World Wide Web manuscript No.
(will be inserted by the editor)

MTP: Discovering High Quality Partitions in Real World Graphs

Yongsub Lim · Won-Jo Lee ·
Ho-Jin Choi · U Kang

Received: date / Accepted: date

Abstract Given a real world graph, how can we find a large subgraph whose partition quality is much better than the original? How can we use a partition of that subgraph to discover a high quality global partition? Although graph partitioning especially with balanced sizes has received attentions in various applications, it is known NP-hard, and also known that there is no good cut at a large scale for real graphs.

In this paper, we propose a novel approach for graph partitioning. Our first focus is on finding a large subgraph with high quality partitions, in terms of conductance. Despite the difficulty of the task for the whole graph, we observe that there is a large connected subgraph whose partition quality is much better than the original. Our proposed method MTP finds such a subgraph

This work was supported by the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2015K1A3A1A14021055), and IT R&D program of MSIP/IITP [10044970, Development of Core Technology for Human-like Self-taught Learning based on Symbolic Approach]. This work was also funded by Institute for Information communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R0190-15-2012, "High Performance Big Data Analytics Platform Performance Acceleration Technologies Development"). The ICT at Seoul National University provides research facilities for this study. The Institute of Engineering Research at Seoul National University provided research facilities for this work.

Y. Lim
Seoul National University
E-mail: yongsub@snu.ac.kr

W. Lee
KAIST
E-mail: mochagold@gmail.com

H. Choi
KAIST
E-mail: hojinc@kaist.ac.kr

U Kang (corresponding author)
Seoul National University
E-mail: ukang@snu.ac.kr

by removing "hub" nodes with large degrees, and taking the remaining giant connected component. Further, we extend MTP to *gb*MTP (Global Balanced MTP) for discovering a global balanced partition. *gb*MTP attaches the excluded nodes in MTP to the partition found by MTP in a greedy way. In experiments, we demonstrate that MTP finds a subgraph of a large size with low conductance graph partitions, compared with competing methods. We also show that the competitors cannot find connected subgraphs while our method does, by construction. This improvement in partition quality for the subgraph is especially noticeable for large scale cuts—for a balanced partition, down to 14% of the original conductance with the subgraph size 70% of the total. As a result, the found subgraph has clear partitions at almost all scales compared with the original. Moreover, *gb*MTP generally discovers global balanced partitions whose conductance are lower than those found by METIS, the state-of-the-art graph partitioning method.

Keywords Graph partition · Balanced graph partition · Conductance

1 Introduction

In a real world graph, how can we choose a large subset of nodes for which high quality partitions exist compared with the whole graph? How can we find a high quality global balanced partition? Graph partitioning has become an important task due to its wide applications in the real world, including community detection [14], load balancing in distributed systems [52], VLSI design [45], and image segmentation in computer vision [46]. The problem is conceptually well-described and involves grouping nodes so that a group has many internal edges and few external edges, which is usually evaluated by the number of edges across the groups. Especially, in practice, enforcing groups to have balanced sizes is often required. This constraint, however, makes the problem NP-hard, and thus various approaches have been proposed in wide research areas including data mining, computer vision, and theory [46,39,23, 53,56]. Despite such extensive studies, there have been also negative results on graph partitioning targeted at all the nodes for real graphs: e.g., it is difficult to find a good cut at a large scale in real world graphs [30].

In this paper, we deal with two problems. First, we find a large subset of nodes that has high quality partitions compared with the whole graphs. It can be understood to identify a large portion of the total for which the problem has a much better solution than for the total. This approach also has various applications like community detection in social networks where communities clearly exist but are hidden or blurred due to other structural properties of the networks. Second, we find a global balanced partition by extending the first partitioning result for a subset of nodes. To measure quality of a partition, we use conductance [56,30,22,44], a widely used measure described in Section 2. Conceptually, the conductance measures how clearly a group is separated from the other part, and thus especially considers bipartitioning which is used as a basic building block for more general multi-way graph partitioning.

Our main idea is simple and intuitive: remove *problematic* nodes, which we will define soon, and work with the remaining well-handled nodes. For the purpose of graph partitioning, there are two sorts of problematic nodes: 1) large degree nodes called hub nodes which increase interdependency between groups, and 2) spokes attached only to the hub nodes which do not contribute to homogeneity within any group. From this idea, we propose MTP (Minus Top-k Partition) which removes hub nodes and computes a partition only for the remaining giant connected component. As a result, conductance of the resulting partition is much lower than that for the whole graph while the size of the giant connected component (GCC) remains significant—remarkable for partitions at large scales like a balanced partition. MTP is also efficient in terms of time and space. Excluding the partitioning step, the time and space complexities are linear on a graph size. Empirically, using the state-of-the-art graph partitioning method METIS, we show that MTP has a linear running time on a graph size.

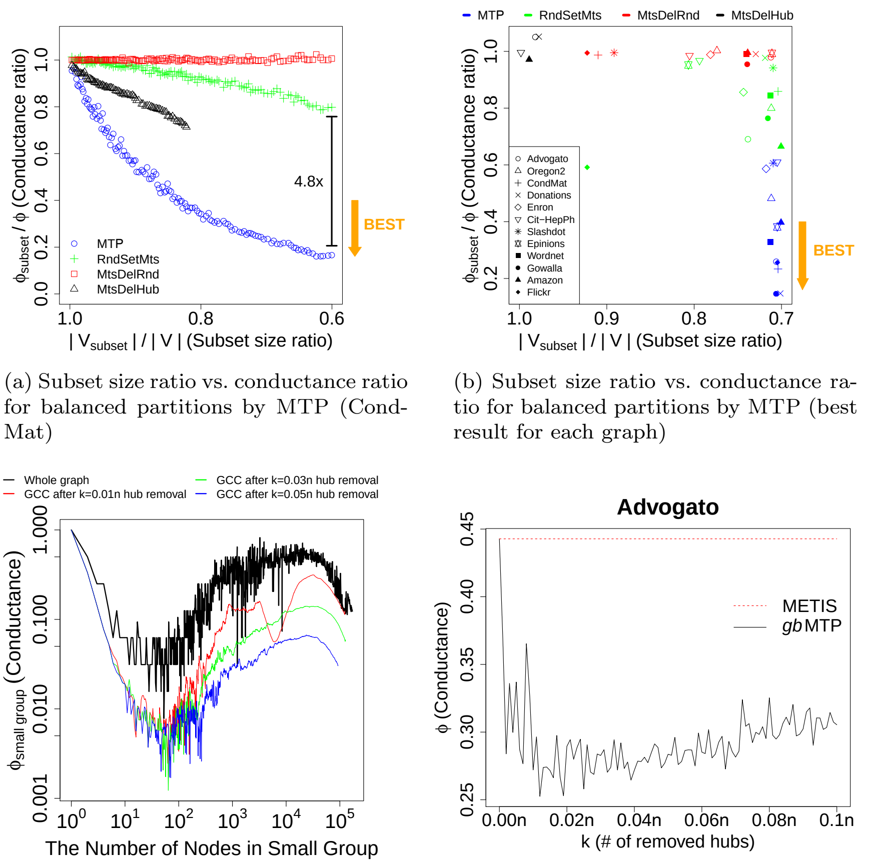
Furthermore, based on a result of MTP, we devise *gb*MTP for global balanced partitioning. It attaches the hubs and the spokes excluded in MTP to a balanced partition of a subgraph found by MTP in a greedy way. Our empirical studies show that in general, *gb*MTP discovers a global balanced partition with lower conductance than that found by METIS. Additionally, we show that *gb*MTP works better than METIS for ℓ-way partitioning for most datasets in terms of not only conductance but also the normalized cut.

Fig. 1 summarizes our results. Fig. 1a shows the result for CondMat graph data where a subset of nodes found by MTP has a balanced partition whose conductance is lower than that for the whole graph, and also than that found by competing methods. Fig. 1b compares MTP and the competitors for all graph data used in our paper; note that MTP consistently outperforms the others. Fig. 1c shows that SUBSETS[1] found by MTP reduce conductance, compared with the whole graph, at all size scales. Fig. 1d shows the performance of *gb*MTP on Advogato graph data. It discovers a global balanced partition with lower conductance than that by METIS which corresponds to $k = 0$ in the plot.

Our main contributions are summarized as follows.

- **New Problem**: We consider the new problem of finding a large subgraph which has much higher quality partitions compared with the original graph. To solve the problem, we develop MTP which removes hub nodes and takes the remaining giant connected component.
- **Extension**: We propose *gb*MTP to find a global balanced partition by extending MTP. It starts with a partial solution which is a balanced partition for the subgraph found by MTP, and enlarges it by attaching the nodes excluded by MTP in a greedy way.
- **Performance**: We show that as more hubs and the corresponding spokes are removed, conductance of a balanced partition for the remaining giant component gets much lower—down to 14% of the original while the GCC

[1] We use SUBSET to indicate a set of nodes in a graph, and SUBSETS for its plural.



(a) Subset size ratio vs. conductance ratio for balanced partitions by MTP (CondMat)

(b) Subset size ratio vs. conductance ratio for balanced partitions by MTP (best result for each graph)

(c) NCP plot for Flickr by MTP

(d) Conductance of global balanced partitions by *gb*MTP over varying k (Advogato)

Fig. 1: Our proposed MTP and *gb*MTP methods outperform competitors. Here, $|V|$ denotes the number of nodes in the original graph; ϕ denotes conductance of a balanced partition by METIS in (a) and (b); ϕ denotes conductance of a balanced partition by *gb*MTP in (d). (a) Performance of MTP for CondMat graph data, compared with other competitors described in Section 5. For a balanced partition, the SUBSET found by MTP has significantly lower conductance than the whole graph and also lower than for SUBSETS found by the competitors. (b) Ratio of subset size vs. ratio of conductance for a balanced partition for each graph and each method. Each point chosen is the one having the minimum conductance among the results with a subset size ratio at least 0.7 in Fig. 7. Note that for all graphs, MTP finds a large SUBSET whose conductance for a balanced partition is effectively reduced compared with that for the whole graph. In contrast, the competitors fail to find such SUBSETS. (c) NCP plot for Flickr data showing that the SUBSET found by MTP has imbalanced partitions at various size scales with lower conductance than does the whole graph. Here, $n = |V|$. Details of the NCP plot is explained in Section 2.1. (d) Performance of *gb*MTP for Advogato graph data. Note that $k = 0$ corresponds to METIS (the red dashed line). On the whole range of k, *gb*MTP outperforms METIS.

Table 1: Symbol table.

Symbol	Definition
G	a graph
V	a set of nodes in the whole graph
V_{SUBSET}	a set of nodes in the SUBSET
n	the number of nodes of the whole graph
m	the number of edges of the whole graph
k	the number of hub nodes removed
ϕ	conductance of a balanced partition for the whole graph
ϕ_{SUBSET}	conductance of a balanced partition for the SUBSET

size remains 70% of the total. We also show that the found SUBSET has partitions with lower conductance than the whole graph at *all* size scales, in addition to the balanced case. The running time of MTP with the state-of-the-art graph partitioning method METIS is linear on a graph size. Lastly, in general, *gb*MTP generally discovers a global balanced partition whose quality is better than that by METIS.

The codes and data used in this paper are available at `http://kdmlab.org/mtpj`. The rest of the paper is organized as follows. In Section 2, we give brief preliminaries and discuss related work. We describe the proposed method MTP based on our main idea and discuss complexities of MTP in Section 3. In Section 4, we present *gb*MTP which is an extension of MTP to compute a global balanced partition. After presenting experimental results including comparisons of MTP and *gb*MTP with other competitors in Section 5, we conclude in Section 6.

Table 1 lists the symbols used in this paper.

2 Background

2.1 Preliminaries

Graph Conductance Conductance is a metric widely used to evaluate the quality of a graph partition [22,44]. Roughly, this is related to how fast a random walker starting in one group can move to another group. Thus, as connectivity of a group gets internally stronger and externally weaker, its conductance gets lower. Given a graph $G = (V, E)$, the formal definition of conductance $\varphi(A)$ for $A \subseteq V$ is as follows.

$$\varphi(A) = \frac{cut(A)}{\min\left\{vol(A), vol(\bar{A})\right\}},$$

where $cut(A) = |\{(u,v) \in E : u \in A, v \in \bar{A}\}|$ and $vol(A) = \sum_{u \in A} \deg(u)$. Note that φ gets smaller as not only the number $cut(A)$ of cross edges tends to be

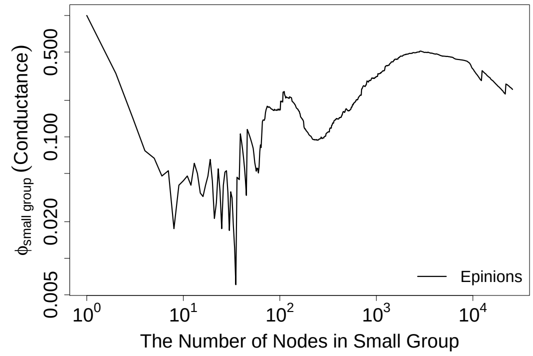


Fig. 2: Example of an NCP plot. This plot shows conductance changes of partitions into two groups at various scales. Recent work [30] reports that NCP plots of real world graphs exhibit V-shapes with the minimum at a small group size of 10 $\sim$ 100.

small but also two groups tend to have similar volumes. However, minimizing φ over $A \subset V$ is known to be NP-hard [22]. This minimum value is called the graph conductance of G. Recent work reports that conductance shows the best performance in finding ground-truth communities [56].

Network Community Profile (NCP) Plot [30] Given a graph, an NCP plot is a plot showing change of conductance over community sizes. Concretely, the x-axis corresponds to the community size and the y-axis to the corresponding conductance. In the original paper [30], drawing the NCP plot in a log-log scale, the authors observed the pattern that NCP plots of real world graphs form V-shapes where the valleys are found around community sizes of 10 $\sim$ 100. This states the important structural property of real world graphs that only at a small scale, a good partition exists. Fig. 2 shows an example of the NCP plot for Epinions graph data[2]. The point at x and y implies that y is conductance for a partition of two groups with sizes x and $n - x$ where $x \leq \lfloor n/2 \rfloor$.

METIS METIS is a graph partitioning method based on multilevel ℓ-way partitioning algorithms [23]. The overall sequence of METIS consists of three phases: coarsening, initial-partitioning, and refining. In the coarsening phase, a graph is coarsened by aggregating nodes. Starting with the original graph $G_0 = (V_0, E_0)$, for every iteration, nodes in V_i are coalesced to form 'larger' nodes, resulting in V_{i+1} of a smaller size than V_i. In the initial-partitioning phase, ℓ-way partitioning of G_T is computed, where T is the number of iterations in the first phase. Among several ℓ-way partitioning algorithms [13, 18], METIS adopts a multilevel recursive bisection algorithm [23]. In the refining

[2] http://snap.stanford.edu/data/index.html

phase, graph G_T is projected to the original graph G_0 by passing through $G_{T-1}, G_{T-2}, ...G_1$ with refinement. A simplified version of Kernighan-Lin partitioning algorithm [24] which incrementally swaps nodes to reduce cross edges of the partitioning was used for the refinement [16,17]. Recently, METIS has been improved in performance especially for power-law graphs [1].

2.2 Related Work

There have been a number of studies on graph partitioning, including METIS [23], spectral clustering [46], cross-association [7], co-clustering [10], and label propagation [52,42]. Despite different objective functions, they explicitly or implicitly share a common concept of partitioned groups: many intra-edges and few inter-edges.

Overlapping Graph Partitioning Often, the problem allows or requires overlapping. For example, in community detection for social networks, it may be more natural that people belong to several communities. For overlapping graph partitioning, in recent years various methods have been proposed, including an axiom based method [5], a probabilistic model based method [15], a matrix factorization based method [57], line grouping [11,2], and a link-space transformation method [31].

Balanced Graph Partitioning One issue frequently encountered in practice for graph partitioning is about balancing sizes of partitioned groups. To handle this size constraint, researchers have proposed various metrics such as normalized cut [46], ratio cut [53], and conductance [22]. In general, directly optimizing such metrics is NP-hard, and thus many approximate algorithms and heuristics have been developed [53,46,28]. However, since they were not designed for strict balancing, optimizing those metrics often results in quite imbalanced partitioning. More strictly balanced partitioning has been also studied theoretically [4,48] and empirically [23,43,6]. The more general problem of size-constrained graph partitioning has been also studied in various fields [26,33,32,38]. Recently, a number of methods to tackle the problem for graph streams have been developed [50,51,49,41]. To be more practical, Xu et al. studied dynamically changed partitioning for graphs processed on a vertex-centric graph processing system like Pregel [54]. Also Xu et al. proposed a graph partitioning method for a system with heterogeneous machines on their computing power [55].

Difficulty of Finding Good Partition at Large Scale Despite numerous graph partitioning methods, it has been shown in several studies [8,30] that it is difficult to find a good cut at a large scale for real graphs. One reason is that the degree distribution of real world graphs is heavy-tailed [12,3], implying the existence of hub nodes that may seriously contribute to a large number of cross edges. Rather than finding a good cut in real graphs, researches aimed at finding and evaluating ground-truth communities have been also done with various approaches [56,40,58,25,9].

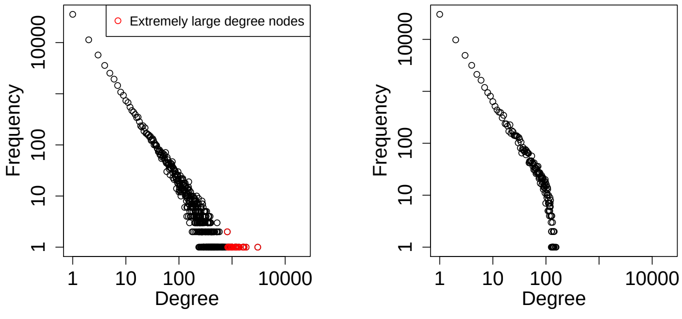
Exploiting Hub Nodes Recently, to analyze graph structure, there have been several studies that exploit the characteristic of the existence of hub nodes. Siganos et al. [47] proposed a method to group hub nodes first and recursively attach the remaining nodes, resulting in a hierarchical grouping model of a graph. In another study [9], the authors observed that the assortativity coefficient of ground-truth communities can be different from that of the whole graph, and proposed edge-weighting methods to decrease the influence of disassortative edges (e.g., hub-spoke edges), leading to finding communities with high similarity to the ground-truth. Other work [20,34], sharing a basic idea with our work, was done on graph compression. They proposed a node ordering method called SlashBurn that places hub nodes in front, and disconnected nodes appearing due to hub removal in back. Their method regards that hub nodes are few but play a considerably important role in graph structure, and thus specially handle such a property of the hubs. Besides, it is also applied to other related tasks including graph summarization [27] and graph visualization [36,37,21]. However, they focused on quickly shattering graphs by removing the hub nodes, and there was no discussion about graph partitioning after their removal. In this paper, whose preliminary version appeared in [35], we follow such a basic idea to analyze a graph having a heavy-tailed degree distribution. In addition to developing MTP, a method to discover a subgraph with a high quality partition in [35], here we propose *gb*MTP to find a high quality global partition for the whole graph, and conduct experiments to evaluate *gb*MTP. Precisely, we show that removing hub nodes remarkably decreases conductance values of partitions of the remaining graph, and attaching the removed nodes in a greedy way results in a global balanced partition whose quality is better than that by METIS.

3 Proposed Method

3.1 Motivation

One well-known characteristic of many real world graphs is that the degree distribution is heavy-tailed. This is distinct from a random graph with an exponential degree distribution. This implies that there exist hub nodes having very large degrees. In graph partitioning, particularly that with balancing, these hub nodes become seriously problematic: due to their diverse neighbors, assigning them to one group would greatly increase interdependency between groups.

Recent work shows that real world graphs are easily shattered by removing hub nodes [20]. Concretely, removing the hub nodes results in a giant connected component of a significant size, and many disconnected components of very small sizes. Although the giant connected component has a structure of hub



(a) Original Graph (b) Shattered by k hubs

Fig. 3: Comparison of degree distributions of the original graph and GCC after removing k hub nodes for Epinions graph data. Here, k is set to 1% of the total nodes in the original graph. Note that in the original graph, there exist hub nodes with extremely large degrees while in the reduced graph, there are no such nodes.

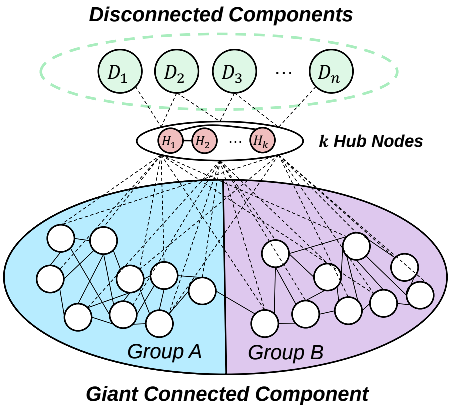


Fig. 4: Illustration of our main idea. We envision a graph consisting of three parts: hub nodes, the giant connected component and disconnected components appearing after hub nodes are removed. The dashed lines represent the edges removed with removal of hubs. Note that after removing the hub nodes, the corresponding giant connected component has a much clearer partition.

nodes similar to the whole graph, we observe that there is no hub node with an extremely large degree, as shown in Fig. 3. This observation motivated us to exploit the hubs and disconnected components for high quality partitions. Below, we explain our method, called Minus Top-k Partition (MTP), to find a large subset of nodes for which high quality partitions exist.

3.2 Minus Top-k Partition (MTP)

The main idea of MTP is to envision a graph as a collection of three parts: hub nodes, spokes only attached to the hub nodes, and the remaining part. Here, the spokes correspond to disconnected components and the remainders correspond to the giant connected component (GCC) after removing the hub nodes. Let $[n] = \{1, \ldots, n\}$ and $G(U)$ is the induced subgraph of $U \subseteq V$. If we remove the set H of hub nodes from a graph, the graph is divided into a set of p connected components $CCSET = \{CC_i \subset V \backslash H : i \in [p]\}$, satisfying

- $CC_1, \ldots, CC_p$ are mutually disjoint sets.
- For every $i \in [p]$ and any pair $(u, v) \in (CC_i)^2$, there is a path between u and v in $G(CC_i)$.
- For every pair $(i, j) \in [p]^2$ and any pair $(u, v) \in CC_i \times CC_j$, there is no path between u and v in $G(V \backslash H)$.

Then, GCC and spokes are formally defined as follows:

$$GCC = \underset{CC \in CCSET}{\text{argmax}} \; |CC|,$$
$$SPOKES = V \backslash (H \cup GCC).$$

The hub nodes become a major obstacle in finding a good partition because their diverse connectivity makes partitioned groups have high interdependency. Our approach is to exclude those problematic nodes and take the remaining giant connected component as a subgraph for which we hope to obtain a high quality partition (see Fig. 4).

MTP first finds and removes the top-k hub nodes from a graph. As a result, the graph is shattered into a number of connected components as described above. Next, MTP finds the GCC among them, which can be done using a standard graph traversal algorithm like the breadth first search (BFS). Last, it computes a partition (A, B) for the GCC, and then outputs (A, B). Although any partitioning method can be applied, in this paper we use METIS, which is considered the state-of-the-art graph partitioning method [30]. Algorithm 1 describes the whole MTP procedure.

MTP is simple, intuitive, and easily implementable. As described in Section 5, MTP discovers a SubsetPartition, a partition of a subgraph, with low conductance. Moreover, we compare MTP with other baseline methods to demonstrate non-triviality of our results. We will see that the baseline methods are not effective in reducing conductance, or that they choose a subgraph consisting of many small connected components for which a partition is not very meaningful.

3.3 Complexity Analysis

Our proposed method MTP is efficient in terms of time consumption and space usage. Excluding the partitioning step, the time complexity and the

```
Algorithm 1: Minus Top-$k$ Partition (MTP)
  Input: Graph $G$, the number of removed hubs $k$
  Output: SubsetPartition $(A, B)$
  Find the top-$k$ high degree nodes in $G$.
  Remove them from $G$.
  Find the giant connected component (GCC).
  Partition the GCC into $(A, B)$.
```

space complexity of MTP are linear on a graph size: $O(n + m)$ and $O(n)$, respectively. The detailed analysis is given below.

Lemma 1 *The time complexity of* MTP *excluding the partitioning step is* $O(n + m)$.

Proof Without computing a partition, MTP consists of the two main steps: 1) removing the top-k hub nodes, and 2) identifying the giant connected component. Step 1) involves finding the top-k hub nodes which can be done in $O(n)$ using Hoare's selection algorithm [19]; Step 2) is done by finding connected components using a standard graph traversal algorithm like the breath-first search (BFS), which takes $O(n+m)$. Hence, the total time complexity excluding the partitioning step becomes $O(n + m)$.

Although we exclude the partitioning step in the analysis since its time complexity varies with algorithms used, we empirically show in Section 5 that MTP with METIS is fast.

The next lemma states the space complexity of MTP.

Lemma 2 *The space complexity of* MTP *excluding the partitioning step is* $O(n)$.

Proof As we stated, MTP involves the two main computation steps: running the selection algorithm for the first k largest degree nodes, and running a connected component algorithm. In the first step, computing degrees of nodes require $O(n)$ space, and Hoare's selection algorithm require no additional space; in the second step, finding connected components requires $O(n)$. Combining all the space requirements, the lemma is proved.

4 Extension of MTP to Global Balanced Partitioning (*gb*MTP)

In this section, we present *gb*MTP, a global balanced graph partitioning algorithm based on MTP. The main idea is to use a balanced SubsetPartition found by MTP as a partial result for the entire graph, and greedily attach the remaining nodes, i.e. hubs and spokes, leading to a global balanced partition. Our proposed *gb*MTP is described in Algorithm 2. The algorithm is divided into the two parts of *attaching step* (Lines 3–11) and *balancing step* (Line 12–18), which are explained in details below.

Let (A, B) be a balanced SubsetPartition discovered by MTP. In the attaching step, *gb*MTP first constructs the ordered set $L = V \setminus (A \cup B)$ and attaches every node $u \in L$ to A or B greedily with respect to conductance change. To complete this step, we define the order in L, which is examined in Line 3 of Algorithm 2. Let H and S be the hubs and the spokes identified during the running of MTP; note that A, B, H and S are disjoint sets and $V = A \cup B \cup H \cup S$. The ordering is determined by the following three rules. First, H is considered before S: i.e. among the nodes not belonging to the subgraph found by MTP, the hubs are considered before the spokes. Second, among H, the nodes are considered in decreasing order on their degrees: i.e. a hub with a larger degree is considered earlier than that with a smaller degree. Third, among S, nodes belonging to the same connected component are considered consecutively and there is no order among connected components: i.e. the placement of a spoke node in the ordering is determined only by its connected component. This ordering is chosen because nodes in H affect partitioning quality more significantly than S. That is, determining the assignments of H before S is better than the opposite order in a greedy approach since high degree nodes in H affect the number of cut edges more than nodes in S. Furthermore, considering S before H is less meaningful because initially S has edges only to H, not to $A \cup B$. Consequently, *gb*MTP needs to carefully partition high degree nodes first and then accordingly assign spoke nodes. Note that L can be constructed during MTP although we separate the computation of L from MTP in Line 2 of Algorithm 2 for clarity.

Although the attaching step results in a global partition, it does not guarantee the partition having balanced sizes. This imbalance is amended in the balancing step. After the attaching step, let A be smaller than B without loss of generality. We need to select $\lfloor (|B| - |A|)/2 \rfloor$ number of nodes that are moved to A. Our approach is to greedily find the best node whose movement to A results in the smallest conductance, and move it from B to A. Applying this movement $\lfloor (|B| - |A|)/2 \rfloor$ times, we obtain balanced sizes for A and B. The problem, however, is that considering all nodes in B as candidates for the movement may take too long time. To overcome this inefficiency, we restrict the candidates for the movement to $B \cap L$, that is, we do not change the assignments made by MTP at the initial step.

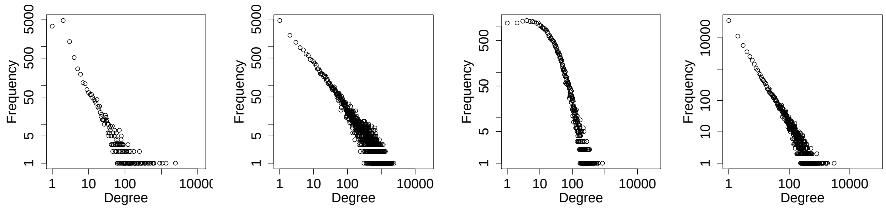
In Section 5, we show that *gb*MTP with a small k outperforms METIS, the state-of-the-art graph partitioning method. Note that as k gets smaller, the balancing step of *gb*MTP finishes more quickly. Especially, we observe that for some graphs, the improvement of *gb*MTP over METIS is significant for the whole range of k used in our experiments.

5 Experiments

In this section, experimental results are used to answer:

Algorithm 2: Global Balanced MTP (gbMTP)

```
Input: Graph $G$, the number of removed hubs $k$
Output: Global Balanced Partition $(A, B)$
$(A, B) \leftarrow MTP(G, k)$.
$L \leftarrow V \setminus (A \cup B)$.
// Attaching Step
foreach $u \in L$ in the order do
    $\tilde{\phi}_A \leftarrow Conductance(A \cup \{u\}, B)$.
    $\tilde{\phi}_B \leftarrow Conductance(A, B \cup \{u\})$.
    if $\tilde{\phi}_A < \tilde{\phi}_B$ then
        $A \leftarrow A \cup \{u\}$.
    else
        $B \leftarrow B \cup \{u\}$.
    end
end
// Balancing Step
// Let $A$ be smaller than $B$ without loss of generality
$Q \leftarrow B \cap L$.
while $||A| - |B|| > 1$ do
    $v \leftarrow \text{argmin}_{u \in Q}\, Conductance(A \cup \{u\}, B)$.
    $A \leftarrow A \cup \{v\}$.
    $B \leftarrow B \setminus \{v\}$.
    $Q \leftarrow Q \setminus \{v\}$.
end
```



(a) Oregon2 (b) Donations (c) Cit-HepPh (d) Epinions

Fig. 5: Degree distributions of some graphs described in Table 2. Note that all the data used, exhibits heavy-tailed degree distributions, which means that our main assumption of the existence of hub nodes holds. The other graphs not shown here also show similar patterns.

Q1 How low conductance does a SubsetBalPartition[3] by MTP have compared with the whole graph and with other naive methods? (Answers in Observations 1 and 3)

Q2 How do conductance values of SubsetBalPartitions found by MTP change over increasing k? (Answer in Observation 2)

[3] a balanced partition for a subset of nodes.

Table 2: Summary of the graphs used in our experiments. The number of nodes and edges are counted after taking the giant connected component with removing direction, weights, and self-edges.

Graph	**Nodes**	**Edges**	**Description**
Advogato[1]	5,054	49,821	Trust network
Oregon2[2]	11,461	32,730	Router connections
CondMat[2]	21,363	91,286	Collaboration network
Donations[3]	23,033	877,625	Who donated whom
Enron[2]	33,695	180,810	Enron email data
Cit-HepPh[2]	34,401	420,784	Citation network
Slashdot[1]	51,083	116,573	Reply network
Epinions[2]	75,877	405,739	Trust network
Wordnet[1]	142,505	642,207	Word association network
Gowalla[2]	196,591	950,327	Online social network
Amazon[2]	334,863	925,872	Co-purchasing network
Flickr[4]	404,733	2,110,078	Social network in Flickr

[1] `http://konect.uni-koblenz.de` [2] `http://snap.stanford.edu/data/index.html`
[3] `http://download.srv.cs.cmu.edu/~mmcgloho/fec/data/fec_data.html`
[4] `http://www.flickr.com`

Q3 How low conductance do SubsetPartitions by MTP at various size scales have compared with the whole graph? (Answer in Observation 4)
Q4 How fast is MTP? (Answer in Observation 5)
Q5 How good global balanced partition does *gb*MTP output? (Answer in Observation 6)
Q6 How fast is *gb*MTP? (Answer in Observation 7)
Q7 How good is *gb*MTP for ℓ-way partitioning? (Answer in Observation 8)

5.1 Settings

To verify our method, we gathered graph data from diverse domains such as social networks, collaboration networks, internet connections, and word association. We took only the giant connected component from each graph and made them have no direction, weight, and self-edges. The statistics and brief description of the graph data are presented in Table 2. Fig. 5 shows degree distributions of some of the graphs. All of them follow heavy-tailed distributions, which means that our assumption of the existence of hub nodes holds.

For partitioning, we use the METIS library of version 5.1.0 given at `http://glaros.dtc.umn.edu/gkhome/views/metis`.

5.2 Performance of MTP

We show how good SubsetBalPartition MTP discovers through extensive experiments. We examine conductance of SubsetBalPartitions found by

MTP over the number k of removed hub nodes. To this end, while varying k from 0 to $0.1n$ with interval $0.001n$, we 1) remove $\lfloor k \rfloor$ number of hub nodes from each graph, 2) run METIS to obtain a balanced partition for the giant connected component, and 3) compute conductance for the partition. Note that $k = 0$ implies applying METIS to the whole graph.

Observation 1 (High Quality SubsetBalPartition) *Conductance of a* SubsetBalPartition *discovered by* MTP *is effectively lower than that of a balanced partition for the whole graph.*

Observation 2 (Better as k Gets Larger) *As the number k of removed hub nodes gets larger, quality gap between a* SubsetBalPartition *by* MTP *and a global balanced partition gets much significant. The conductance of the* SubsetBalPartition *is down to* 14% *of the global one with* subset *size* 70% *of the total.*

Fig. 6 shows changes of conductance of SubsetBalPartitions by MTP and sizes of the corresponding subsets over the number k of removed hub nodes. In general, the conductance of the SubsetBalPartitions is smaller than that for the whole graph. Notably, as k gets larger, the conductance gap gets much significant, which is consistently exposed by all the used graphs.

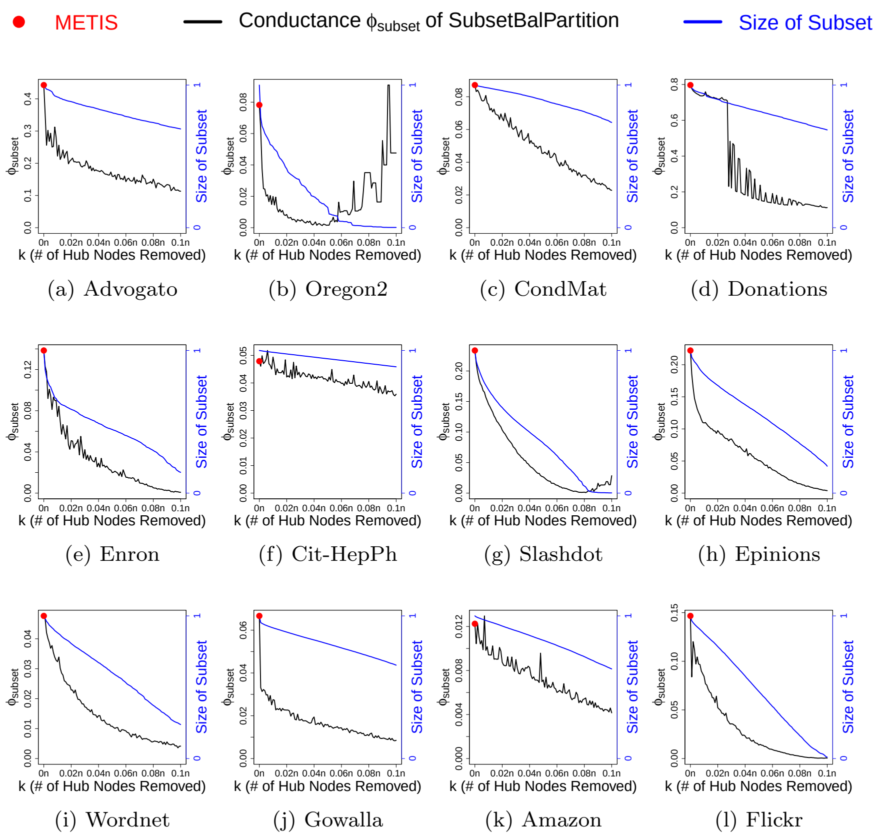
We note that size decreases of the subsets are positively correlated with conductance decreases of the corresponding SubsetBalPartitions. For example, the conductance decrease of a SubsetBalPartition is most remarkable in Oregon2 whose subset size is dramatically reduced over k while Cit-HepPh graph shows the opposite example. Moreover, for all cases, the conductance decrease is much significant compared with the subset size decrease. For example, compared with METIS applied to the whole graph, MTP finds a subset of a size at least 80% of the total, but conductance of the corresponding SubsetBalPartition becomes less than half of the original.

Next, we demonstrate the non-triviality of MTP by comparing with other competitors to find a SubsetBalPartition. Below, the competitors that we consider here are described.

- RndSetMts: Select a random subset of nodes, and apply METIS to that set.
- MtsDelRnd: Compute a balanced partition for the whole graph using METIS, and randomly remove the same number of nodes from each group.
- MtsDelHub: Compute a balanced partition for the whole graph using METIS, and remove the same number of hub nodes from each group.

Observation 3 (Non-triviality of MTP) *The competitors for finding a* SubsetBalPartition *do not decrease conductance effectively, or they result in* subsets *consisting of the giant connected component of an insignificant size and many disconnected components of very small sizes.*

Fig. 7 shows the comparison of the SubsetBalPartitions computed by MTP and the three competitors described above. Given k, the results of RndSetMts and MtsDelRnd are computed by running the methods 10 times and



(a) Advogato (b) Oregon2 (c) CondMat (d) Donations

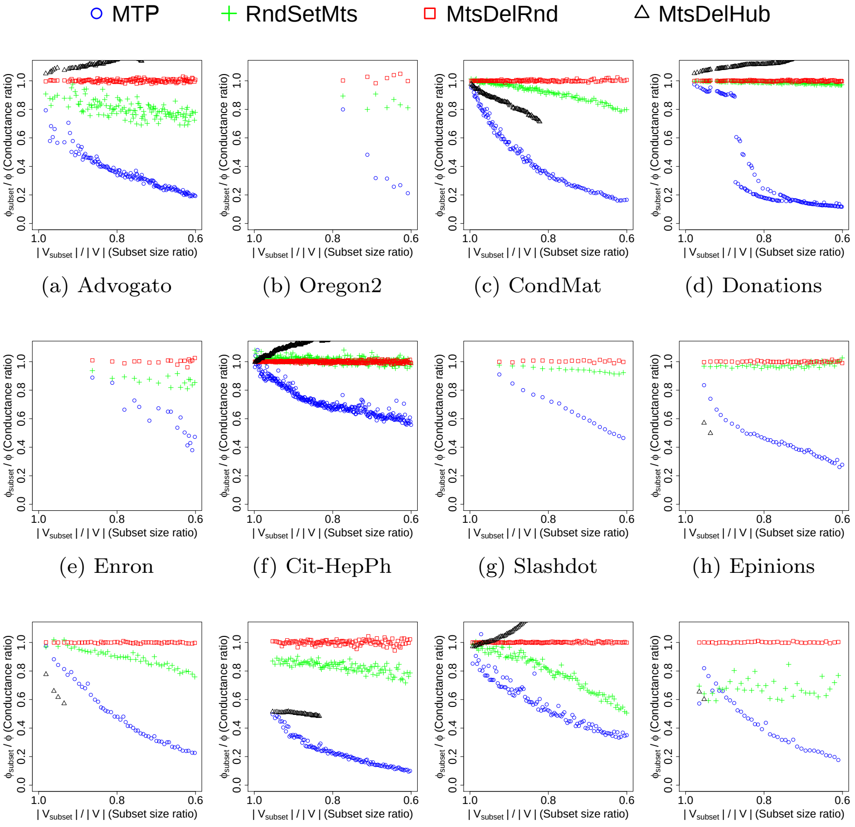
(e) Enron (f) Cit-HepPh (g) Slashdot (h) Epinions

(i) Wordnet (j) Gowalla (k) Amazon (l) Flickr

Fig. 6: MTP finds a large subset of nodes whose conductance for balanced partition is fairly reduced compared with that for the whole graph. For each plot, k denotes the number of hub nodes removed; the red dot denotes conductance computed by METIS for the whole graph; the black line denotes the conductance ϕ_{SUBSET} of the SubsetBalPartitions; and the blue line denotes the ratio of subset sizes over n. Note that the red dot also corresponds to the case of MTP with $k = 0$. Overall, conductance consistently decreases as k gets larger, and its amount is larger than the decrease of subset sizes.

taking averages of the 10 conductance values. For all the competitors, we exclude results if the corresponding subset for which the conductance is computed has a giant connected component of a size less than half of the subset size since the case is less meaningful to compute a balanced partition.

Overall, MTP results in SubsetBalPartitions with much smaller conductance than those made by competitors, especially as k gets larger (e.g., in CondMat), MTP is 4.8x better than RndSetMts, 6.1x better than MtsDelRnd, and 1.9x better than MtsDelHub. Although RndSetMts finds SubsetBalPar-



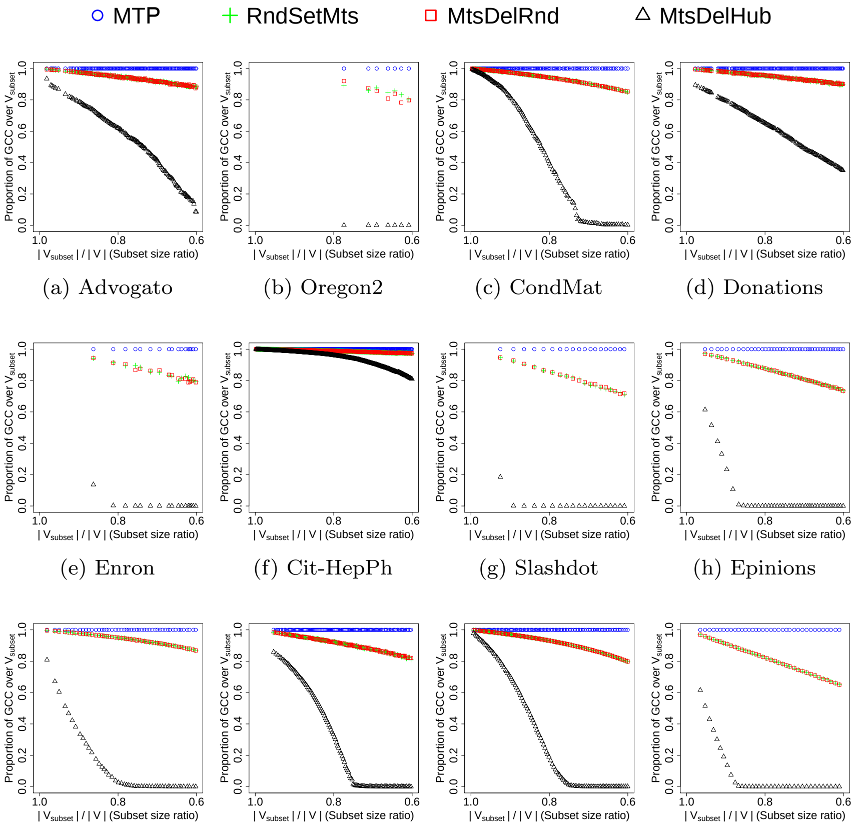
(a) Advogato (b) Oregon2 (c) CondMat (d) Donations

(e) Enron (f) Cit-HepPh (g) Slashdot (h) Epinions

(i) Wordnet (j) Gowalla (k) Amazon (l) Flickr

Fig. 7: MTP outperforms competitors in terms of conductance of a Subset-BalPartition. For RndSetMts and MtsDelRnd involving random processes, each value is computed by the average of results over 10 trials. Note that if the GCC of a computed subset is of a size less than half of the subset size, it is not presented. This is because the case is less meaningful to compute a balanced partition. For example, in CondMat, only the black line is cut off, and in extreme cases like Oregon2, there is no black line at all in the plot. Overall, RndSetMts and MtsDelRnd are not effective in reducing the conductance. Although MtsDelHub seems to reduce the conductance effectively for a few graphs, its corresponding GCC size decreases very rapidly (Fig. 8), implying that the computed subset consists of many small connected components.

titions with low conductance for some graphs like Amazon, MTP still outperforms it. The best result for each method and each graph in Fig. 7, is shown in Fig. 1b.



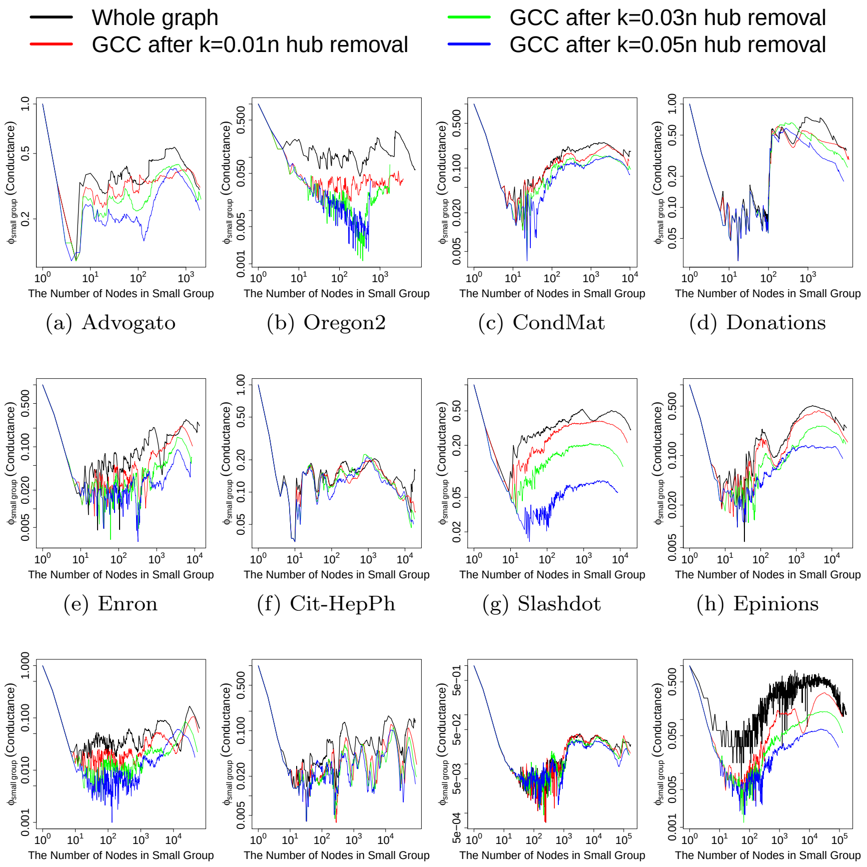
(a) Advogato (b) Oregon2 (c) CondMat (d) Donations

(e) Enron (f) Cit-HepPh (g) Slashdot (h) Epinions

(i) Wordnet (j) Gowalla (k) Amazon (l) Flickr

Fig. 8: MTP finds a connected subset by construction while subsets by competitors are disconnected. The plots show sizes of GCCs belonging to the subsets found by each method. By construction, subsets by MTP are always connected, leading to the value of 1. For RndSetMts and MtsDelRnd, the decrease of the GCC size is linear. On the other hand, for MtsDelHub, the GCC size dramatically decreases, which means that the subsets found become less meaningful even though it has a balanced partition with low conductance. The graphs not shown here also exhibit similar patterns.

Fig. 8 shows sizes of GCCs in subsets found by the four methods. None of the competitors find a connected subset at all, while subsets by MTP are always connected by construction. For RndSetMts and MtsDelRnd, their GCCs in computed subsets are quite large, but the corresponding conductance values are not reduced effectively (Fig. 7). Especially, the GCC size of a subset by MtsDelHub decreases fast with increasing k, implying that the



(a) Advogato (b) Oregon2 (c) CondMat (d) Donations

(e) Enron (f) Cit-HepPh (g) Slashdot (h) Epinions

(i) Wordnet (j) Gowalla (k) Amazon (l) Flickr

Fig. 9: Conductance of a SUBSET by MTP is lower than the whole graph not only for a balanced partition but also for partitions of various imbalanced sizes. The plots show Network Community Profile (NCP) plots [30], explained in Section 2, for each graph with different k values. For each plot, each line is computed by the SNAP library [29] for a SUBSET found by MTP with the specified k. Note that for almost all cases, the NCP plot tends to move down as k gets larger—the pattern is fairly clear, though slightly weaker for Donations, Cit-HepPh and Amazon.

SUBSET consists of small connected components in which a balanced partition becomes less meaningful.

Observation 4 (Good Partitions at All Scales) *A* SUBSET *found by* MTP *has partitions at all scales whose conductance is lower than that of the whole graph at the same scales.*

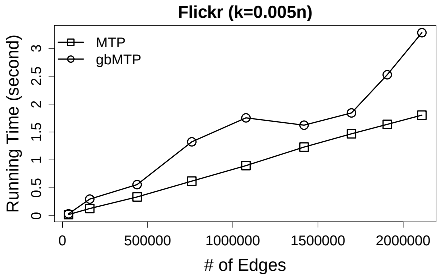


Fig. 10: Running times of MTP and gbMTP. Both run in a (nearly) linear time on the number of edges in a graph. We used principal submatrices of the adjacency matrix of the Flickr graph data. Note that gbMTP rather runs faster for a larger number of edges. This can happen in practice because the speed of gbMTP depends on the number of hubs and the corresponding spokes.

Fig. 9 depicts Network Community Profile (NCP) plots, which we explained in Section 2, for SUBSETS found by MTP with $k \in \{0, 0.01n, 0.03n, 0.05n\}$ for each graph. Each line corresponds to an NCP plot for the SUBSET obtained with the specified k. From the figure, we observe that an NCP plot gets lower as k becomes larger. For most of the graphs, the NCP plots are clearly separated—it is remarkable especially for Slashdot, Wordnet and Flickr. This means that MTP finds a SUBSET in which partitions at various scales have lower conductance compared with those for the whole graph at the same scales. In other words, the found SUBSET by MTP is partitioned much clearly compared with the whole graph at any scale while keeping V-shape patterns observed in real world graphs.

Observation 5 (Linear Running Time of MTP) *Running time of* MTP *is linear on the number of edges in a graph.*

With METIS used for the partitioning step, the running time of MTP is linear on the number of edges in a graph as shown in Fig. 10. We took principal submatrices[4] of the adjacency matrix of Flickr to make graphs with appropriate sizes.

5.3 Performance of gbMTP

We evaluate gbMTP in terms of conductance by comparing with METIS, the state-of-the-art graph partitioning method.

Observation 6 (gbMTP Better than METIS) *With small $k \leq 100$,* gbMTP *finds global balanced partitions whose conductance values are lower than those by* METIS *for almost all graphs. Examining more k values, the*

[4] A principal submatrix with size $n' \times n'$ of a matrix A with size $n \times n$ for $n' \leq n$ is a submatrix by taking the first n' rows and columns from A.

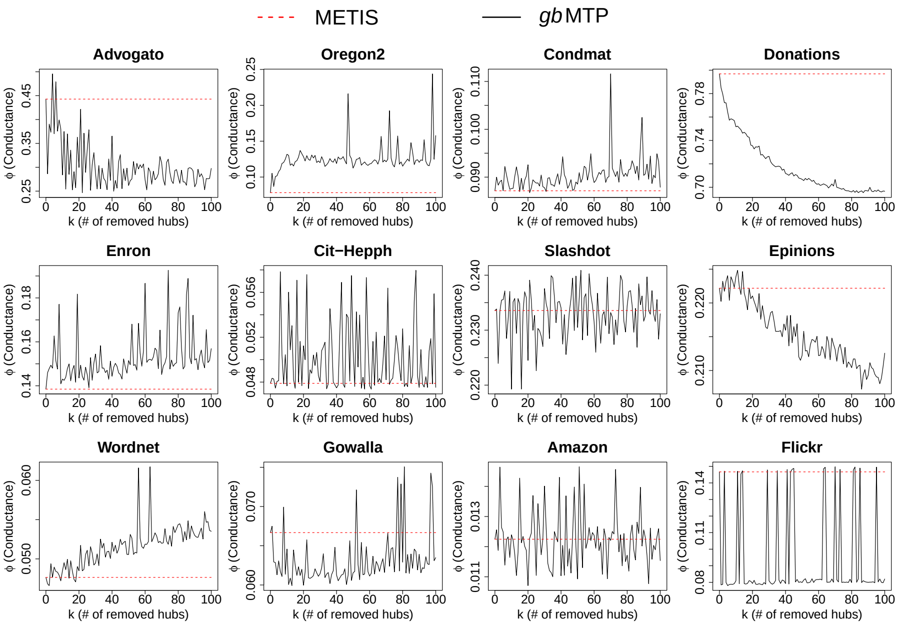


Fig. 11: Performance of gbMTP on varying $k = 0$ to 100. Note that $k = 0$ corresponds to METIS. Except for Flickr, gbMTP finds global balanced partitions whose conductance is lower than those by METIS.

improvement becomes significant for some graphs like Advogato, Donations, and Flickr.

Fig. 11 shows conductance of global balanced partitions found by gbMTP over varying $k = 0$ to 100. We observe that except for Oregon2 and Enron, gbMTP finds a global balanced partition whose quality is better than that by METIS ($k = 0$), denoted by the red dashed line, for some $k > 0$. Especially, for more than half of the graphs, i.e. Advogato, Donations, Slashdot, Epinions, Gowalla, Amazon, and Flickr, the improvements are achieved over wide ranges of k.

Fig. 12 depicts the best improvement achieved by gbMTP over METIS. For each graph, the best partition is chosen from the results shown in Fig. 11 and results additionally obtained by running gbMTP for $k = 0.001n$ to $0.1n$ at the interval of $0.001n$. Note that gbMTP discovers a global balanced partition whose conductance is lower than that found by METIS for most cases. Especially, we observe that for Advogato, Donations, and Flickr, gbMTP outputs significantly improved partitions compared with METIS.

Observation 7 (Linear Running Time of *gb*MTP) *Running time of* gbMTP *is nearly linear on the number of edges in a graph.*

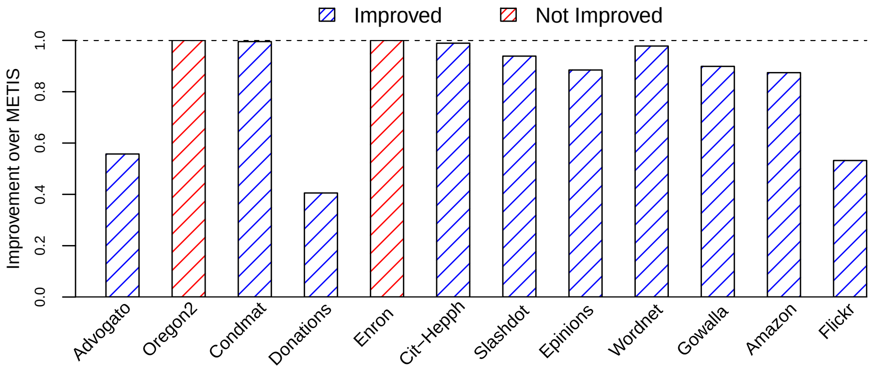


Fig. 12: The best improvement by gbMTP over METIS. In addition to the results in Fig. 11, we obtain more by running gbMTP for $k = 0.001n$ to $k = 0.1n$ at the interval of $0.001n$. Among all the results, the best global balanced partition is compared with METIS. Except for Oregon2 and Enron, gbMTP discovers a balanced partition better than that by METIS, and especially the improvement is significant for Advogato, Donations, and Flickr.

Fig. 10 shows that gbMTP with METIS runs in a nearly linear time on the number of edges in a graph. We used the same principal submatrices as used for MTP. For some cases, it runs faster though the number of edges increase. This is because the amount of operations in gbMTP depends on the number of hub nodes and spokes which are attached to an initial partial partition computed by MTP.

We also conduct experiments for multiway graph partitioning. Note that our gbMTP can be easily extended for ℓ-way graph partitioning. Here, we consider two extensions of gbMTP, which are based on recursive partitioning:

- $gb\text{MTP}_1$: For the first partitioning, gbMTP is used and for successive partitioning, METIS is used.
- $gb\text{MTP}_*$: For every partitioning, gbMTP is used.

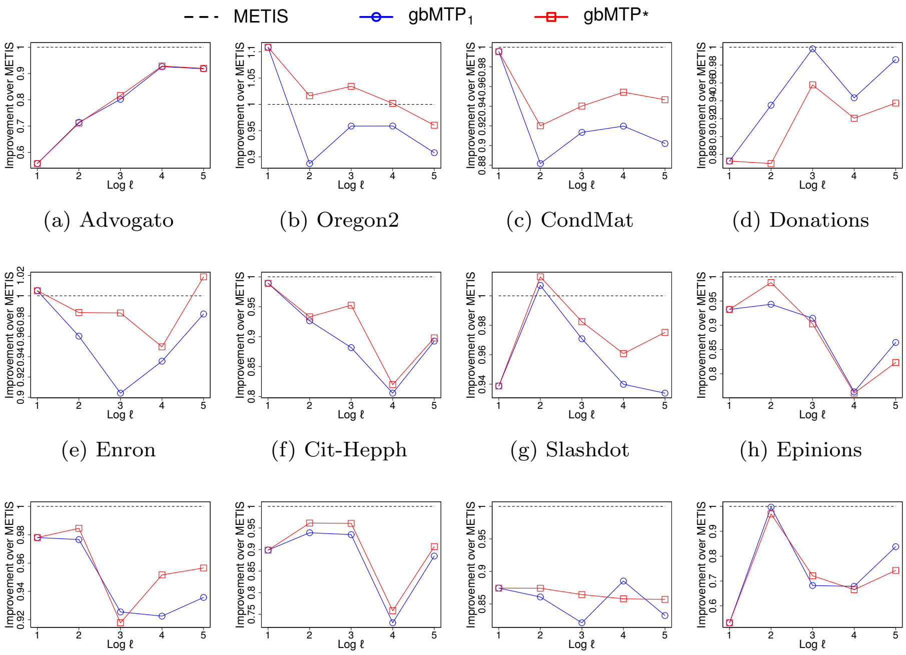
Note that given disjoint sets $V_1, \ldots, V_\ell \subseteq V$ of nodes, conductance is defined by

$$\varphi(V_1, \ldots, V_\ell) = \max_{1 \le i \le \ell} \frac{cut(V_i)}{vol(V_i)}.$$

Observation 8 (gbMTP Better than METIS for ℓ-way Partitioning) *In ℓ-way graph partitioning, $gb\text{MTP}_1$ and $gb\text{MTP}_*$ generally outperform* METIS.

Fig. 13 shows comparison between gbMTP and MTP for ℓ-way graph partitioning. The x-axis corresponds to $\log \ell$. We run $gb\text{MTP}_1$ and $gb\text{MTP}_*$ for $k = \{1, \ldots, 100\}$ and take the best for each $x = \log \ell$. Note that for most cases, the extensions of gbMTP result in lower conductance than METIS.

Especially, gbMTP$_1$ has only three exceptions: $\ell = 2$ for Oregon2, $\ell = 2$ for Enron, and $\ell = 4$ for Slashdot.



(a) Advogato (b) Oregon2 (c) CondMat (d) Donations

(e) Enron (f) Cit-Hepph (g) Slashdot (h) Epinions

(i) Wordnet (j) Gowalla (k) Amazon (l) Flickr

Fig. 13: Performance of ℓ-way partitioning using gbMTP. The x-axis means $\log \ell$. For each x value, we take the best among the results obtained by gbMTP$_1$ and gbMTP$_*$ with $k = \{1, \ldots, 100\}$. For most cases, the gbMTP based extensions find partitions whose conductance is lower than those by METIS.

We additionally evaluate a partition quality using the normalized cut [46]. Given disjoint sets $V_1, \ldots, V_\ell \subseteq V$ of nodes, the normalized cut is defined by

$$Ncut(V_1 \ldots, V_\ell) = \sum_{1 \leq i \leq \ell} \frac{cut(V_i)}{vol(V_i)}.$$

We use $\psi(V_1 \ldots, V_\ell) = Ncut(V_1 \ldots, V_\ell)/\ell$ for the ease of presentation. This makes no effect in our case since the comparison is done at the same ℓ. Fig. 14 shows the comparison result. Note that gbMTP$_1$ generally outperforms METIS. The result involves an important implication. For instance, gbMTP$_1$ is better in conductance but worse in the normalized cut than METIS for Slashdot. By definition, this means that gbMTP$_1$ divides the graph into groups with similar volumes while METIS results in groups with quite different sizes.

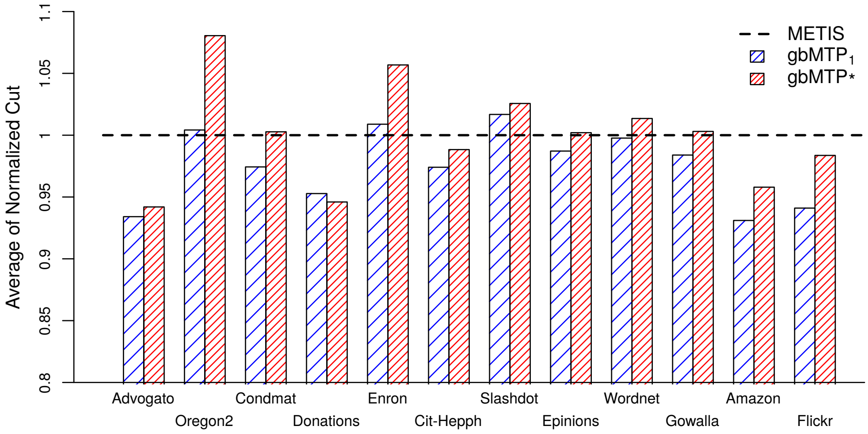


Fig. 14: Comparison between *gb*MTP and METIS in the normalized cut for ℓ-way partitioning. The value for *gb*MTP is calcualted by averaging ψ of the partitioning results with sizes $\{2, 4, 6, 8, 16, 32\}$. Especially, *gb*MTP$_1$ shows good performance: it is better than METIS for most datasets, except only for three.

This characteristic of *gb*MTP$_1$ is highly preferred in many real applications like load balancing for distributed computing.

6 Conclusion

In this paper, we tackle the graph partitioning problem. Although the problem is known to be hard to solve, we observe that real graphs have large subgraphs with high quality partitions for all size scales compared with the original graph. Based on this observation, we propose MTP to discover those subgraphs. Furthermore, we extend MTP to *gb*MTP to find a global balanced partition with low conductance by carefully attaching the remaining nodes to a balanced partition for the found subgraph. Our experimental results show that MTP discovers a SUBSET of a significant size with lower conductance than the whole graph for a balanced partition, down to 14% of the original conductance with a SUBSET of size 70% of the total. We also show that the found SUBSET has partitions whose qualities are higher than those for the whole graph at almost all size scales. Moreover, for most cases, *gb*MTP finds a global balanced partition whose quality is better than that found by METIS, the state-of-the-art partitioning method.

We expect that our research on finding a subgraph having high quality partitions would give a new direction for graph partitioning. Such a subgraph helps understand the original graph structure hidden at the global view, and can be enlarged as a global partition if needed. Future work includes scaling up the graph partitioning methods for very large graphs, using distributed systems.

References

1. Amine Abou-Rjeili and George Karypis. Multilevel algorithms for partitioning power-law graphs. In *Proceedings of the 20th International Conference on Parallel and Distributed Processing*, pages 124–124, 2006.
2. Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.
3. R. Albert, H. Jeong, and A.L. Barabási. Internet: Diameter of the world-wide web. *Nature*, 401(6749):130–131, 1999.
4. Reid Andersen, Fan R. K. Chung, and Kevin J. Lang. Local graph partitioning using pagerank vectors. In *47th Annual IEEE Symposium on Foundations of Computer Science*, pages 475–486, 2006.
5. Sanjeev Arora, Rong Ge, Sushant Sachdeva, and Grant Schoenebeck. Finding overlapping communities in social networks: Toward a rigorous approach. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 37–54, 2012.
6. Florian Bourse, Marc Lelarge, and Milan Vojnovic. Balanced graph edge partition. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1456–1465, 2014.
7. Deepayan Chakrabarti, Spiros Papadimitriou, Dharmendra S. Modha, and Christos Faloutsos. Fully automatic cross-associations. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 79–88, 2004.
8. Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proc. Natl. Acad. Sci. U.S.A.*, 99(25):15879–15882, 2002.
9. Marek Ciglan, Michal Laclavík, and Kjetil Nørvåg. On community detection in real-world networks and the importance of degree assortativity. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1007–1015, 2013.
10. Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98, 2003.
11. T. S. Evans and R. Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical Review E*, 80(1):016105+, 2009.
12. Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 251–262, 1999.
13. C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proceedings of the 19th Design Automation Conference*, pages 175–181, 1982.
14. Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.
15. Prem Gopalan, David M. Mimno, Sean Gerrish, Michael J. Freedman, and David M. Blei. Scalable inference of overlapping communities. In *26th Annual Conference on Neural Information Processing Systems*, pages 2258–2266, 2012.
16. Bruce Hendrickson and Robert Leland. The chaco user's guide version 2.0. Technical report, Sandia National Laboratories, 1995.
17. Bruce Hendrickson and Robert Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM J. Sci. Comput.*, 16(2):452–469, 1995.
18. Bruce Hendrickson and Robert W. Leland. A multi-level algorithm for partitioning graphs. In *Proceedings Supercomputing*, page 28, 1995.

19. C. A. R. Hoare. Algorithm 65: Find. *Communications of the ACM*, 4(7):321–322, July 1961.
20. U. Kang and Christos Faloutsos. Beyond 'caveman communities': Hubs and spokes for graph compression and mining. In *11th IEEE International Conference on Data Mining*, pages 300–309, 2011.
21. U. Kang, Jay Yoon Lee, Danai Koutra, and Christos Faloutsos. Net-ray: Visualizing and mining billion-scale graphs. In *Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference*, pages 348–361, 2014.
22. Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *J. ACM*, 51(3):497–515, 2004.
23. George Karypis and Vipin Kumar. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, 48(1):96–129, 1998.
24. B.W. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *The Bell Systems Technical Journal*, 49(2), 1970.
25. Alireza Khadivi, Ali Ajdari Rad, and Martin Hasler. Network community-detection enhancement by proper weighting. *Physical Review E*, 83(4):046104, 2011.
26. Vladimir Kolmogorov, Yuri Boykov, and Carsten Rother. Applications of parametric maxflow in computer vision. In *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
27. Danai Koutra, U Kang, Jilles Vreeken, and Christos Faloutsos. VOG: summarizing and understanding large graphs. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 91–99, 2014.
28. Kevin J. Lang and Satish Rao. A flow-based method for improving the expansion or conductance of graph cuts. In *Integer Programming and Combinatorial Optimization, 10th International IPCO Conference*, pages 325–337, 2004.
29. Jure Leskovec. http://snap.stanford.edu/snap/.
30. Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web*, pages 695–704, 2008.
31. Sungsu Lim, Seungwoo Ryu, Sejeong Kwon, Kyomin Jung, and Jae-Gil Lee. Linkscan*: Overlapping community detection using the link-space transformation. In *IEEE 30th International Conference on Data Engineering*, pages 292–303, 2014.
32. Yongsub Lim, Kyomin Jung, and Pushmeet Kohli. Energy minimization under constraints on label counts. In *11th European Conference on Computer Vision*, pages 535–551, 2010.
33. Yongsub Lim, Kyomin Jung, and Pushmeet Kohli. Efficient energy minimization for enforcing label statistics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(9):1893–1899, 2014.
34. Yongsub Lim, U. Kang, and Christos Faloutsos. SlashBurn: Graph compression and mining beyond caveman communities. *IEEE Trans. Knowl. Data Eng.*, 26(12):3077–3089, 2014.
35. Yongsub Lim, Won-Jo Lee, Ho-Jin Choi, and U. Kang. Discovering large subsets with high quality partitions in real world graphs. In *2015 International Conference on Big Data and Smart Computing*, pages 186–193, 2015.
36. Zhiyuan Lin, Nan Cao, Hanghang Tong, Fei Wang, U Kang, and Duen Horng Chau. Interactive multi-resolution exploration of million node graphs. In *IEEE VIS*, 2013.
37. Zhiyuan Lin, Nan Cao, Hanghang Tong, Fei Wang, U. Kang, and Duen Horng (Polo) Chau. Demonstrating interactive multi-resolution large graph exploration. In *13th IEEE International Conference on Data Mining Workshops*, pages 1097–1100, 2013.
38. Kiyohito Nagano, Yoshinobu Kawahara, and Kazuyuki Aihara. Size-constrained submodular minimization through minimum norm base. In *Proceedings of the 28th International Conference on Machine Learning*, pages 977–984, 2011.
39. Joseph Naor and Roy Schwartz. Balanced metric labeling. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 582–591, 2005.
40. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

41. Joel Nishimura and Johan Ugander. Restreaming graph partitioning: simple versatile algorithms for advanced balancing. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1106–1114, 2013.
42. Usha N. Raghavan, Reka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Reveiw E*, 76(3), 2007.
43. Venu Satuluri and Srinivasan Parthasarathy. Scalable graph clustering using stochastic flows: applications to community discovery. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 737–746, 2009.
44. Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27 – 64, 2007.
45. Arunabha Sen, Haiyong Deng, and Sumanta Guha. On a graph partition problem with application to vlsi layout. *Inf. Process. Lett.*, 43(2):87–94, 1992.
46. Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.
47. Georgos Siganos, Sudhir Leslie Tauro, and Michalis Faloutsos. Jellyfish: A conceptual model for the as internet topology. *Communications and Networks, Journal of*, 8(3):339–350, 2006.
48. Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 81–90, 2004.
49. Isabelle Stanton. Streaming balanced graph partitioning algorithms for random graphs. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1287–1301, 2014.
50. Isabelle Stanton and Gabriel Kliot. Streaming graph partitioning for large distributed graphs. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1222–1230, 2012.
51. Charalampos E. Tsourakakis, Christos Gkantsidis, Bozidar Radunovic, and Milan Vojnovic. FENNEL: streaming graph partitioning for massive scale graphs. In *Seventh ACM International Conference on Web Search and Data Mining*, pages 333–342, 2014.
52. Johan Ugander and Lars Backstrom. Balanced label propagation for partitioning massive graphs. In *Sixth ACM International Conference on Web Search and Data Mining*, pages 507–516, 2013.
53. Song Wang and Jeffrey Mark Siskind. Image segmentation with ratio cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):675–690, 2003.
54. Ning Xu, Lei Chen, and Bin Cui. Loggp: A log-based dynamic graph partitioning method. *PVLDB*, 7(14):1917–1928, 2014.
55. Ning Xu, Bin Cui, Lei Chen, Zi Huang, and Yingxia Shao. Heterogeneous environment aware streaming graph partitioning. *IEEE Trans. Knowl. Data Eng.*, 27(6):1560–1572, 2015.
56. Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *12th IEEE International Conference on Data Mining*, pages 745–754, 2012.
57. Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Sixth ACM International Conference on Web Search and Data Mining*, pages 587–596, 2013.
58. Tianbao Yang, Rong Jin, Yun Chi, and Shenghuo Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 927–936, 2009.