Journal of Network and Computer Applications 86 (2017) 59–71

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Hypergraph partitioning for social networks based on information entropy modularity[☆]

Wenyin Yang[a,b], Guojun Wang[a,c,*], Md Zakirul Alam Bhuiyan[a,d], Kim-Kwang Raymond Choo[e,f]

[a] School of Information Science and Engineering, Central South University, Changsha 410083, China
[b] School of Electronic and Information Engineering, Foshan University, Foshan 528000, China
[c] School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China
[d] Department of Computer and Information Sciences, Fordham University, New York, NY 10458, USA
[e] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631, USA
[f] School of Information Technology & Mathematical Sciences, University of South Australia, Adelaide, SA 5095, Australia

ARTICLE INFO

Keywords:
Scale-free network
Social network partitioning
Hypergraph partitioning
Information entropy
Modularity

ABSTRACT

A social network is a typical scale-free network with power-law degree distribution characteristics. It demonstrates several natural imbalanced clusters when it is abstracted as a graph, and expands quickly under its generative mechanism. Hypergraph is superior for modeling multi-user operations in social networks, and partitioning the hypergraph modeled social networks could ease the scaling problems. However, today's popular hypergraph partitioning tools are not sufficiently scalable; thus, unable to achieve high partitioning quality for naturally imbalanced datasets. Recently proposed hypergraph partitioner, `hyperpart`, replaces the balance constraint with an entropy constraint to achieve high-fidelity partitioning solutions, but it is not tailored for scale-free networks, like social networks. In order to achieve scalable and high quality partitioning results for hypergraph modeled social networks, we propose a partitioning method, `EQHyperpart`, which utilizes information-Entropy-based modularity Q value (EQ) to direct the hypergraph partitioning process. This EQ considers power-law degree distribution while describing the “natural” structure of scale-free networks. We then apply simulated annealing and introduce a new definition of hyperedge cut, micro cut, to avoid the local minima in convergence of partitioning, developing `EQHyperpart` into two specific partitioners, namely: `EQHyperpart-SA` and `EQHyperpart-MC`. Finally, we evaluate the utility of our proposed method using classical social network datasets, including Facebook dataset. Findings show that `EQHyperpart` partitioners are more scalable than competing approaches, achieving a tradeoff between modularity retaining and cut size minimizing under balance constraints, and an auto-tradeoff without balance constraints for hypergraph modeled social networks.

1. Introduction

Social networks, a source of big data (Saleh et al., 2013), face complicated scalability challenges in data management, due to the large numbers of data nodes and complex data relationships. Such challenges are compounded in online social networks. Scaling out is generally employed to address scaling problems by deploying data over several servers, such as virtual machines in the cloud, and partitioning the query loads between these servers (Pujol et al., 2010). Hash-based partitioning approach is widely used in commodity systems, such as dynamo and Cassandra. However, neglecting data relations still leads to heavy query loads (Pujol et al., 2009; Yuan et al., 2012). To improve partitioning quality, approaches based on modeling social network structure and user interactions have been proposed (Pujol et al., 2010; Curino et al., 2010; Turk et al., 2014). Generally, social networks are abstracted by graphs, and the query loads partitioning problem can be reduced to graph partitioning problems.

Graph partitioning comprises dyadic graph partitioning (Arora et al., 2009; Karypis and Kumar, 1998) and hypergraph partitioning (Karypis et al., 1999; Karypis and Kumar, 1999; Yaros and Imielinski, 2013). While there exist numerous partitioning solutions based on usual graphs, or dyadic graphs, there has been limited focus on the use of hypergraph partitioning. Recent studies have argued that hypergraphs are more powerful than dyadic graphs for modeling groups in

[☆] This is an extended version of the conference paper (Yang et al., 2015), with more than 50% new content.

[*] Corresponding author at: School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China.

E-mail addresses: cswyyang@163.com (W. Yang), csgjwang@gmail.com (G. Wang), mbhuiyan3@fordham.edu (M.Z.A. Bhuiyan), raymond.choo@fulbrightmail.org (K.-K.R. Choo).

http://dx.doi.org/10.1016/j.jnca.2016.10.002
Received 16 May 2016; Received in revised form 25 September 2016; Accepted 4 October 2016
Available online 06 October 2016

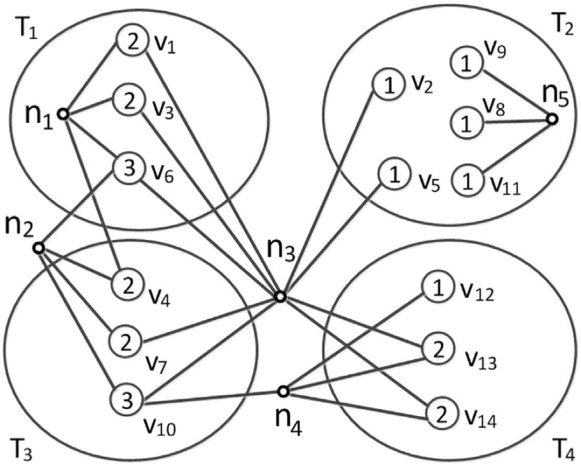

Fig. 1. An example of hypergraph partitioning. A hypergraph with 14 vertices and 5 hyperedges is partitioned into 4 parts, leading to a cutsize of 4.

many domains (Turk et al., 2014; Heintz and Chandra, 2014; Yang and Wang, 2015; Guzzo et al., 2014), including partitioning and other operations on complex networks (Turk et al., 2014; Yang and Wang, 2015). Therefore, in this paper, we mainly focus on the partitioning of hypergraph modeled social networks. For instance, social network users could be denoted by vertices of a hypergraph, and the multi-user operation, such as multi-gathering of data from multiple social network (e.g., Facebook) friends in a single operation, could be modeled via a hyperedge in the hypergraph. Inter-server query costs of multi-way interactions are defined as the cut cost of hypergraphs. The hypergraph partitioning problem is to partition the vertices of a hypergraph into k disjoint nonempty equal-size partitions, such that the number of the hyperedges connecting vertices in different partitions (called the cut), or the cut size, is minimized. An example of undirected hypergraph partitioning is depicted in Fig. 1. Here, a hypergraph $H = (V, N)$ with vertex set V={v_1, v_2,..., v_{14}} and hyperedge set N={n_1, n_2,..., n_5} is partitioned into four parts, namely: T={T_1, T_2,..., T_4}, where the cut size is 4 since hyperedges n_1, n_2, n_3, and n_4 are in cut state. The values in vertex circles denote the weight of vertices, assigned as the degree of vertex in this hypergraph.

Min-cut graph partitioning (Karypis and Kumar, 1998; Karypis et al., 1999; Karypis and Kumar, 1999) and modularity optimization (Aaron et al., 2004; Kwak et al., 2009; Newman, 2006; Blondel et al., 2008) are popular network partitioning solutions. Min-cut graph partitioning algorithms are generally designed to produce equal sized partitions such that the number of inter-partition edges or hyperedge is minimized, and modularity optimization algorithms aim to maximize the modularity (see Pujol et al., 2010; Newman, 2006), a.k.a. Q value, which is the most common method and measure designed to estimate the strength of dividing a network into modules. Networks with a high Q value have dense connections between the nodes within modules, but sparse connections between nodes in different modules. However, the social network is a typical scale-free network characterized by power-law distribution of degrees, community structures and quick expansion under its generative mechanism (Barabàsi and Bonabeau, 2003), and existing approaches are generally inadequate for the hypergraph partitioning of it. For example,

- Community structures of scale-free networks are not fully considered. Popular min-cut hypergraph partitioning tools, such as hMETIS (Karypis et al., 1999) and khMETIS (Karypis and Kumar, 1999), provide load balancing while performing min-cut partitioning. However, min-cut and load balance are not necessarily the only constraints for all applications (Yaros and Imielinski, 2013), including complex networks partitioning. Social networks possess typical "quasi-balanced" datasets with natural clusters, and the cluster structures are foundation for the expansion of networks; thus, it is important to also consider community structures to ensure scalability.
- Modularity optimization algorithms are not well adapted to changing graph structures when social networks grow rapidly. Specifically, existing partitioning algorithms based on community detection are sensitive to graph structure (Kwak et al., 2009), and the node placement solutions are unstable. The statistical feature of complex systems could be used to model the modularity; thus, mitigating this sensitivity (Deng et al., 2009).
- Most hypergraph partitioners do not consider the power-law distribution of degrees. For example, the partitioners hyperpart (Yaros and Imielinski, 2013) uses information entropy as a constraint for hypergraph partitioning. Compared to equal-sized partitioners, such as hMETIS and khMETIS, hyperpart is able to produce high-fidelity partitions for imbalanced datasets. In other words, if the dataset is composed of different sizes of clusters due to natural characteristics, then the fixed-part partitioning result could reserve the natural group members within a part to the utmost extent, even though it is imbalanced in size. However, hyperpart is not suitable for scale-free networks, including social networks, because it uses an information entropy that treats every node equally. This does not allow one to reflect the degree distribution feature of scale-free networks.

To address the above gaps, we proposed the use of a scale-free network featured information entropy in our previous work (Yang et al., 2015), which considers the vertex degrees rather than the vertex amounts of communities in the hyperpart to reflect the energy distribution in the network system. Since the distribution rules of entropy coincide with the vertex degree distribution, we used Q values computed based on the information entropy to depict the community structures of scale-free networks, and to direct the min-cut hypergraph partitioning process, producing high-fidelity partition results. The statistic characteristics of this information entropy could overcome the drawback due to unstable assignment. The preliminary experiment findings detailed in Yang et al. (2015) demonstrate that this partitioning can achieve tradeoff between modularity retaining and cut size minimizing of hypergraph modeled complex networks.

We also remark that a partition method designed to get tradeoff among load balance, minimum cut cost, and modularity, is more scalable in the long run than existing min-cut hypergraph partitioning algorithms (e.g., hMETIS and khMETIS). Therefore, while partitioning, we seek to address the three above mentioned factors to ensure scalability. In addition, to avoid the local minima in the search for optimum values of Q value and cut cost, simulated annealing algorithm and microcut are adopted for optimization. We propose *micro cut* as a measure for hypergraph cut state, which provides microhints to hypergraph partitioners for facilitating a view into the future moves.

The contributions of this paper are two-fold.

(1) We put forward a social network partitioner, EQHyperpart, which utilizes information Entropy based modularity Q (EQ) to direct the low cost partitioning process. It considers modularity maintaining, cut size minimizing and balance factors during partitioning to improve scalability.
(2) We propose a new hyperedge cut metric, *micro cut*, and optimize the partitioning quality of EQHyperpart by adopting simulated annealing and micro cut, forming two variant partitioners, namely: EQHyperpart-SA and EQHyperpart-MC.

We then evaluate the partition quality of our proposed partitioning algorithm with competing hypergraph partitioners, such as hMETIS, khMETIS, and hyperpart. Findings demonstrate that EQHyperpart is more scalable, and EQHyperpart achieves an auto-tradeoff between cut size, modularity, and balance level.

The organization of the remainder of this paper is as follows. In Section 2, we introduce background knowledge and related work. Next, we present the definition of scale-free information entropy and EQ in Section 3. Section 4 describes our proposed EQHyperpart algorithm and micro cut. Section 5 presents our evaluation findings. Finally, in Section 6, we conclude this paper and outline future work.

2. Background and related work

2.1. Social networks modeling

A social network structure comprises a set of social actors, sets of dyadic ties, and other social interactions between actors. Existing related research on social networks could be classified into two categories, namely: one is to analyze the social network structures and study the theories explaining the patterns observed in these structures (Jin et al., 2013; Wilson et al., 2011; Jiang et al., 2010); and the other is to solve existing problems based on social networks (Wang et al., 2012; Ma et al., 2015; Carullo et al., 2015; Zaerpour et al., 2009; Carullo et al., 2013). Modeling the social network as a graph, namely social graph, is an effective and widely used mathematical tool in existing studies.

Generally, social networks could be modeled as undirected graphs or directed graphs according to their properties (Jin et al., 2013). Classical undirected social graphs consist of friendship and interaction graphs (Wilson et al., 2011). The former indicates logical friendship between social network users, and the latter reflects visible interaction in social network, such as wall posts and other messages. Typical directed social graphs include latent graph (Jiang et al., 2010) and following graph. The former models passive actions of social network users (e.g., profile browsing), while following graph reveals the subscription relationship in social networking applications such as Twitter.

In addition, social networks could be categorized into uniform and weighted models. Uniform social network model treats every edge equally, i.e. users access any of their friends' data in equal probability. Weighted social network models assign diverse meanings to edges based on the underlying motivations.

Moreover, social graph could be divided into dyadic social graph and social hypergraph. Dyadic social graphs only reflect the two-way relations of social network users via edges, while social hypergraphs can capture multi-way relations, such as multi-way interactions, multi-user static relationships, and multi-user dynamic operations, via hyperedges (Turk et al., 2014). Hypergraphs are generally more effective in modeling groups, but the concept of a group depends on the social networking context (Heintz and Chandra, 2014). For example, one hyperedge in Fig. 1 could represent one group of social network users who interact closely via recent wall postings, or those have the same group label, like “colleague”, “classmate”, etc.

2.2. Information entropy

Entropy was first introduced as a thermodynamic concept in 1872 (Boltzmann, 2003), and subsequently used in information theory (Shannon, 1948). The macro significance of entropy is a measure of the uniformity of system energy distribution, representing the object state as stable or not stable. The information entropy (also known as Shannon entropy) is used to characterize the uncertainty about the source of information, and increases with more sources of greater randomness. The source can be characterized by the probability distribution of its samples. When taken from a finite sample, Shannon defined the entropy H of a discrete random variable X with possible values $\{x_i | i = 1, \ldots, N\}$ and probability mass function $P(x_i)$ as (1), where b is the base of the logarithm, and the unit of entropy is Shannon (defined by IEC 80000-13) for b=2. Shannon entropy has been applied in modeling of complex systems (Popescu et al., 2013) and Big Data applications (Popescu, 2016):

$$H(X) = -\sum_{i}^{N} P(x_i)\log_b P(x_i) \tag{1}$$

2.3. Hypergraph partitioning

A hypergraph is the generalization of a graph, where a hyperedge, a.k.a net, can connect a group of vertices. The concept of undirected hypergraph was first given by C. Berge in 1970s (Berge, 1985), and directed hypergraph theories were subsequently developed. Partitioning is an important concept in both graph and hypergraph theories. Given a hypergraph H, k-way partitioning assigns vertex set V of H to k disjoint nonempty partitions, aiming to minimize a given cost function of such an assignment (see Fig. 1).

Pins allocation in very-large-scale integration (VLSI) design and computation load distribution in parallel computing are typical applications of hypergraph partitioning, in which the number of parts are known *a priori*, and balance constraint are preferred (Karypis et al., 1999). Hypergraph partitioning of social network could be adopted for user data allocation among fixed number of servers to improve inter-user data access performance (Turk et al., 2014; Yang and Wang, 2015).

With certain constraints such as balance, the problem of optimally partitioning a hypergraph is known to be NP-hard (Borndörfer and Heismann, 2015). There has been a number of heuristic algorithms with near-linear runtime presented in the literature. Kerninghan-Lin (KL) algorithm (Kernighan and Lin, 1970) is the pioneer of both graph partitioning and community structure, which enables vertex swaps in an equal-sized bisection to reduce its cut. The strict equipartition requirement is relaxed to a balance constraint by the Fiduccia-Mattheyses (FM) algorithm (Fiduccia and Mattheyses, 1982), which introduces single-vertex move rules and enables k-way partitioning based on recursively bi-partitioning. Later, a direct k-way partitioning based on FM algorithm is put forward (Fiduccia and Mattheyses, 1982). However, execution time increases as graph size grows. Karypis and Kumar (1998) proposed a multilevel framework for hypergraph partitioning, including coarsening, partitioning and uncoarsening phases. FM and / or its derivatives are applied in the middle phase. Partitioning tools, hMETIS (Karypis et al., 1999) and khMETIS (Karypis and Kumar, 1999), were then developed to implement the multilevel framework via recursive bisection and direct k-way approaches, respectively. Following KL algorithm (Kernighan and Lin, 1970), several direct k-way partitioning methods based on FM algorithm (Fiduccia and Mattheyses, 1982) and a multilevel framework (Karypis and Kumar, 1998) for hypergraph partitioning are proposed. For example, hMETIS and khMETIS are designed to implement the multilevel framework via recursive bisection and direct k-way approaches, respectively. UMPa (Deveci et al., 2015) is a multi-objective hypergraph partitioner also using multilevel. This framework is effective in reducing both execution time and cut size, but is limited to low levels of imbalance.

In hMETIS, unbalance factor (UBfactor) denoted by b is used to control deviation of each part size, within the upper bounce $n\cdot((50 + b)/100)^{\log_2 k}$ and lower bounce $n\cdot((50 - b)/100)^{\log_2 k}$, where n is the number of vertices and k is the number of parts. While in khMETIS, the heaviest part size is up to $(1 + b/100)\cdot(n/k)$, where b is also the unbalance factor, n is the number of vertices and k is the number of parts, indicating that the weight of the heaviest partition should not be more than b% greater than the average weight. For example, in Fig. 1, the unbalance factor is b=10 under hMETIS metric while b=43 under khMETIS metric, assuming that each having unit vertex weight.

Yaros proposed the use of information entropy in (2) as an imbalance constraint (Yaros and Imielinski, 2013), such that it enables the proposed partitioner, hyperpart, to find high-fidelity solutions for given levels of imbalance. In other words, the partitioning result is more similar to the actual partition result. However, $P(x_i)$ in (1) is assigned as $|V_i|/|V|$ in (2), where k denotes the number of parts, and $|V_i|$ denotes the number of vertex in part i. This implies every node is treated equally and the node importance is ignored, which can be reflected by node degrees. Therefore, such an approach cannot be applied to a scale-free complex network directly, which is characterized by the power-law degree distribution:

$$E_u = -\sum_{i}^{k} \frac{|V_i|}{|V|} \log_b\left(\frac{|V_i|}{|V|}\right) \tag{2}$$

Hyperedge cut is a standard cost function measuring the partition quality. Cut size refers to the number of nets in cut state, that is spanning more than one partition. In addition, the number of parts a cut net spans is referred to as the Sum of External Degrees (SOED). The similar $K-1$ measure has penalty of parts spanned outside the base part. For example, in Fig. 1, the cut sizes of net n_2 and n_3 are both 1, but SOED of n_2 and n_3 are 2 and 4, respectively. Thus, the $K-1$ cut sizes of them are 1 and 3, respectively. hMETIS aims to minimize cut size, while khMETIS is designed to optimize $K-1$ cut size or SOED. The typical objective of hypergraph partitioning is to minimize inter-partition communication, and in this context, $K-1$ cut size is the most suitable metric.

2.4. Community detection

Complex networks are naturally divided into subgroups. Originating from the small-world concept, the mutuality of ties, the frequency of ties among subgroup members, or the closeness of subgroup members could lead to a community within complex network. The community structure detector seeks to find those subgroups of complex networks. Modularity is a widely used metric to evaluate the detection capabilities of complex networks. The value of modularity, i.e. Q value, is defined in (3), where k denotes the number of communities, e_{ii} is the ratio of the number of edges inside community i to the total number of edges in the whole network, and e_{ij} is the ratio of the number of edges between community i and j to the total number of edges in the whole network. To increase Q value, inner-connection should be higher and inter-connection should be lower:

$$Q = \sum_{i}^{k}\left(e_{ii} - \left(\sum_{j}^{k} e_{ij}\right)^2\right) \tag{3}$$

Current detection methods include minimum-betweenness (Newman and Girvan, 2004), clustering centrality (Santo et al., 2004), random walk (Pons and Latapy, 2005), eigenvectors of matrices (Newman, 2006), target function optimization (Newman, 2004), and methods based on density of edge connection (Palla et al., 2005). Detection speed (He and Chen, 2015) and detecting overlapping community (Eustace et al., 2015) are the focus of recent research in recent times, and a number of information-theoretic methods have been introduced. Examples include detection methods based on information-theoretic entropy (Deng et al., 2009), minimum description length (MDL) mutual information (Martin and Bergstrom, 2007), and information bottle (Shen et al., 2008). These methods are used in dyadic graph modeled complex networks. A review of the literature suggests that community structure detection in hypergraph modeled networks remains a topic that is understudied. One of the few work on hypergraph modeled networks is that of Xie (Zheng et al., 2012), who proposed to view hyperedges as vertices and establish a network for hyperedges by their similarity. This modularity method is then applied to find communities for document words association.

A key limitation of community structure detection methods for both usual graphs (Aaron et al., 2004; Deng et al., 2009; Newman, 2004; He and Chen, 2015; Eustace et al., 2015) and hypergraph (Zheng et al., 2012) is the possibility that no good division of the network exists. However, the goal of graph partitioning is to find the best division of the network regardless of whether a good division exists (Newman, 2006). Despite this limitation, community structure detection methods could be adapted to find a good division.

3. EQ in hypergraph modeled social networks

In this section, we describe the social networks modeled using hypergraph in Section 3.1 prior to introducing the definitions of E (i.e. information entropy reflecting characteristics of scale-free networks) and EQ (i.e. modularity based on the entropy in hypergraph) in Sections 3.2 and 3.3, respectively.

3.1. Hypergraph modeled social networks

In our work, social networks are modeled using hypergraph, i.e. vertices in hypergraph may represent the social network users, and hyperedges may indicate the relationships among users. Diverse context could be assigned to weights of vertices or hyperedges, such as workload and access cost. Hub nodes with higher degree in social network may indicate owning more friends or being more active. New nodes with little degree tend to link the hub nodes according to the generation mechanism of social networks. Interconnections within a group become closer than those among groups, forming community structures gradually. Therefore, the degree of nodes, or generally, the weight of vertices, implies the probability of joining a community. Hypergraph partitioning is one of the key approaches with extensive application in mining of subgroups. Minimum cut of hypergraph is more intuitive and effective than that of dyadic in some cases (Yang and Wang, 2015).

3.2. Definition of entropy for social networks

In scale-free complex networks, including social networks, the degree distribution follows the power-law, indicating that the energy is unevenly distributed in these networks, which is also known as *nonhomogeneity*. The latter property demonstrates that a scale-free network is a kind of *ordered network*, while scaled network, such as random network, belongs to *disordered network*. Entropy is used to quantify the property of *order*.

Existing studies on *Barabàsi–Albert* (BA) model (Barabàsi et al., 2000) suggest that the scale-free network is caused by the growing and preferential attachment mechanisms that the new nodes preference to connect with hub nodes. To be specific, when a new node joins a scale-free network, the probability of node i chosen to be connected by the new node is decided by the degree of node i. Therefore, the degree of a node can be a baseline metric to reflect the importance of a vertex in a network. Therefore, the importance of vertex i can be defined by:

$$I_i = \frac{d_i}{\sum_{i=1}^{N} d_i}, \tag{4}$$

where N is the number of vertices in the network, and d_i is the degree of vertex i. We assume that $d_i > 0$; thus, $I_i > 0$. The importance of vertices is different in an ordered network. In a scaled network, however, the importance of vertices is roughly equivalent, and this is why a scaled network is called a disordered network. To quantitatively measure the order, network structure entropy is defined by:

$$E = -\sum_{i=1}^{N} I_i \log I_i \tag{5}$$

It is trivial to prove that when the network is completely uniform (i.e. $I_i = 1/N$), E reaches the peak. When all the vertices connect to one hub vertex, say the first vertex (i.e. $d_1 = N-1$, $d_j = 1\ (j > 1)$), E falls to the bottom, because the network is the most nonuniform.

3.3. Definition of Q value based on entropy

As mentioned in Section 3.1, energy distributes unevenly in an ordered scale-free network, which is associated with community structures. It can be inferred that the energy concentrates inside the community, leading to uneven distribution of the energy and resulting in the (obvious) community structure. We suggest using Entropy-based Q value (EQ) to describe the community structure property for a scale-free network. The denser within communities and the sparser among communities, the greater Q value will become and the more obvious the community structure will be. Thus, we define EQ as the difference between Community Structure Entropy (CSE) and Inter-Community Entropy (ICE).

3.3.1. Community Structure Entropy (CSE)

Based on the preferential attachment mechanisms of scale-free network, it can be inferred that when the community structure is formed, the new node chooses a community to join that also follows the preferential mechanisms. In other words, the importance, say the quantity, of a community determines the probability of new node's accession.

Let $Y = (y_1, y_2, \ldots, y_{|y|})$ be a variable of community, and $X = (x_1, x_2, \ldots, x_{|x|})$ be a variable of node. We define CSE as a conditional entropy $H(Y|X)$ by (6) to measure the uncertainty or the disorder situation of X, on the condition of already existing Y, where N is the total number of vertices, M is the total number of communities, and $P(x_i, y_j) = P(y_j|x_i)P(x_i)$ is the joint probability.

$$H(Y|X) = -\sum_{i=1}^{N}\sum_{j=1}^{M} P(x_i, y_j)\log P(y_j|x_i) = -\sum_{i=1}^{N}\sum_{j=1}^{M} P(y_j|x_i)P(x_i)\log P(y_j|x_i) \tag{6}$$

In the community structure entropy, $P(x_i)$ denotes the importance of a node i, which is evaluated by (4), and $P(y_j|x_i)$ represents the probability that community j contains node i. As we have previously discussed, the probability that a node joins a community is determined by the importance of the community. Therefore, $P(y_j|x_i)$ is defined as m_j/N, the proportion of node numbers of community j to the total node numbers of the whole network. Thus, CSE is calculated by (7), where Z_{ij} is an assignment matrix. If node i is assigned to community j, then $Z_{ij} = 1$, otherwise 0:

$$E_{CS} = -\frac{\sum_{i=1,Z_{ij}=1}^{N}\sum_{j=1}^{M} d_i * \left(\frac{m_j}{N}\right) * \log\left(\frac{m_j}{N}\right)}{\sum_{i=1}^{N} d_i} \tag{7}$$

3.3.2. Inter-Community Entropy (ICE)

ICE refers to the uncertainty among communities in scale-free networks. We focus on the ICE of hypergraph modeled scale-free network in this paper. It is well known that multilevel partitioning framework is popular in many hypergraph partitioners, which comprises three phases, namely: coarsening, partitioning, and uncoarsing.

Hypergraph is coarsened by merging vertices and / or edges using some heuristics algorithms. In an extreme case, a vertex-level hypergraph could be coarsened to a community-level one. Inspired by this, the communities can be viewed as super-vertices after coarsening, and the association among communities can be considered to be the cut hyperedges among communities. Hence, ICE can be defined as (8), where C is the current $K-1$ cut size of the hypergraph modeled network and M is the total number of communities:

$$E_{IC} = -C*\left(\frac{1}{M}\right)*\log\frac{1}{M} \tag{8}$$

3.3.3. Entropy-based Q value

According to the idea of modularity introduced in Section 2.4, based on the definition of CSE and ICE, the entropy-based Q value EQ is defined by:

$$EQ = E_{CS} - (E_{IC})^2 \tag{9}$$

4. Hypergraph partitioning based on EQ

In this section, we first discuss the basic idea of `EQHyperpart` algorithm. Then, we present two optimization methods for the local minima avoidance. Here, Section 4.1 introduces the optimized algorithm by adopting Simulated Annealing (SA), named `EQHyperpart-SA`, in detail. The definition of micro cut and the optimization approach based on micro cut, forming `EQHyperpart-MC`, are discussed in Section 4.2.

4.1. Algorithm

Note that, in our previous work (Yang et al., 2015), we presented the basic idea of a hypergraph partitioning algorithm based on EQ, named as `EQHyperpart`, which is designed based on the idea of `khMETIS`, the k-way counterpart of `hMETIS`, but the minimum cutsize metric is replaced by the maximum EQ.

According to the single-vertex move rules, in each iteration, `EQHyperpart` selects the vertex with the highest gain and the highest delta-EQ-value to move and freeze, until all the vertices are frozen. Then, `EQHyperpart` rollbacks to the point with the highest EQ value and unfreezes all vertices and starts the new move operations again. The best partition solution search terminates when the EQ does not increase in the last round or the iteration number exceeds a threshold.

This basic partitioning solution, however, is easy to converge to a local optima in the solution space. We choose SA, a probabilistic technique for approximating the global optimum of a given function, to solve this problem. SA starts from an admissible solution of the problem (denoted by S in (10)). The search strategy in the neighbourhood of such a solution will be more intensive in the more promising regions, penalizing the searches that move far from these regions but accepting with certain probability P_s, defined in (10), and searches that worsen the solution (denoted by S' in (10)). Here, the function $y = E(x)$ refers to the energy level of the given solution, t is the temperature and k is a constant. The temperature t, which is the acceptance criterion governed by a random number generator and a control parameter (a.k.a cooling ratio r), slowly modifying its value by $t = r*t$ $(0 < r < 1)$, drives the system towards the final solution, which corresponds to a local minimum of the objective function (Selim and Alsultan, 1991):

$$P_s = k*\exp\left(-\frac{E(S') - E(S)}{t}\right) \tag{10}$$

We adapt SA in `EQHyperpart` to search for the highest EQ value. When EQ value does not increase in the last round, the algorithm rolls back to the point where the second highest EQ value is seen with the probability defined by (10). The `EQHyperpart` based on SA (`EQHyperpart-SA`) comprises a sequence of operations depicted in Algorithm 1.

Firstly, after randomly distributing the pins of vertices, we compute the EQ value of the network, and the possible gains for each vertex (line 1). At the beginning of each iteration, we unfreeze all vertices to be ready for move (line 4). Then, the algorithm enters the inner while loop (lines 5–14). In this loop, we select the best move according to the compound conditions, including gain value, incremental EQ value, and the unbalance ratio (lines 6–7). After performing the move of vertex v from *FP* part to the *TP* part and locking vertex v (line 8), we update the gain values and pin distributions in an incremental manner (line 9) and compute the new EQ value of the whole network (line 10). Then, the EQ_{new} is assigned the largest EQ within this loop (line 11-13) and the corresponding move index is recorded. The inner while loop ends when the whole hypergraph is frozen, and we compare the highest EQ value from this loop with the EQ value from the last one. If EQ_{new} is no higher than EQ_{old}, we unwind sequences of executed moves back to the point where the partition with EQ_{new} is seen, with certain probability, and assign the EQ_{new} to EQ_{old} (lines 15-19). In other case, we rollback to the point when the highest EQ value occurred (lines 20-26). The temperature is updated after the rollback (line 27).

If the number of iteration exceeds the predetermined value, or we achieve the same highest EQ value for certain times, indicating that no increase of EQ is possible for any further move, the outer loop terminates (line 28). Finally, the partitioned result is output.

Besides partitioning under certain balance constraint, EQHyperpart-SA enables maximum modularity partitioning without any constraint. In other words, we can ignore the unbalance constraints in line 6 of Algorithm 1, and achieve a relative reasonable partitioning result, which automatically obtains a tradeoff among the lowest cut, modularity, and balance level. Findings outlined in Section 5 verify the correctness of this function.

Algorithm 1. Algorithm of EQHyperpart-SA.

```
Input: hypergraph $HG = (V, N)$, part number $K$, unbalance factor $\varepsilon$, temperature $t$, cooling ration $r$, iteration control parameter $c$, $l$
Output: partitioned result P=($P_1$, $P_2$,..., $P_K$)
1: Initialize pin distributions, compute EQ value and gains for all possible moves from each vertex's current part to $(K-1)$ other parts.
2: Set $EQ_{new} = EQ$, $EQ_{old} = EQ$
3: repeat
4:   Unfreeze all vertices.
5:   while there is any valid move do
6:     $HGainList$ ← Select the highest gain moves that do not violate unbalance constraints.
7:     $BestMove(v, FP, TP)$ ← Select the highest delta-EQ-value move in $HGainList$.
8:     Move vertex $v$ from $FP$ part to $TP$ part, and freeze $v$.
9:     Update the gains of unfreezed neighbours of $v$ and the pin distributions.
10:    Update $EQ$.
11:      if $EQ > EQ_{new}$ then
12:        Set $EQ_{new} = EQ$
13:      end if
14:  end while
15:  if $EQ_{old} > EQ_{new}$ then
16:    Draw a random number $y \in (0, 1)$
17:    if $y < \exp(-(E(EQ_{new}) - E(EQ_{old}))/t)$ then
18:      Rollback to the point when the non-highest EQ value $EQ_{new}$ is seen.
19:      Set $EQ_{old} = EQ_{new}$
20:    else
21:      Rollback to the point when the highest EQ value $EQ_{old}$ is seen.
22:    end if
23:  else
24:    Rollback to the point when the highest EQ value $EQ_{new}$ is seen.
25:    Set $EQ_{old} = EQ_{new}$
26:  end if
27:  $t = r*t$
28: until $EQ$ does not increase for $c$ times or iteration number exceeds threshold $l$.
```

4.2. Micro cut

To speed up the partition of hypergraph, k-way FM-based partitioners, such as khMETIS, move vertex based on the gain which is defined by the cut cost benefits affected by critical nets. Usually, the cut size or $K-1$ cut size is used as the gain measure. It leads to the well-known issue of convergence to local optima for this kind of partitioners.

Take net n_3 in Fig. 1 as an example. This net owns 9 vertex pins, and spans four parts initially; thus, the $K-1$ cut size of n_3 is 3. It is clear that net n_3 is not critical to any part, because no move operation performed on any pin of net n_3 can change its cut-state (i.e. the $K-1$ cut size). Consider all vertex pins of n_3, the move gains measured by $K-1$ cut size are zero (the best gain possible), from the located part to any other parts. In other words, the partitioner cannot see any benefit in moving any pin. In this case, the partitioner is stuck at this local optima, and makes the best move randomly.

To reduce the nearsightedness of partitioners, we define a measure for cut state, namely *micro cut* (see 11). This metric provides partitioners with some micro hints to inform future moves:

$$\sum_{e\in E}\left[1-\sum_{\pi\in\Pi}\left(\frac{|\pi\cap e|}{\sum_{\pi\in\Pi}|\pi\cap e|}\right)^{(1+\alpha)}\right] \tag{11}$$

Here, E is the set of nets, Π is the set of partitions, and $\alpha > 0$. $|\pi \cap e|$ denotes the number of pins located in part π owned by the net e.

Fig. 2 illustrates the move of net n_3 under the indication by micro cut. With the micro cut, there is a positive gain in moving v_2. The microcut sizes of n_3 listed in Table 1 show that moving v_2 from parts T_2 to T_1 produces the highest gain, and guide the partitioner to take this move. With this move, net n_3 becomes critical to part T_2. At this point, the partitioner is "enlightened", and moves v_5 to part T_1, reducing the $K-1$ cut size from 3 to 2.

In essence, the micro cut metric encourages moving pins from a part with less pins to one with more pins owned by the same net, leading to the decrease of final gains in the $K-1$ cut size metric. In view of this, the micro cut could be applied when there is a tie in highest gain move candidates, or take the place of $K-1$ cut size when calculating the move gains. On the basis of EQHyperpart-SA, we choose the former usage of micro cut, forming a new version of EQHyperpart, named EQHyperpart-MC (EQHyperpart based on Micro Cut).

As shown in Algorithm 2, EQHyperpart-MC combines two cut size metrics to choose proper moves (line 6–7) for further local optima avoidance. In the following sections, the term *EQHyperpart* refers to two variant partitioners, namely: EQHyperpart-SA and EQHyperpart-MC.

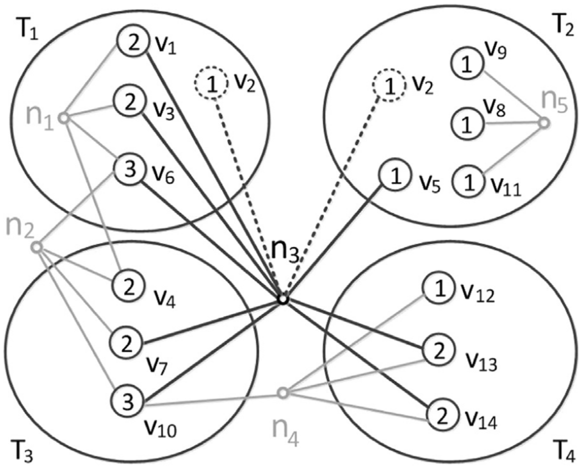

Fig. 2. Move of net n_3 by the hints of move gains under microcut metric.

Table 1
Move gain calculation example. (Take vertex v_2 as an example, with $\alpha = 1.1$.)

Type of cut size	T_2	$T_2 \rightarrow T_1$		$T_2 \rightarrow T_3$		$T_2 \rightarrow T_4$	
	Before move	After move	Move gain	After move	Move gain	After move	Move gain
$K - 1$ Cut	3	3	0	3	0	3	0
Micro Cut	0.1278	0.1186	**0.0092**	0.1223	0.0055	0.1223	0.0055

Algorithm 2. Algorithm of EQHyperpart-MC.

```
Input: hypergraph $HG = (V, N)$, part number $K$, unbalance factor $\varepsilon$, temperature $t$, cooling ration $r$, iteration control parameter $c$, $l$
Output: partitioned result P=($P_1$, $P_2$,..., $P_K$)
Initialize pin distributions, compute EQ value and gains based on $K - 1$ cut size for all possible moves from each vertex's current part to $(K - 1)$ other parts.
Set $EQ_{new} = EQ$, $EQ_{old} = EQ$
repeat
    Unfreeze all vertices.
    while there is any valid move do
        $HGainList$ ← Select moves with the highest gain based on $K - 1$ cut size under unbalance constraints.
        $BestMove(v, FP, TP)$ ← Select move with the highest gain based on micro cut size in $HGainList$.
        Move vertex $v$ from $FP$ part to $TP$ part, and freeze $v$.
        Update the gains of unfreezed neighbours of $v$ and the pin distributions.
        Update $EQ$.
        if EQ > EQ$_{new}$ then
            Set EQ$_{new}$ = EQ
        end if
    end while
    if $EQ_{old} > EQ_{new}$ then
        Draw a random number $y \in (0, 1)$
        if $y < \exp(-(E(EQ_{new}) - E(EQ_{old}))/t)$ then
            Rollback to the point when the non-highest EQ value $EQ_{new}$ is seen.
            Set $EQ_{old} = EQ_{new}$
        else
            Rollback to the point when the highest EQ value $EQ_{old}$ is seen.
        end if
    else
        Rollback to the point when the highest EQ value $EQ_{new}$ is seen.
        Set $EQ_{old} = EQ_{new}$
    end if
    $t = r*t$
until $EQ$ does not increase for $c$ times or iteration number exceeds threshold $l$.
```

5. Experiments

In this section, we evaluate the partitioning quality of EQHyperpart partitioners by evaluating the algorithms using several real-world datasets, in comparison to hMETIS, khMETIS, and hyperpart. Note that experiments are performed by EQHyperpart-SA and EQHyperpart-MC unless stated otherwise.

5.1. Experiments setup

The evaluations are three-fold. Firstly, we evaluate the scalability of EQHyperpart and traditional hypergraph partitioning methods by measuring the increment speed of cross-partition query cost with the growth of network size. Secondly, several metrics are adopted to measure the partitioning quality of EQHyperpart, including the $K - 1$ cut size reducing extend, modularity retainment, and their tradeoff findings. Finally, the auto-tradeoff ability between cut size, modularity retainment, and unbalance level are evaluated without balance constraints in EQHyperpart.

We consider four datasets in the experiment. Three classical real-world social network datasets (i.e. Karate Club, Dolphin social network, and American College Football) are obtained from Mark Newman's personal website (Newman, 2015) and UCI Network Data Repository (UCI, 2015). These three datasets offer the actual community numbers and partition details. Another dataset is part of Facebook data offered by SNAP (Stanford Network Analysis Project) (SNAP, 2015; Mcauley and Leskovec, 2012). It reflects the earlier stage of a large social network. The Fast Girvan–Newman (FGN) algorithm (Newman, 2004) is performed on the Facebook dataset to detect communities, which act as the actual community structure for evaluation of hypergraph partitioning qualities.

These four datasets are modeled using hypergraph prior to partitioning execution, according to the inherent characteristic or inner-association of each dataset. For example, hyperedges are formed by the same associations, like friendship relationship, game relationship, and so forth. Moreover, the hypergraphs can be either unweighted or weighted, which take the degree as vertex weight. However, the hyperpart partitioner only supports unweighted hypergraph partitioning. Table 2 outlines the basic information of these datasets. Note that the unbalance level (UB-level) here uses the unbalance metric of khMETIS.

EQHyperpart is implemented in C++ using STL and Boost libraries, which is capable of processing weighted or unweighted hypergraph. It also allows the use of a hypergraph input file compatible with hMETIS. We perform the experiments on a Windows 2008 server, which has dual 2.0 GHz Intel Xeon processors with 8 GB of RAM.

Table 2
Real-world dataset characteristics.

Dataset	Vertices	Edges	Pins	Communities	$K - 1$ cut size	UB-level
Karate Club	34	78	190	2	11	12 ∣ 16 (w)
Dolphin	62	159	380	2	10	33 ∣ 41 (w)
American Football	115	616	1347	12	347	36 ∣ 33 (w)
Facebook	348~ 4039	11,422~ 88,234	23,192~ 180,507	Unidentified	Unidentified	Unidentified

5.2. Partitioner scalability experiments

As discussed in Section 3.2, the growing and preferential attachment mechanisms guide the generation of scale-free network. Before the next repartitioning point, the scale-free complex networks increase their sizes based on the latest partition results with the generative mechanisms during the interval. We seek to evaluate the scalability of partitioners under these mechanisms, that is, to measure the reduction rate of inter-partition query cost based on different initial placement configurations produced by different partitioners.

User nodes in Facebook form a typical scale-free network, such that they obey the generative mechanisms. In the experiments conducted on the Facebook dataset, the first 350 user nodes in this dataset are partitioned by respective partitioners to obtain the initial vertices placement layouts. Assuming that every node accesses its directed neighbour nodes only once, which are linked by the same hyperedge, followed by the calculation of average query cost of the whole network, denoted by C_{init} in (12). In the next placement phase, the successive nodes are placed in the proper partition according to the growing and preferential attachment mechanisms until the size of network reaches 1000. Finally, the average query cost is recalculated, denoted by C_{grown} in (12). The query cost saving rate R_{CS} is computed by (12), indicating the higher saving rate it gets, the more scalable the partitioner is:

$$R_{CS} = \frac{C_{init} - C_{grown}}{C_{init}} \tag{12}$$

Fig. 3 depicts the query cost saving rate in different unbalance levels produced by hMETIS, khMETIS, and EQHyperpart-SA, respectively. It can be observed that EQHyperpart-SA is more scalable, and this is mainly due to the modularity retaining ability of EQHyperpart-SA and the generative mechanism which impels the node to join the partition with more related nodes.

5.3. Partitioning quality experiments

To evaluate the partitioning quality at different balance levels, five partitioners are performed on the unweighted and weighted social network hypergraphs, respectively, with different unbalance factors. For hMETIS, its unbalance factor (UBFactor) ranges from 1 to 50 in steps of 0.5. UBFactor of khMETIS, EQHyperpart-SA and EQHyperpart-MC ranges from 5 to 100 in steps of 1. The low entropy of hyperpart ranges from the high entropy to half the high entropy, in steps of 0.005. Finally, for each of these partitioners in each test, the partition findings selected for comparison are those with balance levels closest to the EQHyperpart.

5.3.1. K–1 cut size within balance constraints

Statistical results of two datasets, including American Football and Facebook, are shown in Fig. 4, where we display $K-1$ cut size at different balance levels. We observe that EQHyperpart partitioners outperform other competing algorithms in weighted networks, because entropy based Q value is designed for nonuniform distributed scale-free networks. Specifically, EQHyperpart-MC significantly reduces $K-1$ cut size, with the help of micro cut. Note that there are some data missed in findings of hMETIS and hyperpart, because they are not able to produce partitioning results at certain balance level requirements.

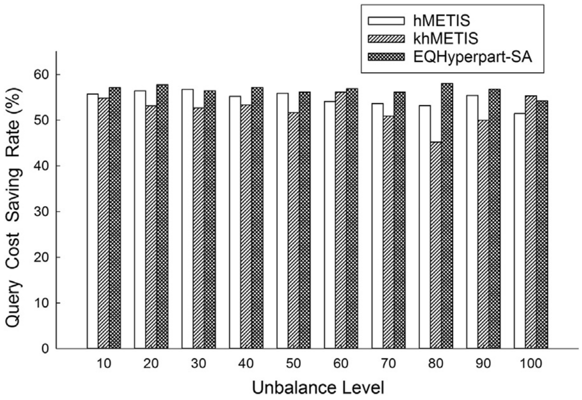

Fig. 3. Scalability comparison for partitioners.

5.3.2. Modularity features retainment ability

To evaluate the retainment ability of modularity features, we adopt the typical metrics in information retrieval system, including recall rate, precision rate, and *F*-score, as evaluation measures.

First, we match the partition $t \in T$ to the natural community $c \in C$ with the maximum degree of overlapping. Then, a confusion matrix is built based on the part-community-match results. Given the confusion matrix, we estimate for each community $c \in C$ the following quantities, namely: $\alpha(c, T)$ is the number of vertices correctly assigned to *c*, $\beta(c, T)$ is the number of vertices incorrectly assigned to c, and $\gamma(c, T)$ is the number of vertices incorrectly not assigned to *c*. Then, the averaged recall is defined by (13) and the averaged precision is defined by (14). In order to consider these two measures, the weighted harmonic mean of them, named F-Score, or F-Measure is widely used. Wherein the $F1$ is the most common, calculated as (15):

$$R(T) = \frac{\sum_c \alpha(c, T)}{\sum_c \alpha(c, T) + \gamma(c, T)} \tag{13}$$

$$P(T) = \frac{\sum_c \alpha(c, T)}{\sum_c \alpha(c, T) + \beta(c, T)} \tag{14}$$

$$F1 = \frac{2*P(T)*R(T)}{P(T) + R(T)} \tag{15}$$

The real world partitioning results (i.e. natural communities) of Karate Club, Dolphin, and American Football Team datasets are known beforehand. We achieve the natural communities of Facebook dataset by performing the FGN algorithm (Newman, 2004). The part numbers correspond to the real community numbers, and the Facebook dataset is divided into 4 parts in this experiment.

Figs. 5 and 6 depict the F-Measure findings at different designated balance levels, produced by five partitioners (i.e. hMETIS, khMETIS, hyperpart, EQHyperpart-SA, and EQHyperpart-MC) on four unweighted hypergraph modeled datasets and four weighted datasets, respectively.

We also take the worse percentage of $K-1$ cut size than true partition into consideration. This represents the deviation percentage of calculated $K-1$ cut size to the one in real world, and provides another perspective for similarity evaluation. The lower the deviation percentage, the more similar the partitioning result is to the real world situation. Figs. 7 and 8 show the $K-1$ cut size deviation relative to true partitions for unweighted and weighted datasets, respectively.

The findings for each dataset can be summarized as follows.

- In terms of Karate Club dataset, EQHyperpart partitioners, including both EQHyperpart-SA and EQHyperpart-MC, outperform in terms of modularity retaining ability, with the highest F-measure values and the closest $K-1$ cut size, especially on unweighted dataset.
- In terms of Dolphin dataset, EQHyperpart partitioners achieve full F-score on stricter unbalance level demand than other partitioners, and maintain the community characteristics when the unbalance upper limitation requirement increases.
- In terms of American Football dataset, the performance of EQHyperpart is modest. According to the natural partitioning of American Football dataset, there exists overlapping communities.

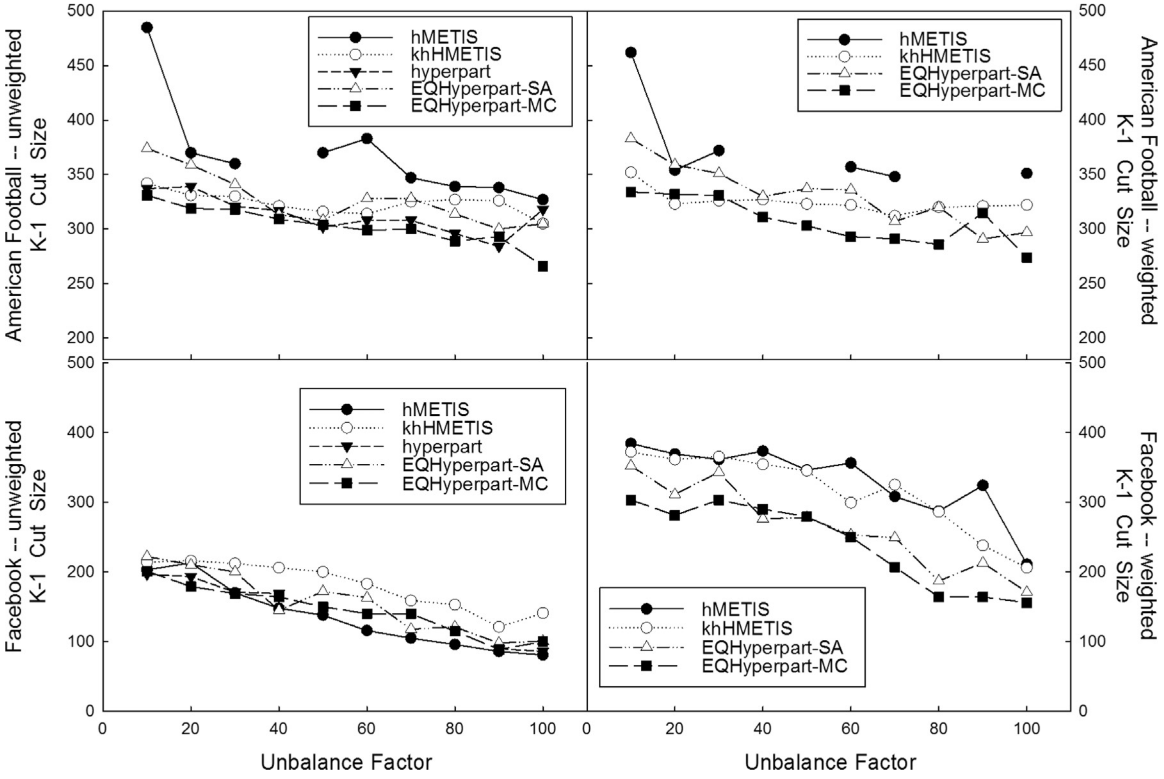

Fig. 4. Comparison of $K-1$ cut size with balance constraints.

The EQ design in EQHyperpart does not consider this particular characteristic of datasets. This may result in the deterioration of partitioning quality on such datasets.

- In terms of Facebook dataset, EQHyperpart partitioners perform better on weighted dataset, with a higher F-measure and a lower $K-1$ cut size deviation. Specifically, EQHyperpart-MC performs better than EQHyperpart-SA in general.

We also noted from the findings that when the unbalance levels are close to the points of true partitions (e.g., 16 in Karate Club, 41 in Dolphin, and 33 in American Football weighted dataset), EQHyperpart is extremely close to the best partitioner in F-

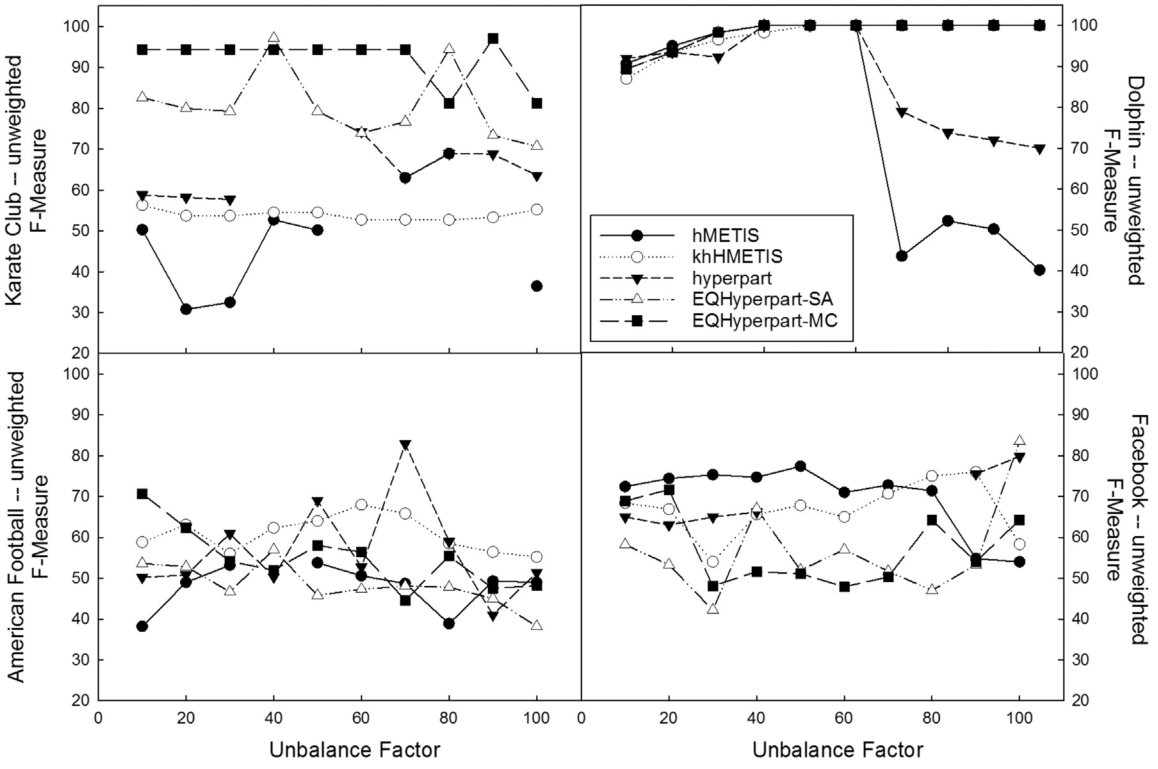

Fig. 5. F-Measure findings for unweighted datasets.

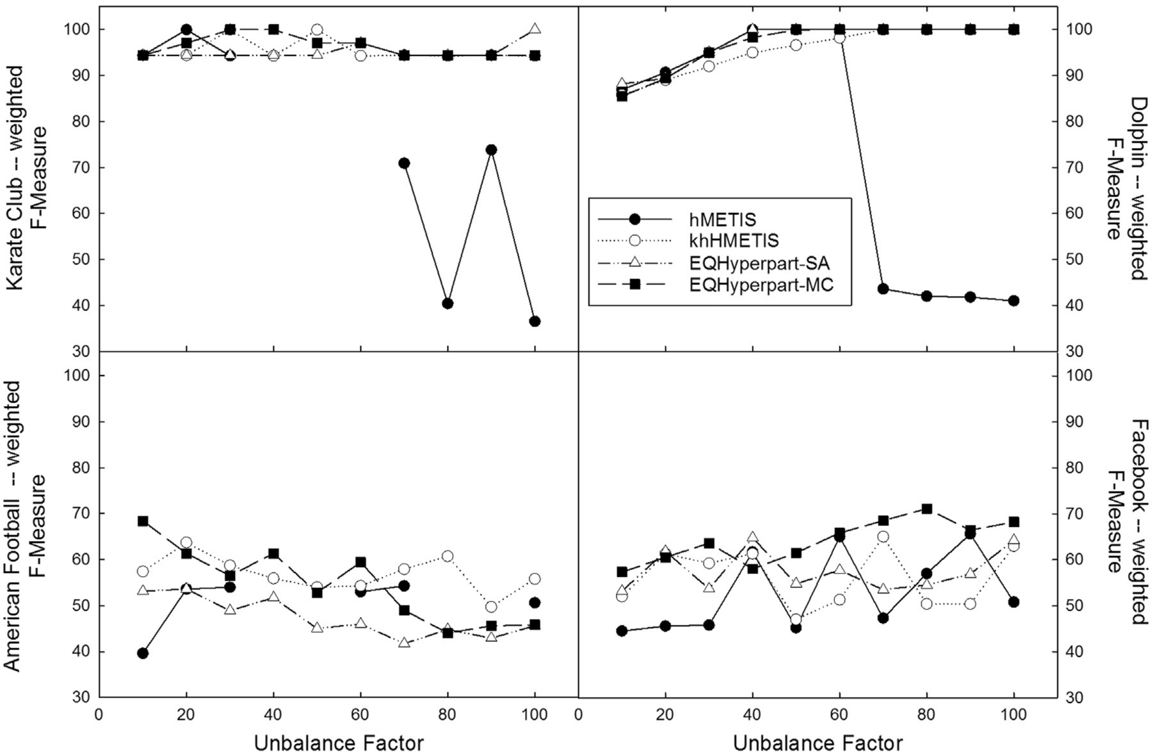

Fig. 6. F-Measure findings for weighted datasets.

Measure, and the $K-1$ cut sizes is extremely close to the real-world values. This implies that EQHyperpart partitioners are capable of maintaining the modularity, according to the nature characteristics of these networks. In addition, partition results produced by EQHyperpart on weighted datasets are better than those on unweighted datasets. This is particularly evident in Karate Club and Facebook datasets. The reason for this is that the weight reflects the node importance, which affects the generation of communities; thus, playing an important role in partitioning results of social networks.

In summary, we have demonstrated that EQHyperpart retains the modularity characteristics of social networks under balance constraints.

5.3.3. *Tradeoff between modularity and cut size*

Because EQHyperpart partitioning favors the increment of modularity retaining ability, and the decrement of $K-1$ cut size, we

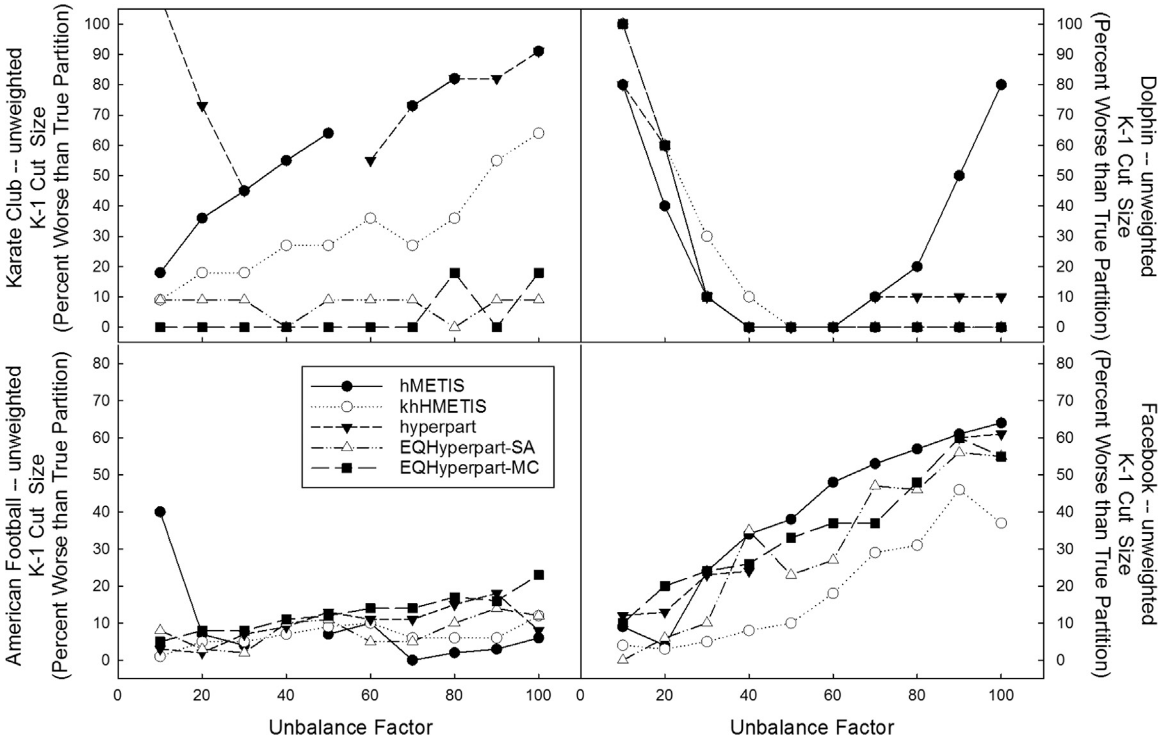

Fig. 7. $K-1$ cut size deviation to true partition for unweighted datasets.

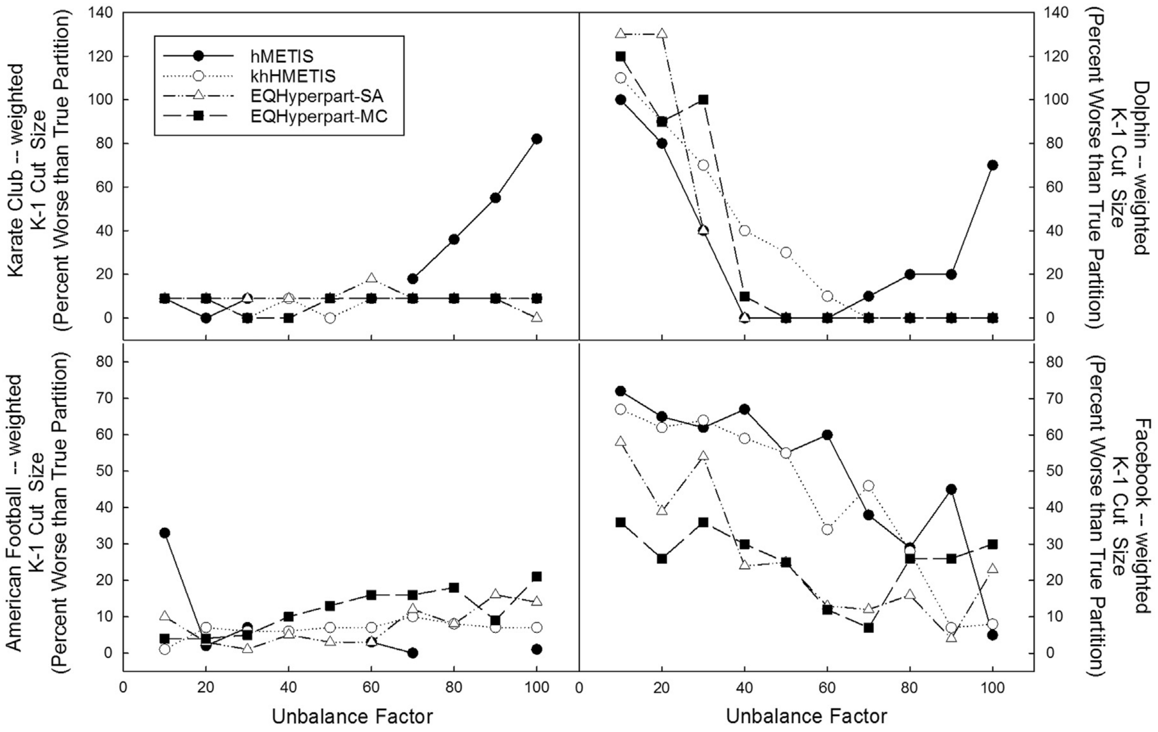

Fig. 8. $K-1$ cut size deviation to true partition for weighted datasets.

quantify the tradeoff between them under balance constraints using (16). Here, FS_i, CS_i and TO_i indicate the F-Score value, $K-1$ cut size and the tradeoff value under unbalance level i, respectively. $|E|$ denotes the number of edges of the dataset, and a and b are weighting coefficients for the two factors, subjected to $a+b=1$:

$$TO_i = a*FS_i*100 + b*\left(1-\frac{CS_i}{|E|}\right)*100. \tag{16}$$

Table 3 shows the average tradeoff of each weighted dataset of the unbalance levels, ranging from 0 to 100 by a factor of 10, and the coefficients a and b are both assigned 0.5. It can be observed that the average tradeoff values of `EQHyperpart` partitioners exceed the values of the other three partitioners; thus, validating its tradeoff ability between modularity retaining and $K-1$ cut size minimizing.

5.4. Auto-tradeoff partitioning experiments

Both `EQHyperpart-SA` and `EQHyperpart-MC` enable partitioning without balance constraints. In other words, `EQHyperpart` partitioners could terminate automatically at a point that results in an appropriate tradeoff between modularity, $K-1$ cut size and the balance level.

General partitioning favors the decrement of $K-1$ cut size and unbalance factor to obtain low query cost and load balancing. We argue that the modularity retaining ability is also important in social network partitioning, so we quantify the auto-tradeoff ability among these three elements using (17). Here, FS_{auto}, CS_{auto} and UB_{auto} indicate the F-Score value, $K-1$ cut size and the unbalance factor value produced by auto-tradeoff partitioner, respectively. $|E|$ denotes the number of edges of the dataset, and UB_{max} is the max unbalance factor. Similarly, a, b, and c are weighting coefficients, subjected to $a+b+c=1$:

$$TO_{auto} = a*FS_{auto}*100 + b*\left(1-\frac{CS_{auto}}{|E|}\right)*100 + c*\left(1-\frac{UB_{auto}}{UB_{\max}}\right)*100. \tag{17}$$

Table 4 illustrates the auto-tradeoff scores of the four unweighted and weighted datasets produced by `EQHyperpart-SA`. The unbalance factor adopts the metric of `hMETIS`; thus, $UB_{max}=50$. Auto-tradeoff score is calculated by setting the coefficients a=0.2, b=0.5 and c=0.3. `EQHyperpart-SA` was performed without any balance constraint for many times, and the most appropriate findings under the coefficient setting are listed in Table 4. In addition, the tradeoff score of real-world partition result for each dataset is included for comparison.

- In the datasets of Karate Club and Dolphin, the tradeoff scores of auto-tuning `EQHyperpart-SA` are approximated to those of real-word placements. The performance on Facebook is modest, and the worst performance is observed on American Football Team.
- On the condition of this coefficient setting, the real-word placement is not necessarily the one with the highest score. Some partitioning

Table 3
Tradeoff findings. (Findings are rounded. *(w)* indicates the dataset is modeled using weighted hypergraph, otherwise, unweighted.)

	hMETIS	khMETIS	hyperpart	EQHyperpart-SA	EQHyperpart-MC
Karate Club	71	72	77	83	**89**
Dolphin	85	**95**	90	**95**	**95**
Football	44	**54**	53	48	53
Facebook	**84**	83	70	78	78
Karate Club(w)	81	**91**		90	**91**
Dolphin(w)	84	94		**95**	94
Football(w)	45	52		47	**53**
Facebook(w)	75	77		78	**81**

Table 4
Auto-tradeoff findings. (Findings are rounded. (*R*) denotes the real-world partition result, and *(w)* denotes the weighted hypergraph modeled dataset, otherwise, unweighted.)

	K − 1 Cut size	UBFactor	*F*-score (%)	Tradeoff score
Karate Club(R)	11	6	100	89
Karate Club	9	9	81	85
Karate Club(w)	12	0	94	91
Dolphin(R)	10	17	100	87
Dolphin	12	15	98	87
Dolphin(w)	10	17	100	87
Football(R)	347	8	100	67
Football	181	33	31	52
Football(w)	181	33	31	52
Facebook(R)	223	12	100	92
Facebook	23	31	53	72
Facebook(w)	198	15	62	83

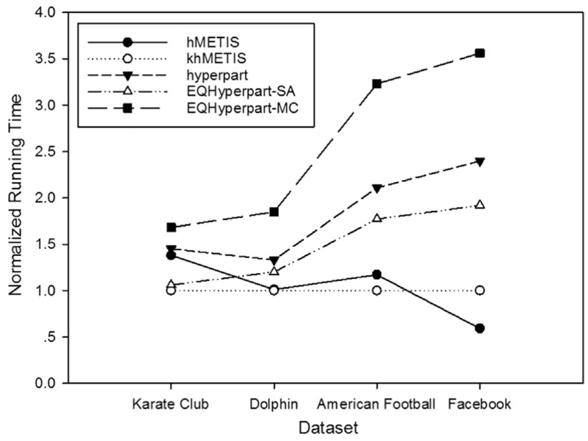

Fig. 9. Normalized running time comparison.

results produced by `EQHyperpart-SA` may have better overall performance. In fact, this is the universal phenomenon under other coefficient settings, but other auto-tuning partitioning findings could not be displayed due to space limitation. This is in accordance with that under balance constraints.

- The auto-tradeoff scores do not vary between unweighted and weighted datasets, with the exception of Facebook dataset. Specifically, `EQHyperpart-SA` achieves a better tradeoff on the weighted Facebook dataset than on the unweighted dataset, as the model of hypergraph affects the effect of auto-tuning partitioning results. We remark that Facebook is a classical scale-free network whose vertex degree distribution obeys the power law, which makes it more suitable to be modeled as a weighted hypergraph than an unweighted one. The larger the social network dataset, the stronger this effect.

In summary, the `EQHyperpart` produces satisfactory results without any balance constraint. In other words, we have validated the effectiveness of the auto-tradeoff partitioning capability of `EQHyperpart`.

5.5. Performance experiments

To evaluate the performance, we employ the execution time of five partitioners over the four unweighted datasets for comparison. Because the partitioning times on different unbalance levels over the same dataset are similar, we calculated the mean value of them for each partitioner on each dataset, and normalized the runtime relative to that of `khMETIS`. As shown in Fig. 9, `khMETIS` and `hMETIS` are effective in reducing execution time. However, this comes at the cost of unsatisfactory partitioning qualities as mentioned above. We also noted from the findings that, `EQHyperpart-SA` runs faster than `hyperpart`, but `EQHyperpart-MC` runs slower than `hyperpart`. The reason for this is that `EQHyperpart-MC` utilizes a more elaborate cut size based on `EQHyperpart-SA`, which needs a little increase on calculation cost in exchange for the decrease of communication cost. Comprehensively, the running performance of `EQHyperpart` is modest and acceptable.

6. Conclusion

In this paper, we studied hypergraph partitioning method for social networks, and presented a hypergraph partitioner, `EQHyperpart`, which utilizes modularity Q based on the scale-free featured information Entropy (EQ) to guide the low cost partitioning process.

We then demonstrated that `EQHyperpart` achieves low cut size while retaining the modularity characteristics at certain balance levels, and has an effective auto-tradeoff partitioning capability. We also proposed two variant partitioners `EQHyperpart-SA` and `EQHyperpart-MC` that utilize simulated annealing and micro cut heuristic to optimize the partition quality. We compared our approach with one state-of-the-art and two popular hypergraph partitioners, and demonstrated that `EQHyperpart` is more scalable and suitable for partitioning social networks, especially on the weighted hypergraph modeled ones.

Future work includes extending this research by expanding our partitioning method and validating the approach using other widely used social network datasets.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant Numbers 61632009, 61472451, 61272151, and 61402543, the High Level Talents Program of Higher Education in Guangdong Province under Funding Support Number 2016ZJ01, the Natural Science Foundation of Guangdong Province in China under Grant Number 2015A030313638, and the Foundation for Distinguished Young Talents in Higher Education of Guangdong in China under Funding Support Number2015KQNCX179.

References

Aaron, C., Newman, M.E.J., Cristopher, M., 2004. Finding community structure in very large networks. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 70 (6), 264–277.
Arora, S., Rao, S., Vazirani, U., 2009. Expander flows, geometric embeddings and graph partitioning. J. ACM 56 (2), 1–37.
Barabàsi, A.L., Bonabeau, E., 2003. Scale-free networks. Sci. Am. 288 (3), 60–69.
Barabàsi, A.-L., Albert, R., Jeong, H., 2000. Scale-free characteristics of random networks: the topology of the world-wide web. Physica A: Stat. Mech. Appl. 281 (1–4), 69–77.
Berge, C., 1985. Graphs and Hypergraphs. Elsevier Science Ltd, Oxford, UK.
Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E., 2008. Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. 30 (2), 155–168.
Boltzmann, L., 2003. Further studies on the thermal equilibrium of gas molecules. Kinet Theory Gases Ser.: Hist. Mod. Phys. Sci. 1, 262–349.
Borndörfer, R., Heismann, O., 2015. The hypergraph assignment problem. Discr. Optim. 15 (4), 15–25.
Carullo, G., Castiglione, A., Cattaneo, G., De Santis, A., Fiore, U., Palmieri, F., 2013. Feeltrust: providing trustworthy communications in ubiquitous mobile environment. In: the 27th IEEE International Conference on Advanced Information Networking and Applications, pp. 1113–1120.
Carullo, G., Castiglione, A., De Santis, A., Palmieri, F., 2015. A triadic closure and homophily-based recommendation system for online social networks. World Wide Web 18 (6), 1579–1601.
Curino, C., Jones, E., Zhang, Y., Madden, S., 2010. Schism: a workload-driven approach to database replication and partitioning. Proc. VLDB Endow. 3 (1–2), 48–57.
Deng, X., Wang, B., Wu, B., Yang, S., 2009. Research and evaluation on modularity modeling in community detecting of complex network based on information entropy. In: the 3rd IEEE International Conference on Secure Software Integration & Reliability Improvement, pp. 297–302.
Deveci, M., Kaya, K., UçarÜmit, B., Çatalyürek, V., 2015. Hypergraph partitioning for multiple communication cost metrics: model and methods. J. Parallel Distrib. Comput. 77, 69–83.
Eustace, J., Wang, X., Cui, Y., 2015. Overlapping community detection using neighborhood ratio matrix. Physica A: Stat. Mech. Appl. 421, 510–521.

Fiduccia, C.M., Mattheyses, R.M.,1982. A linear-time heuristic for improving network partitions. In: Proceedings of the 19th Design Automation Conference, pp. 175–181.

Guzzo, A., Pugliese, A., Rullo, A., Saccà, D., 2014. Intrusion detection with hypergraph-based attack models. In: Graph Structures for Knowledge Representation and Reasoning. Springer, Cham, Switzerland, 2014, pp. 58–73.

He, J., Chen, D., 2015. A fast algorithm for community detection in temporal network. Physica A: Stat. Mech. Appl. 429, 87–94.

Heintz, B., Chandra, A., 2014. Beyond graphs: toward scalable hypergraph analysis systems. ACM SIGMETRICS Perform. Eval. Rev. 41 (4), 94–97.

Jiang, J., Wilson, C., Wang, X., Huang, P., Sha, W., Dai, Y., Zhao, B.Y., 2010. Understanding latent interactions in online social networks. ACM Trans. Web 7 (4), 369–382.

Jin, L., Chen, Y., Wang, T., Hui, P., Vasilakos, A.V., 2013. Understanding user behavior in online social networks: a survey. Commun. Mag. IEEE 51 (9), 144–150.

Karypis, G., Kumar, V., 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. 20 (1), 359–392.

Karypis, G., Kumar, V., 1999. Multilevel k-way hypergraph partitioning. In: Proceedings of the 36th Annual ACM/IEEE Design Automation Conference, pp. 343–348.

Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S., 1999. Multilevel hypergraph partitioning: applications in VLSI domain. IEEE Trans. Very Large Scale Integr. Syst. 7 (1), 69–79.

Kernighan, B.W., Lin, S., 1970. An efficient heuristic procedure for partitioning graphs. Bell Syst. Tech. J. 49 (2), 291–307.

Kwak, H., Choi, Y., Eom, Y.H., Jeong, H., Moon, S., 2009. Mining communities in networks: a solution for consistency and its evaluation. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference. ACM, New York, NY, USA, pp. 301–314.

Ma, T., Zhou, J., Tang, M., Tian, Y., Al-Dhelaan, A., Al-Rodhaan, M., Lee, S., 2015. Social network and tag sources based augmenting collaborative recommender system. IEICE Trans. Inf. Syst. 98 (4), 902–910.

Martin, R., Bergstrom, C.T., 2007. An information-theoretic framework for resolving community structure in complex networks. Proc. Natl. Acad. Sci. 104 (18), 7327–7331.

Mcauley, J.J., Leskovec, J., 2012. Learning to discover social circles in ego networks. Adv. Neural Inf. Process. Syst., 539–547.

Newman, M.E.J., Girvan, M., 2004. Finding and evaluating community structure in networks. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 69 (2), 026113.

Newman, M.E., 2004. Fast algorithm for detecting community structure in networks. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 69 (6), 066133.

Newman, M.E., 2006. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 74 (3), 92–100.

Newman, M.E.J., 2006. Modularity and community structure in networks. Proc. Natl. Acad. Sci. 103 (23), 8577–8582.

Newman, M.E.J., 2015. Website of Mark Newman, ⟨http://www-personal.umich.edu/mejn/netdata/http://www-personal.umich.edu/ mejn/netdata/⟩.

Palla, G., Derényi, I., Farkas, I., Vicsek, T., 2005. Uncovering the overlapping community structure of complex networks in nature and society. Nature 435 (7043), 814–818.

Pons, P., Latapy, M.,2005. Computing communities in large networks using random walks. In: Proceedings of the 20th International Conference on Computer and Information Sciences. Springer, Berlin, Heidelberg, 2005, pp. 284–293.

Popescu, P.G., Pop, F., Herişanu, A., Ţaˇpuş, N., 2013. New inequalities between information measures of network information content. Math. Probl. Eng. 2013 (2013), 151–164.

Popescu, P.G., Sluşanschi, E.-I., Iancu, V., Pop, F., 2016. A new upper bound for Shannon entropy: a novel approach in modeling of big data applications. Concurr. Comput.: Pract. Exp. 28 (2), 351–359.

Pujol, J.M., Siganos, G., Erramilli, V., Rodriguez, P., Pujol, Siganos, J.M.G., Scaling online social networks without pains. In: Proceedings of NetDB.

Pujol, J.M., Erramilli, V., Siganos, G., Yang, X., Laoutaris, N., Chhabra, P., Rodriguez, P., 2010. The little engine(s) that could: scaling online social networks. ACM SIGCOMM Comput. Commun. Rev. 40 (4), 1162–1175.

Saleh, I., Wei, T., Blake, M.B., 2013. Social-network-sourced big data analytics. Internet Comput. IEEE 17 (5), 62–69.

Santo, F., Vito, L., Massimo, M., 2004. Method to find community structures based on information centrality. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 70 (5), 148–168.

Selim, S.Z., Alsultan, K., 1991. A simulated annealing algorithm for the clustering problem. Pattern Recognit. 24 (91), 1003–1008.

Shannon, C.E., 1948. A mathematical theory of communication. Bell Syst. Tech. J. 27 (3), 379–423.

Shen, H., Cheng, X., Chen, H., Liu, Y., 2008. Information bottleneck based community detection in network. Chin. J. Comput. 31 (4), 677–686.

SNAP, 2015. Stanford large network dataset collection, ⟨http://snap.stanford.edu/data/index.html⟩ (2015).

Turk, A., Oguz Selvitopi, R., Ferhatosmanoglu, H., Aykanat, C., 2014. Temporal workload-aware replicated partitioning for social networks. IEEE Trans. Knowl. Data Eng. 26 (11), 2832–2845.

UCI, 2015. UCI network data repository, ⟨http://networkdata.ics.uci.edu/⟩.

Wang, T.M., Lee, W.T., Wu, T.Y., Wei, H.W., Lin, Y.S., 2012. New P2P sharing incentive mechanism based on social network and game theory. J. Netw. Comput. Appl. 41 (3), 47–55.

Wilson, C., Boe, B., Sala, A., Puttaswamy, K.P.N., Zhao, B.Y., 2011. User interactions in social networks and their implications. Nat. Product Commun. 6 (1), 137–140.

Yang, W., Wang, G., 2015. Directed social hypergraph data allocation strategy in online social networks. J. Chin. Comput. Syst. 36 (7), 1559–1564.

Yang, W., Wang, G., Bhuiyan, M.Z.A., 2015. Partitioning of hypergraph modeled complex networks based on information entropy. In: Proceedings of the 15th International Conference on Algorithms and Architectures for Parallel Processing, pp. 678–690.

Yaros, J.R., Imielinski, T., 2013. Imbalanced hypergraph partitioning and improvements for consensus clustering. In: The 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI). IEEE Press, Piscataway, N.J., USA, pp. 358–365.

Yuan, M., Stein, D., Carrasco, B., Trindade, J.M.F., Lu, Y., 2012. Partitioning social networks for fast retrieval of time-dependent queries. In: 2012 IEEE 28th International Conference on Data Engineering Workshops (ICDEW), pp. 205–212.

Zaerpour, N., Rabbani, M., Gharehgozli, A.H., Tavakkoli-Moghaddam, R., 2009. A comprehensive decision making structure for partitioning of make-to-order, make-to-stock and hybrid products. Soft Comput. 13, 1035–1054.

Zheng, X., Yi, D.Y., Ouyang, Z.Z., Dong, L., 2012. Hyperedge communities and modularity reveal structure for documents. Chin. Phys. Lett. 29 (3), 038902.