The Proof of Theorem 4.1

Previously, BatchEcore [1] first proposed to do community search on HGs and it is still the state-of-the-art method. To address the low engagement problem (i.e., the "spurious viewer" u_3 in Fig. 1 of paper), [1] proposed three different models to find the community in HG, which are edge and vertex repeatable, edge-disjoint, and vertex-disjoint, respectively. The vertex-disjoint model may lead to many unconnected vertices, as the number of vertices in bifurcated meta-structures is often limited. BatchEcore is an algorithm that requires edge-disjoint meta-structures, removing vertices with core numbers less than k in a batch. It computes the number of meta-path-based neighbors of a vertex using the Max Flow algorithm[2]. Referring to BatchEcore, we define $\beta(v, C)$ as the edge-disjoint meta-structure instances which start from vertex v and end at vertex in set C. Our objective is to find a community C such that $\forall v \in C, \beta(v, C) \geq k$. Inspired by this, given the HGs and the searching meta-structure, we reconstruct the original HG into a new graph. It is perfectly feasible to apply the BatchEcore method directly on the reconstructed graph, not only to allow edges that connect the bifurcations to be repeated, but also to efficiently compute the exact $\beta(v, C)$. The requirement that edges at a bifurcation can be repeated is equivalent to the maximum community with edge-disjoint searched by BatchECore.

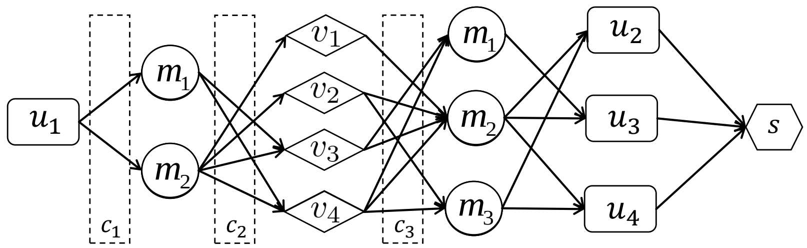

Figure 1: The Flow Network

Proof. BatchEcore algorithm aims to find the largest community, so the $\beta(v, S)$ of each vertex in the community C is computed independently by the maximum flow algorithm. This makes it possible to reuse certain edges when computing $\beta(v', S)$ for different vertices v', but it does not lead to an increase in the value of k. It can be seen from Fig. 1 that the addition of combinatorial vertices causes changes to the flow network. But this does not change the maximum flow of the network. The max-flow min-cut theorem [2] states that in a flow network, if there is a maximum flow f in the network, the flow value of f is equal to the capacity of the minimum cut. An s-t cut $C = (S, T)$ is a division of the vertices of the network into two parts S, T, and $s \in S$ and $t \in T$. We use c to denote the number of edges that connect the source part of the cut to the sink part. For example, in Fig. 1, $c1, c2, c3$ and so on are all cuts of this flow network, and we denote the smallest cut as c_{min}.

For community search, we construct a flow network with vertex u as the source node and specify the capacity of each edge as 1. There is only one vertex in the layer of the source node u. So if we take u as S and the other vertices as T, the size of the cut, denote as c_1, is the number of vertices connected to u. We use f to denote the maximum flow of this network, obviously, $c \geq f$. The addition of combinatorial vertices will bring a change in the cuts of the two layers, for example, c_2 and c_3 in Fig. 1. Combinatorial vertices are virtual vertices used to connect different vertices that can be connected by a meta-structural bifurcation. Therefore, the minimum value of c_2 and c_3 must be no less than c_1, which can ultimately be formalised as: $min(c_2, c_3) \geq c_1 \geq f$. It can be seen that the addition of combined vertex layers does not change the max flow f. So we can directly use the BatchEcore method on the reconstructed graph, and can achieve the goal of allowing the edges at the bifurcation to be repeated. ◻

REFERENCES

[1] Fang, Y., Yang, Y., Zhang, W., Lin, X., Cao, X.: Effective and efficient community search over large heterogeneous information networks. Proceedings of the VLDB Endowment **13**(6), 854–867 (2020)

[2] Seymour, P.: The matroids with the max-flow min-cut property. Journal of Combinatorial Theory, Series B **23**(2), 189–222 (1977). https://doi.org/https://doi.org/10.1016/0095-8956(77)90031-4, https://www.sciencedirect.com/science/article/pii/0095895677900314