Densest Subgraph Discovery on Large Graphs: Applications, Challenges, and Techniques

Yixiang Fang
School of Data Science,
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
fangyixiang@cuhk.edu.cn

Wensheng Luo
School of Data Science,
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
luowensheng@cuhk.edu.cn

Chenhao Ma
Department of Computer Science,
The University of Hong Kong
Hong Kong, China
chma2@cs.hku.hk

ABSTRACT

As one of the most fundamental problems in graph data mining, the *densest subgraph discovery* (DSD) problem has found a broad spectrum of real applications, such as social network community detection, graph index construction, regulatory motif discovery in DNA, fake follower detection, and so on. Theoretically, DSD closely relates to other fundamental graph problems, such as network flow and bipartite matching. Triggered by these applications and connections, DSD has garnered much attention from the database, data mining, theory, and network communities.

In this tutorial, we first highlight the importance of DSD in various applications and the unique challenges that need to be addressed. Subsequently, we classify existing DSD solutions into several groups, which cover around 50 research papers published in many well-known venues (e.g., SIGMOD, PVLDB, TODS, WWW), and conduct a thorough review of these solutions in each group. Afterwards, we analyze and compare the models and solutions in these works. Finally, we point out a list of promising future research directions. We believe that this tutorial not only helps researchers have a better understanding of existing densest subgraph models and solutions, but also provides them insights for future study.

PVLDB Reference Format:
Yixiang Fang, Wensheng Luo, and Chenhao Ma. Densest Subgraph Discovery on Large Graphs: Applications, Challenges, and Techniques. PVLDB, 15(12): 3766-3769, 2022.
doi:10.14778/3554821.3554895

1 INTRODUCTION

As one of the most fundamental problems in graph data mining, the *densest subgraph discovery* (DSD) problem aims to discover a very "dense" subgraph from a given graph. More precisely, given an undirected graph, the original DSD problem [34] finds a subgraph with the highest *edge-density*, which is defined as the number of edges over the number of vertices in the subgraph, and it is often termed as densest subgraph (DS). This problem was also studied on directed graphs [39] by extending the above edge-density for considering the edge directions. The DSD problem lies in the core of graph mining [33], and is widely used in network science [3, 35], biological analysis [27], graph databases [19, 38], system optimization [31–33], and graph compression [12]. Besides, the DSD problem is also closely related to other fundamental graph problems, such as network flow and bipartite matching [54]. Due to the theoretical and practical importance, researchers from the database, data mining, computer science theory, and network communities designed efficient and effective solutions to the DSD problem.

Despite the wide applications of DSD, the DSD problem is a very challenging task because: (1) the exact DSD solutions (e.g., [25, 34]) often involve the computation of maximum flow, which has a very high time complexity; and (2) many real-world graphs are often with huge sizes (e.g., Facebook user network has more than 2.89 billion monthly active users as of October 2021[1]). Thus, the first key challenge is to develop efficient algorithms. To improve the efficiency of DSD, researchers have tried many different techniques as shown in the literature, which are summarized as follows: (1) following the prune-and-verify framework (e.g., [25, 44]) to locate the DS; (2) proposing approximate algorithms with theoretical guarantees that sacrifice some accuracy for achieving higher efficiency (e.g., [15]); and (3) developing distributed algorithms to compute in a parallel manner (e.g., [5]).

Besides, many real networks are not just undirected or directed graphs, and a real application (e.g., community detection) often needs not just one single DS, while the original DSD problem studied on undirected and directed graphs is only able to return one single DS. Therefore, the second key challenge is how to perform effective DSD such that it can well satisfy different requirements on different graphs. To this end, some researchers have attempted to extend the original DSD problem and solutions for bipartite graphs (e.g., [1]), multilayer graphs (e.g., [37]), uncertain graphs (e.g., [48]). Meanwhile, many variants of the DSD have been studied to satisfy different practical requirements, such as clique-density-based DSD (e.g., [57]), pattern-density-based DSD (e.g., [25]), densest k-subgraph [4], density-friendly graph decomposition [56], locally DSD (e.g., [51]), DS deconstruction (e.g., [14]), etc.

In summary, there are many existing works focusing on different aspects of the DSD topic. Nevertheless, there is a lack of systematic review and comparison study among them, except for two very preliminary works [18, 26] which briefly reviewed the works in the general area of dense subgraph computation, with a little attention to the topic of DSD. To this end, in this tutorial, we aim to provide a comprehensive review of these *densest* subgraph discovery works, which directly use the *edge-density* definition, or density definitions extended from it. In other words, other dense subgraph models [24], such as k-core, k-truss, k-clique, k-edge connected component, and so on, will not be covered by this tutorial.

In the tutorial, we will first offer an introduction to the research field to highlight the popularity and applications of DSD, and also

Proceedings of the VLDB Endowment, Vol. 15, No. 12 ISSN 2150-8097.
doi:10.14778/3554821.3554895

[1] https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

Table 1: Classification of existing DSD works.

Graph type	Original DSD problem		Variants of original DSD problem
	Exact solutions	Approx. solutions	
Undirected graphs	Unweighted case [15, 25, 34] Weighted case [20]	2-approximation [10, 15, 25] 2(1+ϵ)-approximation [5] $(1+\epsilon)$-approximation [16, 54] DS maintenance [5, 8, 23, 36, 54]	Clique-density-based DSD [25, 47, 53–55, 57] Pattern-density-based DSD [25] Densest k-subgraph [4, 7, 9, 11, 16, 50] Size-bounded DSD [2] Top-k overlapping DSD [21, 22, 28] Maximum total density DSD [6] Density-friendly graph decomposition [20, 56] Locally DSD [42, 51] DS deconstruction [14] Top-k DSD maintenance [49]
Directed graphs	Unweighted case [15, 39, 40, 44, 45] Weighted case [45]	$O(\log n)$-approximation[39] 2-approximation[15, 44, 45] 2(1+ϵ)-approximation[5] $(1+\epsilon)$-approximation [43, 54] DS maintenance [5, 45, 54]	Densest at least k_1, k_2 subgraph [40]
Others	Uncertain graphs [60]	Bipartite graphs [1, 35, 47] Multilayer graphs [29, 30, 37] Uncertain graphs [48]	Dense connected subgraphs [58]

⋆ The "original DSD problem" means that given an undirected/directed graph, return the subgraph with the largest corresponding edge-density.
⋆ Note: n is the number of vertices in the graph; k, k_1, and k_2 are integers; $\epsilon > 0$ is a real value; the approximation ratio is defined as the ratio of the density of the DS over that of the subgraph returned.

discuss the key challenges of DSD. Subsequently, we classify existing DSD solutions according to the problem definitions and solutions, as shown in Table 1, which covers around 50 research papers published in many well-known venues (e.g., SIGMOD, PVLDB, and TODS), and then review the representative solutions thoroughly. We will also analyze and compare the specific problems and solutions on undirected graphs, directed graphs, and other graphs, respectively. Finally, we will offer a list of promising future research directions of DSD, which can provide researchers with some good starting points to work in this research area.

2 TUTORIAL OUTLINE

This tutorial is tailored for typical VLDB attendees who are expected to be aware of the broad area of databases but may or may not be actively working on graphs and networks. This will be a 1.5 hours tutorial. Below is the outline of the tutorial.

2.1 Introduction, Applications, and Challenges (20 minutes)

This part aims to provide the necessary background to the audience. It will consist of an introduction to the research field to highlight the popularity and applications of densest subgraph discovery. We will first introduce the definitions of graph density [25, 34, 39, 40, 44, 45, 53, 57], discuss the interesting applications of DSD over real big graphs, and finally present the key challenges of DSD.

Specifically, we will first review the DSD definitions on different kinds of graphs, and then highlight a wide spectrum of specific real applications that can benefit from the solutions of DSD in various areas, such as network science [17, 35], biological analysis [27, 52], graph databases [19, 38, 59], system optimization [31–33], and fraud detection [35].

Subsequently, we will present the two key challenges of DSD: (1) how to develop efficient DSD solutions? and (2) how to perform effective DSD such that it can well satisfy different requirements on different graphs. The first challenge stems from the fact that the exact solutions of DSD problems are often with high time complexity, which poses a great challenge for achieving higher efficiency with a good approximation. To facilitate efficient densest subgraph discovery, the following techniques are often adopted in the literature: (1) following the prune-and-verify framework (e.g., [25, 44, 45]) to locate the densest subgraph in some small-scale cohesive subgraphs; (2) proposing approximation algorithms (e.g., [5, 15, 25, 44, 45, 55]) that sacrifice some accuracy for achieving higher efficiency; and (3) developing distributed algorithms to search in a parallel manner. In addition, building index structures or doing some pre-processing (e.g., [14]) can also accelerate the DSD process. For the second challenge, researchers often introduce new variants of DSD and solutions, by borrowing insights from the original DSD problem and also considering the characteristics of these graphs (e.g., multiple layers of the graph) with application requirements (e.g., threshold on the number of vertices).

2.2 DSD on Undirected Graphs (25 minutes)

This part mainly reviews the DSD solutions on undirected graphs:
• **Exact solutions for the original DSD problem.** We will first introduce the linear programming form of the DSD problem [34] and the flow network model [41]. Subsequently, we will present the exact solutions based on min-cut of flow-networks [25, 34] and linear programming [15, 34]. Finally, we will discuss the time complexity of the exact algorithms and the pruning strategies [25] to improve efficiency.
• **Approximation solutions for the original DSD problem.** We will first introduce the concepts of the approximation algorithm and the approximation ratio. Then the solutions with different approximation ratios will be extensively reviewed, including the $(2(1+\epsilon)$-approximation algorithm [5], 2-approximation algorithms [15, 25], and $(1+\epsilon)$-approximation algorithm [54]). Afterwards, we will discuss how to maintain the DS on dynamic graphs. Finally, we will compare these solutions in terms of accuracy and efficiency.
• **Variants of the original DSD problem.** Table 1 shows various variants of DSD on undirected graphs. We will first introduce clique-density-based DS [25, 47, 53–55, 57], where the clique-density was extended from edge-density and triangle-density [53] is a special case of clique-density. We will also show that by replacing a k-cliques with a arbitrary pattern graph, we can obtain the pattern-density [25]. Next, we will extensively discuss the algorithms for computing the DS's based on clique-density and pattern-density.

After that, we will introduce the DSD problem with size bound, including densest k-subgraph [4, 7, 9, 11, 13, 46, 50] with k vertices and size-bounded DS [2] with at least k vertices. Meanwhile, we will show the approximation solutions for these problems. Afterward, DS's with overlapping constraints will be introduced, such as top-k overlapping DS [21, 22, 28] and maximum summary density DS limited overlap [6]. Finally, other variants and corresponding solutions, including top-k DSD and its maintenance [49], locally DSD [51], density-friendly graph decomposition [20, 56], DS deconstruction [14] will also be briefly introduced.

In addition, we will extensively analyze and compare these DSD solutions from different angles, such as time complexity, space complexity, input parameters, approximation ratio, and application scenarios. Note that for ease of illustration, some toy examples will be used when introducing the problems and solutions.

2.3 DSD on Directed Graphs (10 minutes)

This part mainly reviews the DSD solutions on directed graphs:
• **Exact solutions for the original DSD problem.** We will first introduce the definition of graph density on directed graphs. Then we will present solutions of exact algorithms of the DSD problem on directed graphs [15, 39, 40, 44, 45]. Finally, we will compare the exact algorithms of the original DSD problems on undirected and directed graphs in terms of the main ideas and algorithm complexities.
• **Approximation solutions for the original DSD problem.** We will first present the solutions with different approximation ratios for the DSD problem on directed graphs, including the $\log n$-approximation algorithm [39], $2(1+\epsilon)$-approximation algorithm [5], 2-approximation algorithms [15, 44, 45], and $(1+\epsilon)$-approximation algorithms [43, 54]. After that, we will introduce the maintenance algorithm of the DS on dynamic directed graphs [5, 45, 54].

In addition, a variant of the DSD problem on directed graphs [40] will be introduced. Besides, we will also comprehensively analyze and compare the DSD solutions on directed graphs.

2.4 DSD on Other Graphs (20 minutes)

This part mainly reviews the works of DSD variants on bipartite graphs, multilayer graphs, uncertain graphs, and dual networks.
• **DSD on bipartite graphs.** We will first introduce the definition of bipartite graphs and its density. Based on these concepts, we will present the DSD problem on bipartite graphs [1], and a more general case called (p,q)-biclique-based DS [35, 47]. Afterwards, we will discuss both exact and approximation DSD solutions.
• **DSD on multilayer graphs.** We will first introduce the definition of multilayer graphs. Then we will introduce the density definition on multi-layer graphs, namely common density [37], which is extended from edge-density, and multi-layer density [29, 30], which is an improved version of the common density. After that, we will present the algorithms to solve these two problems respectively.
• **DSD on uncertain graphs.** We will first introduce the definition of uncertain graphs. Subsequently, we will introduce the density definitions on uncertain graphs, namely expected density [60] and robust density [48]. Finally, we will review both exact and approximation solutions to the DSD problem on uncertain graphs.

2.5 Future Research Directions (15 minutes)

We will discuss a list of promising future research directions on the topic of DSD. Here, we just show three directions for lack of space.
• **DSD on heterogeneous graphs.** As summarized in Table 1, the original DSD problems on undirected and directed graphs have been extended for bipartite graphs, multilayer graphs, uncertain graphs, and dual graphs, which actually can be considered as special cases of the heterogeneous graph that often involves vertices and edges with multiple types. The heterogeneous graphs are prevalent in various domains such as knowledge graphs, bibliographic networks, and biological networks. Thus, a promising future research direction is to derive a unified density definition for a general heterogeneous graph, such that the density definitions for the above special graphs are its special cases. To do this, we may re-define the density by using some well-known concepts on heterogeneous graphs like motif. Afterwards, the corresponding DSD problem on heterogeneous graphs may be solved by extending the existing solutions.
• **Efficient DSD algorithms.** Here are some research directions:

(1) *Parallel algorithms.* Parallel algorithms (e.g., [5]) usually use distributed computing platforms or multi-core computing resources to accelerate computation. Thus, an interesting future research direction is to study the parallel exact algorithms for the DSD problem.
(2) *Fast approximation algorithms.* Although there are some approximation solutions to the DSD problem, they may still suffer from the low efficiency issue, since real-world graphs are often with huge sizes, calling for faster approximation algorithms with better balance between the quality of results and computational efficiency.

• **Application-driven variants of DSD.** As aforementioned, there are many variants of the DSD problem, but most of them were not customized for some specific application scenarios. Consequently, an interesting future research direction is to study the application-driven variants of the DSD problem, by carefully considering the requirements of real-life scenarios. For example, the DSD solutions can be used for detecting network communities [17]. However, in a geo-social network, a community often contains a group of users that are not only linked densely, but also have close physical distance. Thus, it would be interesting to study how to incorporate the distance into the DSD problem.

In summary, after the tutorial, attendees will be familiar with:

(1) The typical applications and key challenges of DSD;
(2) The representative DSD solutions on undirected graphs, directed graphs, and other graphs;
(3) The representative variants of DSD problem and solutions over different kinds of graphs;
(4) A list of promising future research directions on DSD.

3 PRESENTERS

Yixiang Fang is an Associate Professor at the School of Data Science, The Chinese University of Hong Kong, Shenzhen. His research interests mainly focus on the areas of data management, data mining, and artificial intelligence over big data. Most of his research works were published in top-tier conferences (e.g., PVLDB, SIGMOD, ICDE, and NeurIPS) and journals (e.g., TODS, VLDBJ, and TKDE). He received the 2021 ACM SIGMOD Research Highlight Award. He is an editorial board member of the journal of Information & Processing Management (IPM). He has also served as PC members for several top conferences (e.g., PVLDB, ICDE, and KDD) and reviewers for top journals (e.g., TKDE and VLDBJ).
Wensheng Luo is a Postdoctoral Fellow at the School of Data Science, The Chinese University of Hong Kong, Shenzhen. His research interests lie in parallel computing and data management/mining, especially for the graph data. Most of his research works were published in top-tier conferences and journals (e.g., ICDE and TKDE).
Chenhao Ma is a Postdoctoral Fellow in the Department of Computer Science, the University of Hong Kong. His research interests

mainly lie in the areas of graph data management and data mining. His research works were often published in top-tier database conferences and journals (e.g., PVLDB, SIGMOD, and TODS). He received the 2021 ACM SIGMOD Research Highlight Award.

Note: This tutorial has not been offered elsewhere before.

ACKNOWLEDGMENTS

This work was supported by NSFC under Grant 62102341, and Basic and Applied Basic Research Fund in Guangdong Province under Grant 2022A1515010166.

REFERENCES

[1] Reid Andersen. 2010. A local algorithm for finding dense subgraphs. *ACM TALG* 6, 4 (2010), 1–12.
[2] Reid Andersen and Kumar Chellapilla. 2009. Finding dense subgraphs with size bounds. In *WAW*. Springer, 25–37.
[3] Albert Angel, Nick Koudas, Nikos Sarkas, Divesh Srivastava, Michael Svendsen, and Srikanta Tirthapura. 2014. Dense subgraph maintenance under streaming edge weight updates for real-time story identification. *VLDB J.* 23, 2 (2014), 175–199.
[4] Yuichi Asahiro, Kazuo Iwama, Hisao Tamaki, and Takeshi Tokuyama. 2000. Greedily finding a dense subgraph. *Journal of Algorithms* 34, 2 (2000), 203–221.
[5] Bahman Bahmani, Ravi Kumar, and Sergei Vassilvitskii. 2012. Densest Subgraph in Streaming and MapReduce. *PVLDB* 5, 5 (2012).
[6] Oana Denisa Balalau, Francesco Bonchi, TH Hubert Chan, Francesco Gullo, and Mauro Sozio. 2015. Finding subgraphs with maximum total density and limited overlap. In *WSDM*. 379–388.
[7] Aditya Bhaskara, Moses Charikar, Venkatesan Guruswami, Aravindan Vijayaraghavan, and Yuan Zhou. 2012. Polynomial integrality gaps for strong sdp relaxations of densest k-subgraph. In *SODA*. SIAM, 388–405.
[8] Sayan Bhattacharya, Monika Henzinger, Danupon Nanongkai, and Charalampos Tsourakakis. 2015. Space-and time-efficient algorithm for maintaining dense subgraphs on one-pass dynamic streams. In *STOC*. 173–182.
[9] Francesco Bonchi, David García-Soriano, Atsushi Miyauchi, and Charalampos E Tsourakakis. 2021. Finding densest k-connected subgraphs. *Discrete Applied Mathematics* 305 (2021), 34–47.
[10] Digvijay Boob, Yu Gao, Richard Peng, Saurabh Sawlani, Charalampos Tsourakakis, Di Wang, and Junxing Wang. 2020. Flowless: Extracting densest subgraphs without flow computations. In *WWW*.
[11] Nicolas Bourgeois, Aristotelis Giannakos, Giorgio Lucarelli, Ioannis Milis, and Vangelis Th Paschos. 2013. Exact and approximation algorithms for densest k-subgraph. In *WALCOM*. Springer, 114–125.
[12] Gregory Buehrer and Kumar Chellapilla. 2008. A scalable pattern mining approach to web graph compression with communities. In *Proc. WSDM 2008*. 95–106.
[13] Cristian S Calude, Michael J Dinneen, and Richard Hua. 2020. Quantum solutions for densest k-subgraph problems. *Journal of Membrane Computing* 2, 1 (2020), 26–41.
[14] Lijun Chang and Miao Qiao. 2020. Deconstruct Densest Subgraphs. In *WWW*. 2747–2753.
[15] Moses Charikar. 2000. Greedy approximation algorithms for finding dense components in a graph. In *APPROX*. Springer, 84–95.
[16] Chandra Chekuri, Kent Quanrud, and Manuel R Torres. 2022. Densest Subgraph: Supermodularity, Iterative Peeling, and Flow. In *SODA*. SIAM, 1531–1555.
[17] Jie Chen and Yousef Saad. 2010. Dense subgraph extraction with application to community detection. *IEEE TKDE* 24, 7 (2010), 1216–1230.
[18] Avery Ching, Sergey Edunov, Maja Kabiljo, Dionysios Logothetis, and Sambavi Muthukrishnan. 2015. One trillion edges: Graph processing at facebook-scale. *PVLDB* 8, 12 (2015), 1804–1815.
[19] Edith Cohen, Eran Halperin, Haim Kaplan, and Uri Zwick. 2003. Reachability and distance queries via 2-hop labels. *SIAM J. Comput.* 32, 5 (2003), 1338–1355.
[20] Maximilien Danisch, T-H Hubert Chan, and Mauro Sozio. 2017. Large scale density-friendly graph decomposition via convex programming. In *WWW*. 233–242.
[21] Riccardo Dondi, Mohammad Mehdi Hosseinzadeh, and Pietro H Guzzi. 2021. A novel algorithm for finding top-k weighted overlapping densest connected subgraphs in dual networks. *Applied Network Science* 6, 1 (2021), 1–17.
[22] Riccardo Dondi, Mohammad Mehdi Hosseinzadeh, Giancarlo Mauri, and Italo Zoppis. 2021. Top-k overlapping densest subgraphs: approximation algorithms and computational complexity. *J. Comb. Optim.* 41, 1 (2021), 80–104.
[23] Alessandro Epasto, Silvio Lattanzi, and Mauro Sozio. 2015. Efficient densest subgraph computation in evolving graphs. In *WWW*. 300–310.
[24] Yixiang Fang, Xin Huang, Lu Qin, Ying Zhang, Wenjie Zhang, Reynold Cheng, and Xuemin Lin. 2020. A survey of community search over big graphs. *VLDB J.* 29, 1 (2020), 353–392.
[25] Yixiang Fang, Kaiqiang Yu, Reynold Cheng, Laks VS Lakshmanan, and Xuemin Lin. 2019. Efficient algorithms for densest subgraph discovery. *PVLDB* 12, 11 (2019), 1719–1732.
[26] András Faragó and Zohre R Mojaveri. 2019. In search of the densest subgraph. *Algorithms* 12, 8 (2019), 157.
[27] Eugene Fratkin, Brian T Naughton, Douglas L Brutlag, and Serafim Batzoglou. 2006. MotifCut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics* 22, 14 (2006), e150–e157.
[28] Esther Galbrun, Aristides Gionis, and Nikolaj Tatti. 2016. Top-k overlapping densest subgraphs. *Data Min. Knowl. Discov.* 30, 5 (2016), 1134–1165.
[29] Edoardo Galimberti, Francesco Bonchi, and Francesco Gullo. 2017. Core decomposition and densest subgraph in multilayer networks. In *CIKM*. 1807–1816.
[30] Edoardo Galimberti, Francesco Bonchi, Francesco Gullo, and Tommaso Lanciano. 2020. Core decomposition in multilayer networks: theory, algorithms, and applications. *TKDD* 14, 1 (2020), 1–40.
[31] David Gibson, Ravi Kumar, and Andrew Tomkins. 2005. Discovering large dense subgraphs in massive graphs. In *VLDB 2005*. Citeseer, 721–732.
[32] Aristides Gionis, Flavio P Junqueira, Vincent Leroy, Marco Serafini, and Ingmar Weber. 2013. Piggybacking on social networks. In *PVLDB*, Vol. 6. 409–420.
[33] Aristides Gionis and Charalampos E Tsourakakis. 2015. Dense subgraph discovery: Kdd 2015 tutorial. In *SIGKDD*. 2313–2314.
[34] Andrew V Goldberg. 1984. *Finding a maximum density subgraph*. University of California Berkeley.
[35] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. Fraudar: Bounding graph fraud in the face of camouflage. In *SIGKDD*. 895–904.
[36] Shuguang Hu, Xiaowei Wu, and TH Hubert Chan. 2017. Maintaining densest subsets efficiently in evolving hypergraphs. In *CIKM*. 929–938.
[37] Vinay Jethava and Niko Beerenwinkel. 2015. Finding dense subgraphs in relational graphs. In *ECML PKDD*. Springer, 641–654.
[38] Ruoming Jin, Yang Xiang, Ning Ruan, and David Fuhry. 2009. 3-hop: a high-compression indexing scheme for reachability query. In *SIGMOD*. 813–826.
[39] Ravindran Kannan and V Vinay. 1999. *Analyzing the structure of large graphs*. Forschungsinst. für Diskrete Mathematik.
[40] Samir Khuller and Barna Saha. 2009. On finding dense subgraphs. In *ICALP*. Springer, 597–608.
[41] Eugene L Lawler. 2001. *Combinatorial optimization: networks and matroids*. Courier Corporation.
[42] Chenhao Ma, Reynold Cheng, Laks VS Lakshmanan, and Xiaolin Han. 2022. Finding Locally Densest Subgraphs: A Convex Programming Approach. (2022).
[43] Chenhao Ma, Yixiang Fang, Reynold Cheng, Laks VS Lakshmanan, and Xiaolin Han. 2022. A Convex-Programming Approach for Efficient Directed Densest Subgraph Discovery. In *SIGMOD*.
[44] Chenhao Ma, Yixiang Fang, Reynold Cheng, Laks VS Lakshmanan, Wenjie Zhang, and Xuemin Lin. 2020. Efficient algorithms for densest subgraph discovery on large directed graphs. In *SIGMOD*. 1051–1066.
[45] Chenhao Ma, Yixiang Fang, Reynold Cheng, Laks VS Lakshmanan, Wenjie Zhang, and Xuemin Lin. 2021. On Directed Densest Subgraph Discovery. *TODS* 46, 4 (2021), 1–45.
[46] Pasin Manurangsi. 2017. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. In *STOC*. 954–961.
[47] Michael Mitzenmacher, Jakub Pachocki, Richard Peng, Charalampos Tsourakakis, and Shen Chen Xu. 2015. Scalable large near-clique detection in large-scale networks via sampling. In *SIGKDD*. 815–824.
[48] Atsushi Miyauchi and Akiko Takeda. 2018. Robust densest subgraph discovery. In *ICDM*. IEEE, 1188–1193.
[49] Muhammad Anis Uddin Nasir, Aristides Gionis, Gianmarco De Francisci Morales, and Sarunas Girdzijauskas. 2017. Fully dynamic algorithm for top-k densest subgraphs. In *CIKM*. 1817–1826.
[50] Tim Nonner. 2016. PTAS for Densest k-Subgraph in Interval Graphs. *Algorithmica* 74, 1 (2016), 528–539.
[51] Lu Qin, Rong-Hua Li, Lijun Chang, and Chengqi Zhang. 2015. Locally densest subgraph discovery. In *SIGKDD*. 965–974.
[52] Barna Saha, Allison Hoch, Samir Khuller, Louiqa Raschid, and Xiao-Ning Zhang. 2010. Dense subgraphs with restrictions and applications to gene annotation graphs. In *RECOMB*. Springer, 456–472.
[53] Raman Samusevich, Maximilien Danisch, and Mauro Sozio. 2016. Local triangle-densest subgraphs. In *ASONAM*. IEEE, 33–40.
[54] Saurabh Sawlani and Junxing Wang. 2020. Near-optimal fully dynamic densest subgraph. In *STOC*. 181–193.
[55] Bintao Sun, Maximilien Danisch, TH Chan, and Mauro Sozio. 2020. KClist++: A Simple Algorithm for Finding k-Clique Densest Subgraphs in Large Graphs. *PVLDB* (2020).
[56] Nikolaj Tatti and Aristides Gionis. 2015. Density-friendly graph decomposition. In *WWW*. 1089–1099.
[57] Charalampos Tsourakakis. 2015. The k-clique densest subgraph problem. In *WWW*. 1122–1132.
[58] Yubao Wu, Ruoming Jin, Xiaofeng Zhu, and Xiang Zhang. 2015. Finding dense and connected subgraphs in dual networks. In *ICDE*. IEEE, 915–926.
[59] Feng Zhao and Anthony KH Tung. 2012. Large scale cohesive subgraphs discovery for social network visual analysis. *PVLDB* 6, 2 (2012), 85–96.
[60] Zhaonian Zou. 2013. Polynomial-time algorithm for finding densest subgraphs in uncertain graphs. In *Proceedings of MLG Workshop*.