Searching Personalized $k$-wing in Bipartite Graphs
(Extended Abstract)

Aman Abidi†, Lu Chen*, Rui Zhou*, and Chengfei Liu*

† National Institute of Education, NTU, Singapore. *Swinburne University of Technology, Melbourne, Australia.
E-mail: †{amana}@nie.edu.sg, *{luchen, rzhou,cliu}@swin.edu.au

Abstract—Enumerating all the bipartite cohesive subgraphs in a bipartite graph has been studied extensively. However, for some applications, one is interested in finding bipartite cohesive subgraphs containing a specific vertex. In this paper, we study a new query-dependent bipartite cohesive subgraph search problem based on $k$-wing model. To address the problem, we propose two efficient and wing number conserving indexing schemes, EquiWing-Graph and a more compact index, EquiWing-Tree, which is achieved by using our proposed $k$-butterfly loose approach and discovered hierarchy properties. Moreover, we discover novel properties that help us localize the scope of the maintenance in our proposed indices at a lower cost for evolving bipartite graphs. Extensive experimental results evidence the efficiency and effectiveness of our proposed approaches.

I. INTRODUCTION

For a given bipartite graph $G = (U, V, E)$, a bipartite cohesive subgraph $B = (U', V', E') \subseteq G$ is defined such that $U'$ and $V'$ are strongly connected via edges in $E'$. Finding bipartite cohesive subgraphs has rich applications such as spam group detection in the web and sponsored advertisement.

Existing models. To cater for the applications discussed above, various bipartite cohesive subgraph models have been studied such as $(\alpha, \beta)$-core, $k$-bitruss, $k$-wing, etc. A $k$-wing subgraph is preferred over $(\alpha, \beta)$-core and $k$-bitruss, as it contains both butterfly (basic building block in a bipartite graph) structure and butterfly connectivity constraint ensuring more cohesiveness. The rationale is that a butterfly represents the strong and stable relationship between two pairs of vertices. Fig. 1 represents an author-paper network from the domain of Computer Science. The 3-bitruss can only discover a single bipartite cohesive subgraph, DB. On the contrary, a $k$-wing model strengthens via butterfly connectivity leading to more strongly (butterfly) connected bipartite subgraphs, i.e., GDM (3-wing) and SDM (3-wing).

Personalized bipartite cohesive subgraph. Distinct vertices in a bipartite graph may have distinct properties, which requires microscopic analysis, i.e., personalized. In Fig. 1, we demonstrate how personalized bipartite cohesive subgraph using $k$-wing model can help authors, for instance $v_8$ and $v_5$, discover their potential future collaborators in Fig. 1, i.e., within their respective $3$-wings. In contrast, if we omit the personalization, the search for authors $v_8$ and $v_5$ would get the same result consisting of all the $3$-wings in Fig. 1. Some of the applications for a personalized bipartite cohesive subgraph search problem includes finding exclusive users for crowdfunding the user’s project [3], and fake review detection

Fig. 1: k-bitruss, k-wing and personalized k-wing

[4]. This problem can be addressed by adapting the algorithm proposed for finding all the $k$-wings, which serves as our baseline. However, the baseline is not efficient due to unawareness of the searching scope. Moreover, $k$-wing maintenance in dynamic bipartite graphs is yet to be explored.

In this paper, we propose the personalized $k$-wing search problem and study the $k$-wing equivalence relationship to summarize the edges of a bipartite graph $G$ into non intersecting groups, i.e. $k$-wing equivalence class. Therefore, forming an efficient and wing number conserving index called EquiWing-Graph (EW-G). Further, we propose a more compact index, EquiWing-Tree (EW-T), which is achieved by using our proposed $k$-butterfly loose approach and discovered hierarchy properties. These indices are used to expedite the personalized $k$-wing search with a non-repetitive access to $G$, which leads to linear algorithms for searching the personalized $k$-wing. Moreover, we discover novel properties to propose algorithms for maintaining the two indices, which substantially reduces the cost of maintenance.

II. BACKGROUND AND BASELINE APPROACH

We first introduce some terminologies and then provide the problem statement. Butterfly (9), given a bipartite graph $G = (U, V, E)$ and four vertices $u, w \in U, v, x \in V$, a butterfly induced by $u, v, w, x$ is a cycle of length of $4$ consisting of edges $(u, v), (w, v), (w, x)$ and $(u, x) \in E$. Butterfly connectivity, given two butterflies $\mathbf{M}_1$ and $\mathbf{M}_2$ in $G$, $\mathbf{M}_1$ and $\mathbf{M}_2$ are butterfly connected if there exists a series of butterflies $\mathbf{M}_1, \ldots, \mathbf{M}_n$ in $G$, in which $n \geq 2$ such that $\mathbf{M}_i = \mathbf{M}_i$, $\mathbf{M}_i = \mathbf{M}_i$, and for $1 \leq i < n$, $\mathbf{M}_i$ and $\mathbf{M}_{i+1}$ share a common edge.

$k$-wing. A bipartite subgraph $H = (U, V, E) \subseteq G$ is a $k$-wing if: (i) $V \in E(H)$, $e$ participates in at least $k$-butterflies. (ii) All pair of edges in $H$ are butterfly connected. (iii) $H$ is maximal.

Wing number, for an edge $e \in E$, it is the maximum possible $k$ such that there exists a $k$-wing in $G$ containing $e$.

Problem Definition. Given a bipartite graph $G = (U, V, E)$, a query vertex $q \in U \cup V$ and an integer $k \geq 1$, we return all $k$-wings containing $q$. 
Baseline Approach. The problem of personalized k-wing search can be addressed by firstly computing the k-bitrusses using the state-of-the-art bitruss algorithm [5]. Then, among the k-bitrusses, we further explore within the k-bitruss containing q to form the personalized k-wings by grouping edges in the k-bitruss containing q together if the edges are butterfly connected. Broadly speaking, it has the following two unnecessary operations. (i) Overhead of accessing ineligible edges. (ii) Redundant access of eligible edges.

III. SUPER GRAPH BASED INDEX

To speed up the personalized k-wing search for real-world applications which may have an extensive number of queries, we first present a novel index, called EquiWing-Graph (EW-G), based on k-wing equivalence adapted from truss equivalence [2]. Using the proposed index, a personalized k-wing query can be processed in linear time w.r.t. the size of the result. The core idea of the index is the introduction of the k-wing equivalent edge sets of a bipartite graph. For a given bipartite graph G, two edges are k-butterfly reachable, if they participate in the same butterfly or are connected by a series of dense butterflies. We group the edges into different equivalent edge sets if edges are k-butterfly reachable and show that edges in the same group have equivalent relationship. EW-G is a super graph based index which is composed of super nodes containing groups of edges having an equivalent relationship.

IV. SUPER TREE BASED INDEX

EW-G is an efficient index, but, there are few limitations. Firstly, unnecessary segregation of super nodes which always occur together in a k-wing. Secondly, the unexplored hierarchical properties of k-wing can be used to reduce the index size. Hence, to address the two drawbacks, we propose the notion of k-butterfly loose connectivity and exploit the hierarchical property among k-wings to form a more compact index with a tree structure. We further study the properties for merging super nodes and aligning the merged super nodes hierarchically, which immediately leads to a novel tree based structure, named as EquiWing-Tree (EW-T). Compared to EW-G, EW-T takes less space and is faster for processing personalized k-wing queries thanks for the compact tree structure.

V. DYNAMIC MAINTENANCE OF INDICES

In this section, we discuss the dynamic maintenance of indices for the insertion/deletion of an edge e in G. After an edge insertion/deletion, we propose the following steps to maintain EW-G. (i) Identification of affected edges: we propose a tight upper bound to estimate the k'-wing subgraph that an edge could move to after insertion/deletion so that we can effectively identify the affected edges. E', with low cost. (ii) Identification of affected super nodes: since all the k’-butterfly equivalent edges in the graph are grouped to form super nodes in EW-G, the scope of the affected super nodes can be derived using the scope of the affected edges. (iii) Dynamic Maintenance Algorithm: using steps (i) and (ii) we propose a new algorithm. We build a new partial EW-G index only for the E' induced subgraph. Further, we carefully identify a subgraph induced by vertices contained in E' that preserves the butterfly connectivity between the affected and unaffected edges. Using this, we can safely replace the affected super nodes with the newly computed partial EW-G. The dynamic update for the EW-T is similar to that for EW-G.

VI. EXPERIMENTS

We conduct a comprehensive evaluation of all the three proposed approaches, Baseline, EW-G and EW-T, across 7 real-world datasets. The evaluation process includes: (i) Comparing the size and index construction time of EW-G and EW-T. (ii) Query processing time, measured by varying degree percentile and the parameter k. (iii) Incremental maintenance comparison of EW-G and EW-T. Detailed results can be found in the full version [1].

Summary of comparison. We observe that EW-T is more compact than EW-G, with BaseLine being the least efficient for query processing time. Index-based approaches outperform BaseLine significantly. As the size difference in EW-G and EW-T increases, EW-G becomes much slower, while EW-T remains at least 10 times faster (Fig. 2). EW-T is the most efficient k-wing search method, surpassing EW-G by at least one order of magnitude for high-degree queries and maximum k values. Incremental update approaches, especially for insertion, are orders of magnitude faster than constructing EW-G and EW-T from scratch upon graph updates. The upper bound effectively estimates the wing number of an inserted edge, avoiding the expensive computation of the precise wing number. Overall, experimental results highlight the superiority of index-based approaches (EW-G and EW-T) over the baseline approach.

Acknowledgment. This work is jointly supported by the ARC Discovery Projects under Grant No. DP200103700 and DP220102191.

REFERENCES

1. Aman Abidi et al. “Searching Personalized k-wing in Bipartite Graphs”. In: TKDE (2022), pp. 1–14.
2. Esra Akbas and Peixiang Zhao. “Truss-based Community Search: a Truss-equivalence Based Indexing Approach”. In: PVLDB 10.11 (2017), pp. 1298–1309.
3. Chun-Ta Lu et al. “Inferring the impacts of social media on crowdfunding”. In: WSDM 2014, pp. 573–582.
4. Ahmet Erdem Sariyuce and Ali Pinar. “Peeling Bipartite Networks for Dense Subgraph Discovery”. In: WSDM 2018, pp. 504–512.
5. Kai Wang et al. “Efficient Bitruss Decomposition for Large-scale Bipartite Graphs”. In: ICDE 2020, pp. 661–672.