristic in real-world graphs. Consequently, a small number of vertices tend to exhibit exceptionally high out-degrees or in-degrees, leading to pronounced imbalances in the outcomes. Such skewed outcomes severely curtail the meaningful insights attainable through directed graph analysis. Interestingly, H-Approx exhibits superior equilibrium between the sizes of S and T. For instance, when applied to the DBpedia graph, H-Approx identifies a subgraph with $|S| = 1225$ and $|T| = 1298$. The outcomes produced by M-Approx invariably maintain an equitable balance between the sizes of S and T due to the inherent strict criteria in the algorithm's design.

Table 2. The sizes of S and T returned by each algorithm

	TW	AD	EN	EP	AU
H-Approx	$(16, 29)$	$(401, 386)$	$(626, 813)$	$(796, 631)$	$(287, 353)$
A-Approx	$(1, 238)$	$(1, 785)$	$(1, 1566)$	$(3035, 1)$	$(1, 4965)$
M-Approx	$(17, 17)$	$(387, 387)$	$(679, 679)$	$(697, 697)$	$(322, 322)$
Core-Approx	$(1, 238)$	$(1, 785)$	$(17, 158)$	$(3035, 1)$	$(1, 4965)$
	AM	YT	WT	BA	DB
H-Approx	$(243939, 143922)$	$(1875, 1937)$	$(1179, 1222)$	$(7170, 4059)$	$(1225, 1298)$
A-Approx	$(2751, 1)$	$(1, 28564)$	$(1, 100022)$	$(97848, 1)$	$(612308, 1)$
M-Approx	$(429, 429)$	$(1886, 1886)$	$(1197, 1197)$	$(5270, 5270)$	$(1248, 1248)$
Core-Approx	$(2751, 1)$	$(1, 28564)$	$(1, 100022)$	$(95762, 2)$	$(612308, 1)$

To get a deeper insight into different algorithms and measures, we evaluate the various densities associated with outcomes produced by different algorithms. Table 3 presents the outcomes for the Epinions dataset. Notably, the densities of outcomes from H-Approx and M-Approx closely align (as evident in the second and fourth rows), which can be attributed to the similar or identical sizes of S and T derived by these algorithms. Conversely, the outcomes generated by A-Approx and Core-Approx showcase considerable density fluctuations across different measures. This fluctuation results from the imbalances in the sizes of S and T. For instance, the outcome produced by A-Approx displays an arithmetic mean-based density of 1518, whereas its harmonic mean-based density, minimum mean-based density, and geometric mean-based density are only 1.99, 1, and 55.09, respectively (as shown in the third row). A noteworthy observation is the higher geometric mean-based density exhibited by outcomes from H-Approx and M-Approx compared to those from Core-Approx. This discrepancy underscores the drawbacks stemming from imbalanced S and T distributions. We also conduct the comparison on YouTube, as shown in Table 4, where a similar phenomenon can be observed.

Table 3. Cross-comparison between different algorithms and measures on Epinions

	$(\lvert S\rvert, \lvert T\rvert, \lvert E\rvert)$	ρ_h	ρ_a	ρ_m	ρ_g
H-Approx	$(796, 631, 63718)$	89.30	90.51	80.05	89.91
A-Approx	$(3035, 1, 3035)$	1.99	1518	1	55.09
M-Approx	$(697, 697, 61724)$	88.56	88.56	88.56	88.56
Core-Approx	$(3035, 1, 3035)$	1.99	1518	1	55.09

Table 4. Cross-comparison between different algorithms and measures on YouTube

	$(\lvert S\rvert, \lvert T\rvert, \lvert E\rvert)$	ρ_h	ρ_a	ρ_m	ρ_g
H-Approx	$(1875, 1937, 168333)$	88.32	88.34	86.90	88.33
A-Approx	$(1, 28564, 28564)$	1.99	14282.5	1	169.01
M-Approx	$(1886, 1886, 166543)$	88.30	88.30	88.30	88.30
Core-Approx	$(1, 28564, 28564)$	1.99	14282.5	1	169.01

6.2 Efficiency Evaluation of Different Algorithms

In this experiment, we evaluate the efficiency of different algorithms on all datasets. As shown in Figure 2, the running time of H-Approx and M-Approx are almost the same with each other on all datasets. For example, on YouTube, their running times are 0.45 seconds and 0.42 seconds, respectively. This alignment is congruent with their underlying theoretical time complexities, both characterized by $\mathcal{O}(n + m)$. A-Approx and Core-Approx are around one or two orders of magnitude slower than H-Approx and M-Approx. For example, on YouTube, the running times of A-Approx and Core-Approx are 5.97 seconds and 17 seconds, respectively. This is because A-Approx (resp. Core-Approx) needs to find the (x, y)-core that maximizes $x + y$ (resp. $x \times y$), which leads to the theoretical time complexity of $\mathcal{O}(\sqrt{m}(n + m))$.

7 Related Work

Densest subgraph detection on undirected graphs. Goldberg [12] laid the groundwork for the problem of identifying the densest subgraph in undirected

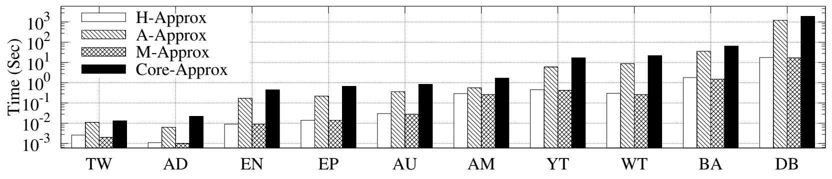


Fig. 2. Running time on all datasets

graphs. This problem seeks the subgraph with the highest edge-density among all subgraphs, where the edge-density of a graph $G = (V, E)$ is defined as $\frac{|E|}{|V|}$. Goldberg proposed an exact algorithm based on the max-flow technique. In [10], Charikar proposed a 2-approximation algorithm which takes linear time cost with regard to the graph size. Some variants of the densest subgraph detection problem are also studied, such as the k-clique densest subgraph detection [13–15], the locally densest subgraph detection [16, 17], the size-bounded densest subgraph detection [18], and the (p, q)-biclique densest subgraph detection [19]. However, these approaches primarily address undirected graphs and are not directly applicable to the distinct problem formulations tackled in this study.

Densest subgraph detection on directed graphs. The directed version of the densest subgraph problem was introduced by Kannan and Vinay [2]. In [10], an exact algorithm was proposed by solving $\mathcal{O}(n^2)$ linear programmings. Later, another exact algorithm was proposed in [3] by extending the max-flow based technique. However, these exact algorithms are time-consuming and are not suitable for larger directed graphs. In [2], Kannan and Vinay proposed an $\mathcal{O}(\log n)$-approximation algorithm to compute the directed densest subgraph. In [10], Charikar designed a 2-approximation algorithm with a time complexity of $\mathcal{O}(n^2(n+m))$. Bahmani et al. [8] proposed a $2(1+\epsilon)$-approximation algorithm in the streaming model ($\epsilon > 0$). Recently, Ma et al. [6] proposed an 2-approximation algorithm with time complexity of $\mathcal{O}(\sqrt{m}(n + m))$. However, these algorithms are tailored to address the geometric mean-based directed densest subgraph problem, and their adaptation to the novel problem formulations addressed in this paper remains unexplored.

8 Conclusion

In this paper, we complemented the existing studies of directed densest subgraph identification by introducing three alternative density measures to capture the density between two vertex subsets S and T of a directed graph. Based on these density measures, we formulated three distinct directed densest subgraph problems, and proposed corresponding approximation/heuristic algorithms. Extensive experiments on real-world directed graphs demonstrated the effectiveness and efficiency of our techniques. One interesting direction of future work is to design exact algorithms for the formulated directed densest subgraph problems.

Acknowledgements. This work was partially supported by the Australian Research Council Funding of DP220103731.

References

1. Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics reports*, 533(4):95–142, 2013.
2. Ravindran Kannan and V Vinay. *Analyzing the structure of large graphs*. Universität Bonn. Institut für Ökonometrie und Operations Research, 1999.
3. Samir Khuller and Barna Saha. On finding dense subgraphs. In *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 597–608, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
4. Jaewon Yang, Julian McAuley, and Jure Leskovec. Detecting cohesive and 2-mode communities indirected and undirected networks. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 323–332, 2014.
5. Saurabh Sawlani and Junxing Wang. Near-optimal fully dynamic densest subgraph. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 181–193, 2020.
6. Chenhao Ma, Yixiang Fang, Reynold Cheng, Laks V.S. Lakshmanan, Wenjie Zhang, and Xuemin Lin. Efficient algorithms for densest subgraph discovery on large directed graphs. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 1051–1066, New York, NY, USA, 2020. Association for Computing Machinery.
7. Gregory Buehrer and Kumar Chellapilla. A scalable pattern mining approach to web graph compression with communities. In *Proceedings of the 2008 international conference on web search and data mining*, pages 95–106, 2008.
8. Bahman Bahmani, Ravi Kumar, and Sergei Vassilvitskii. Densest subgraph in streaming and mapreduce. *Proceedings of the VLDB Endowment*, 5(5), 2012.
9. Aristides Gionis and Charalampos E. Tsourakakis. Dense subgraph discovery: Kdd 2015 tutorial. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 2313–2314, New York, NY, USA, 2015. Association for Computing Machinery.
10. Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, APPROX '00, page 84–95, Berlin, Heidelberg, 2000. Springer-Verlag.
11. Vladimir Batagelj and Matjaz Zaversnik. An o (m) algorithm for cores decomposition of networks. *arXiv preprint cs/0310049*, 2003.
12. Andrew V Goldberg. Finding a maximum density subgraph. 1984.
13. Yixiang Fang, Kaiqiang Yu, Reynold Cheng, Laks VS Lakshmanan, and Xuemin Lin. Efficient algorithms for densest subgraph discovery. *Proceedings of the VLDB Endowment*, 12(11):1719–1732, 2019.
14. Charalampos Tsourakakis. The k-clique densest subgraph problem. In *Proceedings of the 24th international conference on world wide web*, pages 1122–1132, 2015.
15. Bintao Sun, Maximilien Danisch, TH Hubert Chan, and Mauro Sozio. Kclist++: A simple algorithm for finding k-clique densest subgraphs in large graphs. *Proceedings of the VLDB Endowment (PVLDB)*, 2020.
16. Lu Qin, Rong-Hua Li, Lijun Chang, and Chengqi Zhang. Locally densest subgraph discovery. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 965–974, 2015.
17. Tran Ba Trung, Lijun Chang, Nguyen Tien Long, Kai Yao, and Huynh Thi Thanh Binh. Verification-free approaches to efficient locally densest subgraph discovery. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1–13. IEEE, 2023.

18. Reid Andersen and Kumar Chellapilla. Finding dense subgraphs with size bounds. In *International workshop on algorithms and models for the web-graph*, pages 25–37. Springer, 2009.
19. Michael Mitzenmacher, Jakub Pachocki, Richard Peng, Charalampos Tsourakakis, and Shen Chen Xu. Scalable large near-clique detection in large-scale networks via sampling. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 815–824, 2015.