Finding Better Web Communities in Digraphs via Max-Flow Min-Cut

Chung Chan, Ali Al-Bashabsheh, Da Sun Handason Tam and Chao Zhao

***Abstract*—We consider the web community detection problem by providing a cost function that, not only penalizes external connections, but also rewards the internal ones. Our formulation addresses limitations of cut-clustering and extends web communities to digraphs. The formulation is parametric, resulting in a hierarchy of communities that is representable in linear storage and computable in a linear number of maxflow computations. Experimental results on synthetic and real-world datasets show that the proposed method can find better web communities and more densest subgraphs than previous formulations. Simple examples also show it can return different and more meaningful communities than other formulations based on graph conductance, map equation and modularity score.**

I. Introduction

In this work, we consider the problem of graph clustering, see e.g., [1], where communities of highly related nodes are identified based on their interconnections. This has significant applications in studying social and biological systems where related entities tend to interact more with each other compared to unrelated ones. Although the notion of community is intuitive, the challenge, however, is to find a precise mathematical definition that efficiently identifies the communities of interest.

The most widely accepted definition of a community, or graph cluster, is perhaps that of [2–4]. Roughly speaking, a community is a group of nodes that share stronger connections among themselves compared with the rest of the graph. Such a definition naturally inspires graph cut [5, 6], modularity [4, 7], random walk [8] based methods, to name a few, for identifying communities. These methods have to refine the original definition of a community for efficient computation and storage. In particular, web communities defined in [2, 5] are NP-hard to compute, and so a cut-clustering algorithm was proposed in [5] that finds approximate web communities by minimizing a weighted sum of the external connections and the size of the community. It was shown that the computation reduces to finding the mincuts of some modified graphs, and the set of all such communities, or cut-clusters, form a hierarchy, which can be represented efficiently by a dendrogram.

However, the hierarchical structure of cut-clusters rely on the graph being undirected. Furthermore, we find that cut-clustering often fails to return dense subgraphs but instead returns the less interconnected peripherals of the dense subgraphs. This behaviour is not surprising since, for a given size, a set of nodes having fewer external connections does not necessarily imply they have more internal connections. In other words, a dense subgraph may have many external connections with the rest of the graph compared with the external connections of a sparser subgraph. In this work, we primarily focus on improving the cut-clustering algorithm. Our proposed solution not only resolves the problem with no additional cost in computation and storage, but also extends to cluster digraphs or more general non-graphical networks with a submodular cost function.

This paper is organized as follows. In Section II, we introduce the notion of web communities for digraphs and the approximate solution by cut-clustering for undirected graphs. In Section III, we improve the formulation of cut-clustering and extend it to digraphs. In Sections IV and V, we explain the hierarchical structure of the communities and its polynomial-time computation. In Section VI, we give test results comparing the proposed algorithm to cut-clustering on two small example graphs. Detailed comparisons with cut-clustering and other algorithms can be found in Appendices A and B along with experiments on both synthetic and real-world data sets. The proofs and detailed calculations of some of the examples are given in Appendices C and D.

II. Preliminaries

We consider a digraph with non-negative real-valued edge weights and a finite set V of $|V| > 1$ vertices represented by the adjacency matrix

$$\boldsymbol{A} := [a_{ij}] \in \mathbb{R}_+^{|V|\times|V|}.$$

For convenience, we write

$$w(B,C) := \sum_{i\in B}\sum_{j\in C} a_{ij} \quad \text{for } B,C \subseteq V,$$

$w(i,C) := \sum_{j\in C} a_{ij}$ for $i \in V$, and $w(B,j) := \sum_{i\in B} a_{ij}$ for $j \in V$. The weight a_{ij} of edge $(i,j) \in V^2$ is taken to mean the influence of i on j, which needs not be equal to the influence a_{ji} of j on i for $i \neq j$. In the following definition, we generalizes the idea of web communities in [2, 5] from graphs to digraphs.

Definition 1 A *web community* is a non-empty subset $C \subseteq V$ of the vertices that satisfies

$$w(i, C \setminus \{i\}) > w(V \setminus C, i), \ \forall i \in C, \tag{1}$$

i.e., the total external influence on any member of a web community is strictly smaller than the influence of that member on the remaining members of the community. □

C. Chan (corresponding author, email: chung.chan@cityu.edu.hk), D.S.H. Tam, and C. Zhao are with the Department of Computer Science, City University of Hong Kong.

A. Al-Bashabsheh is with the Big Data and Brain Computing (BDBC) center at Beihang University, Beijing, China.

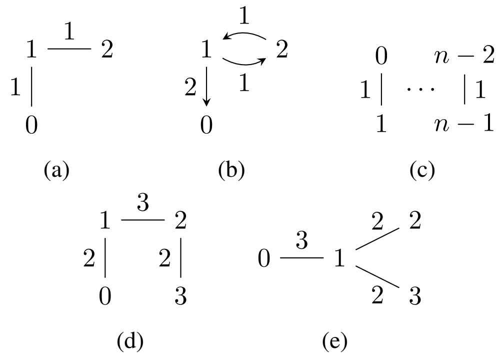


Fig. 1: The weighted graphs for Examples 1–4

Note that singleton sets $\{i\}$ for $i \in V$ are not web communities due to the strict inequality in (1). If $\boldsymbol{A}$ is symmetric, which corresponds to a graph, the above definition reduces to the original definition in [5]. The following example shows that our extension is non-trivial as the web communities can depend on the directions of the edges.

Example 1 Consider the undirected graph in Fig. 1a where $w(i,j) = 1$ for $\{i,j\} \in \{\{0,1\},\{1,2\}\}$, and the digraph in Fig. 1b where $w(1,0) = 2$ and $w(1,2) = w(2,1) = 1$. The two graphs differ only in the directions of the edges, i.e., they have the same values of $w(i,j) + w(j,i)$ for all $i,j \in V$. However, the two graphs have different web communities: $\{0,1,2\}$ for the undirected graph and $\{1,2\}$ for the digraph. $\{0,1,2\}$ is not a web community in the digraph in Fig. 1b since node 0 does not have any influence on other nodes, violating the condition (1). □

An issue with the above definition of communities is that there can be exponentially many of them as illustrated below.

Example 2 For the graph in Fig. 1c where the edges form a perfect matching, the web communities are all the non-empty subsets of the matched pairs. The number of communities is $2^{|V|/2} - 1$, which is exponential in $|V|$. □

Since there can be exponentially many web communities, it is impractical to enumerate all of them for a large graph. Even for graphs of moderate sizes, returning many communities without any form of organization or measures of quality is not helpful, especially if the goal is to study a large graphical network by breaking it down into subnetworks.

In the case when the graph is undirected, [5] proposed the cut-clustering algorithm below:

1) For any parameter $\alpha \in \mathbb{R}$, add a new node s to the graph.
2) For each node $t \in V$, add an edge between s and t with weight α.
3) Construct a Gomory-Hu tree of the graph. Remove s from the tree and return the resulting connected components, i.e., a partition of V. If the Gomory-Hu tree is not unique, construct the tree that gives the finest partition.

The above procedure is called cut-clustering since, by the property of the Gomory-Hu tree, each of the resulting connected components corresponds to an s–t mincut of the graph augmented with s. We will refer to the non-singleton components as cut-clusters at threshold α. It was shown that α serves as a parameter that measures the quality of the returned clusters in terms of graph expansion [5, (3.3)]. However, the extension to digraphs is unclear since Gomory-Hu trees are defined for undirected graphs. Another limitation is that a cut-cluster may not be a web community as one of the nodes in a cut-cluster can fail to satisfy (1) [5, Lemma 3.1].

III. A better formulation

We propose the following more sophisticated definition of communities parameterized by different quality requirements. The communities will be shown to have a meaningful hierarchical structure that can be computed and represented in polynomial-time using maxflow algorithms.

Definition 2 Given $\beta \in [0,1]$, define for $\alpha \in \mathbb{R}$

$$\hat{f}(\alpha) := \min_{C \subseteq V : |C| \geq 1} f_\alpha(C) \quad \text{where} \tag{2a}$$

$$f_\alpha(C) := f(C) + \alpha \cdot |C| \tag{2b}$$

$$f(C) := (1-\beta) \cdot w(V \setminus C, C) - \beta \cdot w(C, C), \tag{2c}$$

where we have made the dependency on β implicit for notational simplicity. The set of communities is defined as

$$\mathcal{C} := \bigcup\nolimits_{\alpha \in \mathbb{R}} \mathcal{S}_\alpha \tag{3}$$

where $\mathcal{S}_\alpha$ is defined as the collection of $C \subseteq V$ such that $|C| > 1$, i.e., C is non-singleton, and

$$f_\alpha(C) = \hat{f}(\alpha) < \min_{B \subsetneq C : |B| \geq 1} f_\alpha(B), \tag{4}$$

i.e., C is an inclusion-wise minimal solution to (2a). For each community $C \in \mathcal{C}$, we define

$$\sigma(C) := \sup\{\alpha \in \mathbb{R} \mid C \in \mathcal{S}_\alpha\} \tag{5}$$

as a measure of the strength of the community. □

Subsequently, unless otherwise specified in the context, a community will refer to one according the definition above. In the definition, the parameters α and β allow for a tuning of the quality of the communities. Namely, the cost function $f(C)$ (2c) penalizes the *external influence* $w(V \setminus C, C)$ and rewards the *internal influence* $w(C, C)$. Since the entire set V trivially minimizes the external influence and maximizes the internal influence, we further penalize the size of C in the objective function $f_\alpha(C)$ (2b) with $\alpha \geq 0$ to obtain more compact communities. A simple connection to web communities is given by the following result, which provides a strong guarantee on the quality of the communities than that of the cut-clusters in [5, Lemma 3.1].

Proposition 1 *Every $C \in \mathcal{S}_\alpha$ defined with (4) satisfies*

$$w(i, C) > w(V \setminus C, i) + (\alpha - \beta d_i) \quad \forall i \in C, \tag{6a}$$

$$w(V \setminus C, i) \geq w(i, C) - (\alpha - \beta d_i) \qquad \forall i \in V \setminus C, \tag{6b}$$

where $d_i := w(V \setminus \{i\}, i)$ is the in-degree of vertex i. □

Corollary 1 *A community $C \in \mathcal{C}$ defined in* (3) *is a web community if $\sigma(C) > \beta \max_{i \in C} d_i$.* □

Equation (6a) relates our communities and web communities by bounding the gap between the internal and external influences in terms of the community parameters. For instance, for $\beta = 0$, a community is always a web community for $\alpha \geq 0$, and, moreover, α provides a lower bound on the gab between the internal and external influences. Note that while the parameter β might appear to have an undesirable effect by diminishing the gap between the internal and external influences, and so leading to communities that are not web communities, it is one of the contributions we claim in this work. Indeed, as will be argued in a subsequent section, there can be meaningful communities that may or may not be web communities, and they cannot be identified unless $\beta > 0$.

Proof (Sketch) (6a) follows from that fact that $C \setminus \{i\}$ for any $i \in C$ and $C \in \mathcal{C}$ is a strictly suboptimal solution to (2), while (6b) follows from the fact that $C \cup \{i\}$ for any $i \in V \setminus C$ is a feasible but not necessarily optimal solution. The corollary follows from (6a) since $C \in \mathcal{S}_\alpha$ for some α arbitrarily close to $\sigma(C)$ by the definition (5) of $\sigma(C)$. ■

The following example illustrates the definition of the communities and its desired property.

Example 3 Consider Fig. 1c as in Example 2. Assuming $\beta = 0$, we have $f(C) = w(V \setminus C, C)$ by (2c). It is easy to see that $\hat{f}(0) = 0$ because the solution to (2a) when $\alpha = 0$ are the unions of the matched pairs $C_i := \{2i, 2i+1\}$ for $0 \leq i < n/2$. By (4), $\mathcal{S}_0$ is the set of matched pairs C_i's since they are inclusion-wise minimal solutions that are non-singleton. Similarly, it is straightforward to show that for

- $\alpha < 0$: $\hat{f}(\alpha) = \alpha|V|$ and $\mathcal{S}_\alpha = \{V\}$.
- $\alpha \in [0, 1)$: $\hat{f}(\alpha) = 2\alpha$ and $\mathcal{S}_\alpha = \{C_i \mid 0 \leq i < n/2\}$.
- $\alpha \geq 1$: $\hat{f}(\alpha) = 1 + \alpha$ and $\mathcal{S}_\alpha = \emptyset$.

By (3), the set $\mathcal{C}$ of communities consists of V and the matched pairs C_i's. By (5), $\sigma(V) = 0$ and $\sigma(C_i) = 1$. By Corollary 1, since $\sigma(C_i) > 0 = \beta \sum_{i \in C} d_i$, C_i's are web communities. □

In the above example, it can be seen that $\mathcal{C}$ captures the essential web communities for Fig. 1c, which form a meaningful hierarchy with respect to the quality measure σ.

IV. Community hierarchy

The following result shows that, similar to hierarchical clustering methods, the set of communities forms a hierarchy and so can be represented by a dendrogram in linear storage.

Theorem 1 *For $\alpha_1 \geq \alpha_2 \geq 0$ and $C_i \in \mathcal{S}_{\alpha_i}$ for $i = 1, 2$, we have $C_1 \subseteq C_2$ or $C_1 \cap C_2 = \emptyset$. Furthermore, $C_1 \subsetneq C_2$ implies $\alpha_1 > \alpha_2$. Hence, the set $\mathcal{C}$ of communities can be represented by a dendrogram with the strength σ measuring the cophenetic similarity of the dendrogram.* □

The hierarchical structure can be observed from Example 3, where matched pairs C_i's are disjoint communities with the same strength $\sigma(C_i) = 1$. Since the trivial community V contains a matched pair, it has a strictly smaller strength $\sigma(V) = 0$ as expected. We remark that the proof of the theorem relies only on the submodularity (discussed in Section V) of f_β, and so the theorem extends to submodular functions that are not necessarily defined in terms of graph cut.

To understand the parameter α as a measure of similarity, we will strengthen Proposition 1 to give a more precise interpretation of α as a bound on the marginal change in the cost of a community.

Theorem 2 *Each $C \in \mathcal{S}_\alpha$ satisfies*

$$\alpha < \min_{B \subsetneq C : |B| \geq 1} \frac{f(B) - f(C)}{|C \setminus B|}, \tag{7a}$$

$$\alpha \geq \max_{A \subseteq V : A \supsetneq C} \frac{f(C) - f(A)}{|A \setminus C|}. \tag{7b}$$

(By convention, we set the r.h.s. of (7b) *to $-\infty$ if $C = V$.)* □

In other words, α is both a lower bound (7a) on the marginal increase in the cost f when the community shrinks, and an upper bound (7b) on the marginal decrease in the cost f when the community expands. The above theorem can be viewed as a generalization of Proposition 1 because (6a) and (6b) are the special cases when we further impose $B = C \setminus \{i\}$ for $i \in C$ in (7a), and $A = C \cup \{i\}$ for $i \in V \setminus C$ in (7b) respectively. As the following corollary shows, the result also ties back to the notion of graph expansion used to measure cluster quality.

Corollary 2 *Each $C \in \mathcal{S}_\alpha$ satisfies for $\beta = 0$*

$$\frac{w(V \setminus C, C)}{|V \setminus C|} \overset{\text{(i)}}{\leq} \alpha \overset{\text{(ii)}}{<} \min_{B \subsetneq C : |B| \geq 1} \frac{w(C \setminus B, B)}{|C \setminus B|} \tag{8a}$$

and, for $\beta = 1$,

$$\max_{A \subseteq V : A \supsetneq C} \frac{w(A, A) - w(C, C)}{|A \setminus C|} \overset{\text{(iii)}}{\leq} \alpha \overset{\text{(iv)}}{<} \frac{w(C, C)}{|C| - 1}. \tag{8b}$$

□

Proof (ii) and (iii) follow from (7a) and (7b) with $\beta = 0$ and 1, respectively. (i) and (iv) follow from (7b) and (7a) by choosing $A = V$ with $\beta = 0$ and $B \subsetneq C : |B| = 1$ with $\beta = 1$, respectively. ■

V. Computation

In this section, we will derive a polynomial-time algorithm that returns the proposed community hierarchy $\mathcal{C}$ for any given $\beta \in [0, 1]$. The implementation is available at [9].

A. Divide-and-conquer

We first rewrite (2) as a two-step minimization

$$\hat{f}(\alpha) = \min_{t \in V} \hat{f}_t(\alpha), \quad \text{where} \tag{9a}$$

$$\hat{f}_t(\alpha) := \min_{C \subseteq V : t \in C} f_\alpha(C) \tag{9b}$$

and f_α is defined in (2b). The minimization in (9b) has a unique inclusion-wise minimum solution due to a well-known result in submodular function minimization (see [10]), and the fact that f (and thereby f_α) is submodular, i.e., $\forall C_1, C_2 \subseteq V$,

$$f(C_1) + f(C_2) \geq f(C_1 \cup C_2) + f(C_1 \cap C_2). \tag{10}$$

This can be seen by rewriting f, defined in (2c), using the identity $w(V \setminus C, C) + w(C, C) = \sum_{i \in C} d_i$ as

$$f(C) = w(V \setminus C, C) - \beta \sum_{i \in C} d_i, \tag{11}$$

which is submodular in C since the graph cut $w(V \setminus C, C)$ is submodular (and the sum $\sum_{i \in C} d_i$ is modular) [10].

Definition 3 For $\alpha \geq 0$, $\beta \in [0,1]$, and $t \in V$, a community candidate, or candidate for short, $C_{\alpha,t}$ is defined as the inclusion-wise minimum solution to the minimization (9b). □

(Similar to communities, we make the dependency on β implicit for notational simplicity.) For a given $\beta \in [0,1]$, let T^*_α be the set of optimal solutions t to the minimization (9a), then we have the following proposition.

Proposition 2 *The set of inclusion-wise minimal solutions to (2) is equal to the set of minimal candidates $C_{\alpha,t}$ s.t. $t \in T^*_\alpha$. In other words, the set of communities $\mathcal{S}_\alpha$ can be written as*

$$\operatorname{minimal}\{C_{\alpha,t} \mid t \in T^*_\alpha\} \setminus \{\{i\} \mid i \in V\}, \tag{12}$$

PROOF Follows directly from Definitions 2 and 3. ■

Example 4 Consider the undirected graph in Fig. 1d with $V := \{0,1,2,3\}$. It is straightforward to enumerate all the web communities (1) as

$$\{1,2\}, \{0,1,2\}, \{1,2,3\}, \{0,1,2,3\}. \tag{13}$$

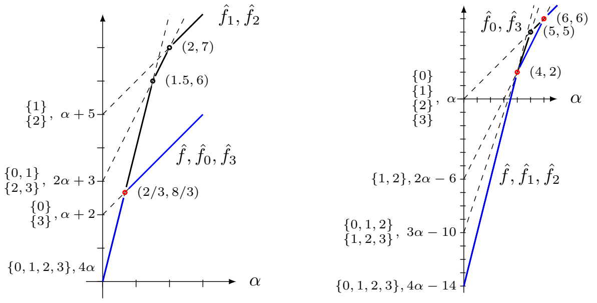
For $\beta = 0$, Fig. 2a shows the plots (against α) of the functions $\hat{f}(\alpha)$ (blue), $\hat{f}_t(\alpha)$ (solid), and $f_\alpha(C)$ (dashed black).[1] For $\beta = 1$, the same functions are shown in Fig. 2b. The candidates are determined by the solid lines, e.g., for $\beta = 0$, we have $C_{\alpha,t}$ as

$$\begin{array}{ccccl} t=0 & t=1 & t=2 & t=3 & \\ \{0\} & \{1\} & \{2\} & \{3\}, & \alpha \geq 2 \\ \{0\} & \{0,1\} & \{2,3\} & \{3\}, & \alpha \in [1.5, 2) \\ \{0\} & V & V & \{3\}, & \alpha \in [\frac{2}{3}, 1.5) \\ V & V & V & V, & \alpha < 2/3\ . \end{array} \tag{14}$$

Note that each entry in (14) is the inclusion-wise minimum solution to (10). Propositions 2 asserts that, for a given β, the set of communities is specified by the blue line as

$$\begin{array}{cc} \beta = 0 & \beta = 1 \\ \begin{cases} \emptyset, & \alpha \geq 2/3 \\ \{0,1,2,3\}, & \alpha < 2/3 \end{cases}, & \begin{cases} \emptyset, & \alpha \geq 6 \\ \{1,2\}, & \alpha \in [4,6) \\ \{0,1,2,3\}, & \alpha < 4 \end{cases}. \end{array} \tag{15}$$

All the communities at $\beta = 0$ (trivial in this example) are web communities as dictated by Corollary 1. Note also that the web community $\{1,2\}$ is captured at $\beta = 1$ but not $\beta = 0$, i.e., when the cost function (2c) rewards the internal influence. This situation may persist in general, and shows the benefit of choosing $\beta > 0$. Namely, if there is a dense subgraph that is moderately connected with the rest of the graph, then it is reasonable to favor such a subgraph over a sparse one, even if the sparse subgraph exhibits weaker connection to the rest of the graph.



(a) $\beta = 0$

(b) $\beta = 1$

Fig. 2: The plots of $\hat{f}_t(\alpha)$ (solid) against α for the undirected graph in Fig. 1d. The optimal $\hat{f}_t(\alpha)$ define $\hat{f}(\alpha)$ (solid blue).

In the example above, the non-trivial community at $\beta = 1$ is a web community, which is desirable. However, for the same reason mentioned in the example, a community at $\beta > 0$ may still be desirable even if it is not a web community. For instance, consider the graph in Fig. 1e. The set $\{0,1\}$ is not a web community since node 1 is more connected with the rest of the graph than it is with node 0. Despite this, it is still desirable to consider this set as a community since it is the desnsest 2-subgraph. This set is indeed a community according to our formulation and can be returned at $\beta = 1$ and $\alpha \in [2,6)$. □

B. Using maxflow algorithm

Since $f(C)$ is a submodular function, the minimization problem (2) can be solved using any submodular function minimization (SFM) algorithm. However, a generic SFM algorithm is computationally expensive. Below we reduce the problem of finding the candidate of $t \in V$ at any α, β to the mincut problem of the following augmented graph.

Definition 4 Let α, β, and t be as in Definition 2 and let $s \notin V$ be some additional node. The (α, β, t)-augmented digraph is the digraph on $V \cup \{s\}$ whose edge weight $w_{\alpha,\beta,t} : (V \cup \{s\})^2 \to \mathbb{R}_{\geq 0}$ is defined as

$$w_{\alpha\beta t}(i,j) := \begin{cases} w(i,j), & i \in V, j \in V \backslash \{t\} \\ w(i,j) + \beta d_i, & i \in V, j = t \\ \alpha, & i = s, j \in V \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Theorem 3 *The candidate $C_{\alpha,t}$ is the unique inclusion-wise minimum set C such that $(\{s\} \cup V \backslash C, C)$ is an s–t mincut of the (α, β, t)-augmented digraph.* □

[1] The figure shows $f_\alpha(C)$ only for the relevant subsets $C \subseteq V$ achieving (9b), i.e., the candidates. In other words, all suppressed lines lie above the curve of $\hat{f}_t$ for all $t \in V$.

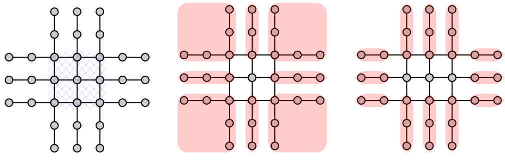


Fig. 3: An example graph that exhibits a grid-like center that connects to threads of nodes along the grid's perimeter. Communities and cut-clusters are highlighted using a crosshatch (blue) and no-pattern (red) marks, respectively. (a) Shows the returned community for $\alpha \in [\frac{31}{20}, \frac{27}{16})$ and $\beta = 0.7$, and (b) the cut-clusters for $\alpha \in [\frac{1}{8}, \frac{1}{2})$ and (c) the cut-clusters for $\alpha \in [\frac{1}{2}, 1)$. There are no other non-trivial (i.e., the entire set) solutions.

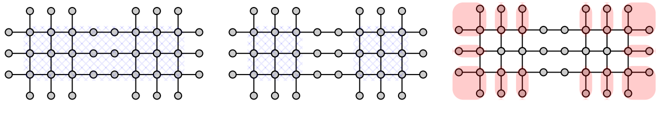


Fig. 4: Similar to Fig. 3. (a) Shows the returned community for $\alpha \in [\frac{37}{20}, \frac{58}{25})$ and $\beta = 0.85$, (b) the communities for $\alpha \in [\frac{58}{25}, \frac{75}{32})$ and $\beta = 0.85$, and (c) the cut-clusters for $\alpha \in [\frac{4}{41}, 1)$. There are no other, non-trivial, solutions.

PROOF (SKETCH) The theorem follows by showing that for any $C \subseteq V : t \in C$, the incut function of the augmented graph evaluated at C is equal to $f(C) + \alpha \cdot |C|$ plus some constant independent of C, i.e., (2) is equivalent to the s–t mincut problem on the augmented graph. ■

For a given α and β, the set $\mathcal{C}$ of communities can be computed as follows:

1) Run the maxflow algorithm n times, namely, for each $t \in V$ run the maxflow algorithm on the (α, β, t)-augmented graph to obtain $\hat{f}_t(\alpha)$ and its unique minimum solution $C_{\alpha,t}$.
2) Compute $\hat{f}(\alpha)$ as the minimum $\hat{f}_t(\alpha)$ over $t \in V$, and retain only the associated non-singleton $C_{\alpha,t}$ for $\mathcal{S}_\alpha$.

The second step can run in linear time and so the first step gives the overall complexity.

To compute the communities for all $\alpha \geq 0$, we can use the parametric maxflow algorithm of [11] to compute $\hat{f}_t(\alpha)$ for all $\alpha \geq 0$. The running time of the parametric algorithm is indeed the same as that of the push-relabel maxflow algorithm. Hence, we only need n maxflow computations on the augmented graphs to obtain $\hat{f}_t(\alpha)$ and $C_{\alpha,t}$ for all $\alpha \geq 0$ and $t \in V$. We can then compute $\hat{f}(\alpha)$ as the minimum $\hat{f}_t(\alpha)$ over $t \in V$, and retain the associated non-singleton $C_{\alpha,t}$ for $\mathcal{S}_\alpha$. This step can be computed in $O(n^2)$ because there are at most n line segments for each $\hat{f}_t(\alpha)$ and $\hat{f}(\alpha)$.

VI. Experimental results

To illustrate the benefits of the proposed communities over cut-clustering for larger networks than those in Fig. 1, we implemented and tested both algorithms on the graphs in Figs. 3 and 4. The graph in Fig. 3 contains a grid-like center, with peripheral-like chains of nodes attached to the grid's perimeter. The graph in Fig. 4 is similarly constructed, but with two grid-like centers. In both figures, it is desirable to differentiate the (denser) grid-like centers from the (sparser) peripherals.

The desired center is identified as a community in Fig. 3a, while cut-clustering only returns undesirable solutions, Figs. 3b and 3c. (See the figure's caption for details.) Note that the desired center is not a web community since each of its four corner nodes does not satisfy (1), which demonstrates the benefit of choosing $\beta > 0$. This also explains the undesirable behaviour of cut-clustering since a cut-cluster allows at most one of its members to violate (1) [5, Lemma 3.1]. (See Appendix A for more details.) Similar observations can be made using Fig. 4, where it is not hard to see that the same issues extend to larger, and other types of, graphs. More experimental results on synthetic and real-world data sets can be found in Appendix B.

References

[1] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics Reports*, vol. 659, pp. 1–44, 2016.

[2] G. W. Flake, S. Lawrence, and C. L. Giles, "Efficient identification of web communities," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 150–160.

[3] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.

[4] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p. 066133, 2004.

[5] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis, "Graph clustering and minimum cut trees," *Internet Mathematics*, vol. 1, no. 4, pp. 385–408, 2004.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[7] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[8] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[9] C. Chan, A. Al-Bashabsheh, D. S. H. Tam, and C. Zhao, "Finding better web communities in digraphs via max-flow min-cut," GitHub repository: https://github.com/ccha23/Better-Web-Communities, 2019.

[10] S. Fujishige, *Submodular functions and optimization*, 2nd ed. Elsevier, 2005.

[11] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan, "A fast parametric maximum flow algorithm and applications," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 30–55, 1989.

[12] K. Nagano, Y. Kawahara, and K. Aihara, "Size-constrained submodular minimization through minimum norm base," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 977–984.

[13] S. Fujishige, "Lexicographically optimal base of a polymatroid with respect to a weight vector," *Mathematics of Operations Research*, vol. 5, no. 2, pp. 186–196, 1980.

[14] ——, "Theory of principal partitions revisited," in *Research Trends in Combinatorial Optimization*. Springer, 2009, pp. 127–162.

[15] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[16] M. Rosvall, D. Axelsson, and C. T. Bergstrom, “The map equation,” *The European Physical Journal Special Topics*, vol. 178, no. 1, pp. 13–23, 2009.

[17] S. Fortunato and M. Barthelemy, “Resolution limit in community detection,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[18] A. Lancichinetti and S. Fortunato, “Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities,” *Physical Review E*, vol. 80, no. 1, p. 016118, 2009.

APPENDIX A
COMPARISON WITH CUT-CLUSTERS

In this section we compare our formulation to cut-clusters, which were introduced in [5] as a tractable alternative to web communities. The following proposition relates the cut-clusters to the candidates of Definition 3. We start by providing an explicit characterization of cut-clusters, which is absent from [5].

Proposition 3 *When the graph is undirected and $\beta = 0$, then, for $\alpha \geq 0$, the set of inclusion-wise maximal candidates, with the singletons removed, i.e.,*

$$\operatorname{maximal}\{C_{\alpha,t} \mid t \in V\} \setminus \{\{i\} \mid i \in V\}, \tag{17}$$

is equal to the set of cut-clusters returned by the cut-clustering algorithm [5, Fig. 2]. □

PROOF We first define the cut-clusters more precisely as follows. Consider an undirected graph and its α-augmented graph in Definition 4 but with additional edges (t, s) of weight α for $t \in V$. Note that the additional edges (t, s) does not contribute to any s–t cut and so all the s–t cut values remain unchanged. The purpose of the additional edges is to turn the augmented graph into an undirected augmented graph so that a Gomory-Hu tree (of the augmented graph) is well-defined. Let T be a Gomory-Hu tree of the undirected augmented graph and $\mathcal{P}$ be the set of connected components of T after removing s. Recall from [5] and the end of Section II that the set of cut clusters at threshold α is the set of non-singleton elements in $\mathcal{P}$. Since the Gomory-Hu tree may not be unique, the elements of $\mathcal{P}$ are further required to be inclusion-wise minimal.

We will show that the cut clusters are given by (17). In particular, we will show that

$$\mathcal{P} \subseteq \operatorname{maximal}\{C_{\alpha,t} \mid t \in V\}, \tag{18}$$

which implies the desired result trivially. By the properties of Gomory-Hu tree, for any $t \in V$,

1) the s–t mincut value $\hat{f}_t(\alpha)$ of the augmented graph can be obtained as the minimum weight of an edge in the unique path between s and t in T, and
2) the connected components of T after removing the minimum weight edge are the cut sets $V \setminus C$ and C where C is an optimal solution to $\hat{f}_t(\alpha)$ in (9b).

For simplicity, consider the case where the solution to $\hat{f}_t(\alpha)$ in (9b) is uniquely equal to $C_{\alpha,t}$ for all $t \in V$. Let s be the root of T. It is readily seen that $C_{\alpha,t}$ corresponds to a proper subtree of T by the above properties. Furthermore, each element of $\mathcal{P}$ is equal to $C_{\alpha,t}$ for a neighbor t of s in T, since the path from s to t in T contain only the edge (s, t). This implies (18) because $\mathcal{P}$ corresponds to the maximal proper subtrees of T.

Finally, consider the remaining case where the solution to $\hat{f}_t(\alpha)$ in (9b) may not be unique. It suffices to argue that a Gomory-Hu tree exists where the value of C in Property 2 above is the minimal solution $C_{\alpha,t}$. We will show that it is possible to increase α slightly to ensure uniqueness without changing the minimal solutions to $\hat{f}_t(\alpha)$. More precisely, let $\mathcal{C}_{\alpha,t}$ be the set of solutions to $\hat{f}_t(\alpha)$. Consider any $\alpha' > \alpha$. Since $f_\alpha(C)$ is increasing in α and C, we have

$$C \notin \mathcal{C}_{\alpha',t} \quad \forall t \in V, C \in \mathcal{C}_{\alpha,t} \setminus \{C_{\alpha,t}\}.$$

However, since $f_\alpha(C)$ is continuous in α, it is possible to choose α' sufficiently close to α such that

$$\mathcal{C}_{\alpha',t} = \{C_{\alpha,t}\} \quad \forall t \in V,$$

i.e., $C_{\alpha,t}$ remains optimal and no other s–t cuts become optimal as we go from α to α'. Any Gomory-Hu tree of the α'-augmented graph with edge weight $\hat{f}_t(\alpha')$ replaced by $\hat{f}_t(\alpha)$ satisfy Property 1 and 2. Hence, it can be used as the desired Gomory-Hu tree. ■

Example 5 Consider the undirected graph in Fig. 1d. The web communities are listed in (13). Proposition 3 asserts that the cut-clusters are specified by $\hat{f}_t(\alpha)$ for $t \in V$ (solid curves) in Fig. 2a as

$$\begin{cases} \emptyset, & \alpha \geq 2 \\ \{0,1\}, \{2,3\}, & \alpha \in [1.5, 2) \\ \{0,1,2,3\}, & \alpha < 1.5 \end{cases}, \tag{19}$$

i.e., they are obtained by choosing the maximal elements in each row of (14). Note that, in contrast to communities at $\beta = 0$, a cut-cluster may not be web community, e.g., for the cut-cluster $\{0,1\}$, node 1 has more connections with the external node 2 than it has with the member node 0. This contrast between cut-clusters and communities at $\beta = 0$ can be seen by comparing (6a) and [5, Lemma 3.1]. Namely, while (6a) guarantees a positive gap between the internal and external influences for every member of the community, [5, Lemma 3.1] guarantees this for every member of the cut-cluster except possibly one member. Finally, note that cut-clustering fails to find the web community $\{1,2\}$ since the formulation lacks the ability to reward internal connections, see Example 4. □

In addition to their closer relation to web communities (as observed in the previous example) communities at $\beta = 0$ also come with stronger quality guarantees compared to cut-clusters. This follows by observing that (8a) (ii) provides a stronger guarantee than the bound on the expansion in [5, Theorem 3.3]

$$\alpha \leq \min_{B \subsetneq C : |B| \geq 1} \frac{w(C \setminus B, B)}{\min\{|C \setminus B|, |B|\}}. \tag{20}$$

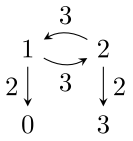


Fig. 5: An example showing the difficulty of extending cut-clusters to digraphs.

More precisely, for undirected graphs, (8a) (ii) implies

$$\begin{aligned}\alpha &< \min_{B\subsetneq C:|B|\geq 1}\left\{\frac{w(C\setminus B,B)}{|C\setminus B|},\frac{w(B,C\setminus B)}{|B|}\right\}\\ &= \min_{B\subsetneq C:|B|\geq 1}\frac{w(C\setminus B,B)}{\max\{|C\setminus B|,|B|\}}.\end{aligned} \tag{21}$$

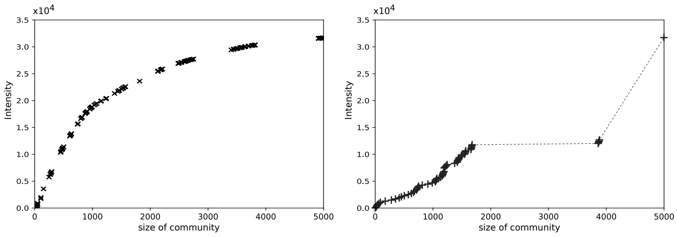
This is the same as (20) except that the minimization in the denominator is replaced by maximization, which gives a potentially better bound.

Finally, in contrast to Theorem 1 which can be extended to a general submodular function, the hierarchical structure for cut-clusters in [5, Lemma 3.9] relies on the specific formulation using the undirected graph cut. In particular, the result does not extend to digraphs where the incut function is not symmetric. For instance, consider the digraph in Fig. 5, which can be obtained from the graph in Fig. 1d by setting $w(0,1) = w(3,2) = 0$. With $\alpha \in [0,1)$ and $\beta = 0$, it can be shown that $C_{\alpha,t}$ for $t = 0$ and 3 are $\{0,1,2\}$ and $\{1,2,3\}$ respectively, and that they are both maximal (over $t \in \{0,1,2,3\}$) for the specified range of α values. However, they are neither disjoint nor subsets of each other.

Appendix B
Comparison with other formulations

In this section we compare our formulation to some existing ones. To simplify the discussions, we rely on simple examples to demonstrate some of the common issues that arise in graph clustering, and to point out similarities / differences between our method and some existing ones. We also present some experimental results on synthetic and real-world data sets to verify that our observatoins extend beyond demonstrating examples.

The work of [12] uses the idea of principle sequence [10, 13, 14] to give a polynomial-time partial solution to the NP-hard size-constrained submodular function minimization problem. When applied to graphical networks, the problem reduces to the densest k-subgraph problem, and the proposed solution can be used to identify a portions of the densest k-subgraphs. In this case the formulation in [12] is similar to (2) with $\beta = 1$ and the constraint $|C| \geq 1$ absent from (2a), i.e., we exclude the empty set and [12] does not. It can be shown that this exclusion allows our formulation to fined more solutions to the densest k-subgraph problem. For instance, in the graph in Fig. 1d, the densest 2-subgraph is $\{1,2\}$, which can be obtained using our algorithm with $\beta = 1$ and $\alpha \in [4,6)$, see Example 4, but cannot be obtained using the formulation in [12] since the empty set achieves a minimum cost of zero over that range of α. For the same reason the formulation in [12] cannot find any community when f is chosen to be the in-cut function, i.e., $\beta = 0$, since the empty set is trivially the optimal solution with zero size and zero in-cut.



(a) Our approach with $\beta = 1$ (b) SSM algorithm. ([12])

Fig. 6: Intensity (i.e., total weight of internal edges) versus the size (number of vertices) of the subgraphs returned by (a) our approach and (b) by [12].

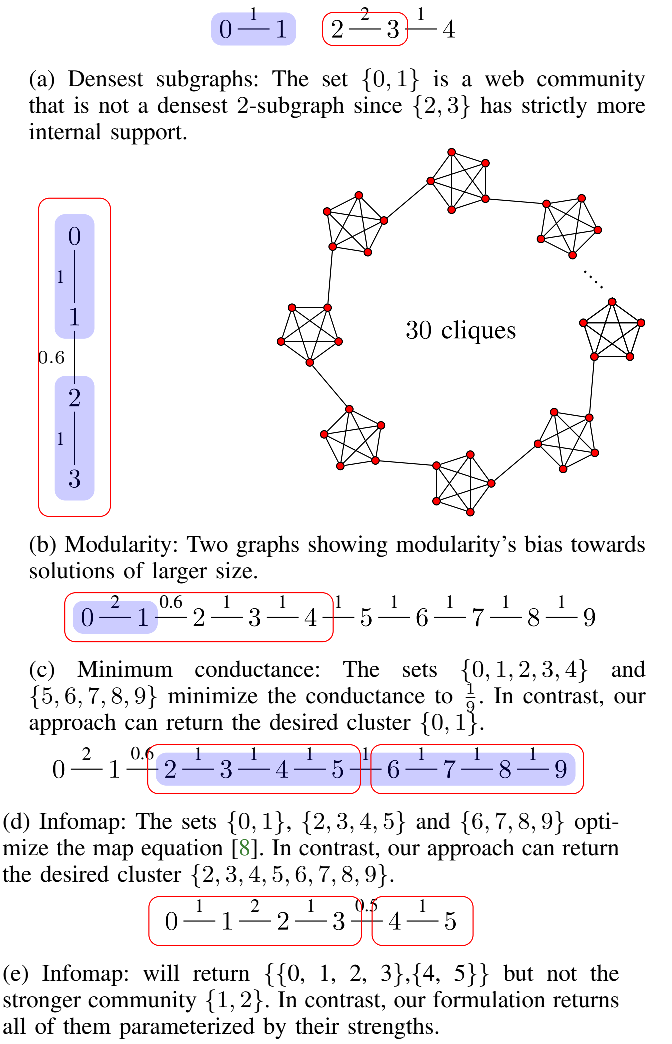
The observation above extends to larger graphs, which is demonstrated in Fig. 6. In this experiment, we applied our algorithm with $\beta = 1$ to the social-network data set *cnr-2000*, which is a directed unweighted graph. We preprocessed the network in the same way as [12]. By comparing the result of our algorithm to their proposed SSM algorithm, we are able to retrieve 51 dense communities with cardinality larger than 2000, while [12] reported that their SSM algorithm can only find two such communities.

We pause here and remark that a web community is not necessarily a densest subgraph.[2] For instance, consider the graph in Fig. 7a. The set $\{0,1\}$ is a web community that is returned by our method, but it is not a densest 2-subgraph since the subgraph $\{2,3\}$ has strictly more internal connections. Hence, any graph clustering algorithm that only returns densest subgraphs will fail to return such a web community.

The modularity [15], Infomap [16], and conductance minimization [6] formulations lead to NP-hard problems, and so rely on approximations. Modularity optimization, despite its popularity, has some limitations as pointed out in [17]. One limitation is the resolution limit where the modularity score tends to favour larger communities. For instance, for the graph in Fig. 7b (left), the modularity score is minimized by the entire set, and so it fails to detect the more meaningful communities $\{0,1\}$ and $\{2,3\}$, which can be identified using our formulation. Another example is the ring of cliques [17] shown in 7b (right) for 30 cliques each of size 5. In this case, the modularity method groups two consecutive cliques as one cluster, while our formulation can identify every clique. Methods that are based on conductance minimization may also suffer since they tend to favour solutions of relatively equal size.[3] An example graph demonstrating this is shown in Fig. 7c. A similiar issue is faced by Infomap as shown in Figs. 7d and 7e.

[2] Not to be confused with our discussion in an earlier section where we remarked that a densest subgraph is not necessarily a web community, e.g., see Fig. 1e.

[3] While this is desirable in some applications, it can also lead to some undesirable solutions.



(a) Densest subgraphs: The set $\{0,1\}$ is a web community that is not a densest 2-subgraph since $\{2,3\}$ has strictly more internal support.

(b) Modularity: Two graphs showing modularity's bias towards solutions of larger size.

(c) Minimum conductance: The sets $\{0,1,2,3,4\}$ and $\{5,6,7,8,9\}$ minimize the conductance to $\frac{1}{9}$. In contrast, our approach can return the desired cluster $\{0,1\}$.

(d) Infomap: The sets $\{0,1\}$, $\{2,3,4,5\}$ and $\{6,7,8,9\}$ optimize the map equation [8]. In contrast, our approach can return the desired cluster $\{2,3,4,5,6,7,8,9\}$.

(e) Infomap: will return $\{\{0,\ 1,\ 2,\ 3\}$,$\{4,\ 5\}\}$ but not the stronger community $\{1,2\}$. In contrast, our formulation returns all of them parameterized by their strengths.

Fig. 7: Examples of weighted graphs where our formulation (highlighted in blue) can return more meaningful communities than existing methods (circulated in red). (Not all communities are highlighted / listed.)

Finally, Fig. 8 shows that many existing methods fail to find strong communities (5) of large sizes as compared to our method. (See the figure's caption for details.) The experiment was conducted on a random graph generated according to the LFR benchmark with $\mu = 0.1$ [18]. However, the LFR model generates graphs that are geared towards partitional clustering methods, and so we expect even more discrepancies when testing on real-world networks that are hierarchical in nature.

Appendix C
Proofs

Proof (Proposition 1) For all $i \in C$, we have $C\backslash\{i\}$ is a feasible solution to (2) since $|C| \geq 2$, and so by the optimality of C

$$f(C)+\alpha\cdot|C| < f(C\backslash\{i\})+\alpha\cdot|C\backslash\{i\}|,$$

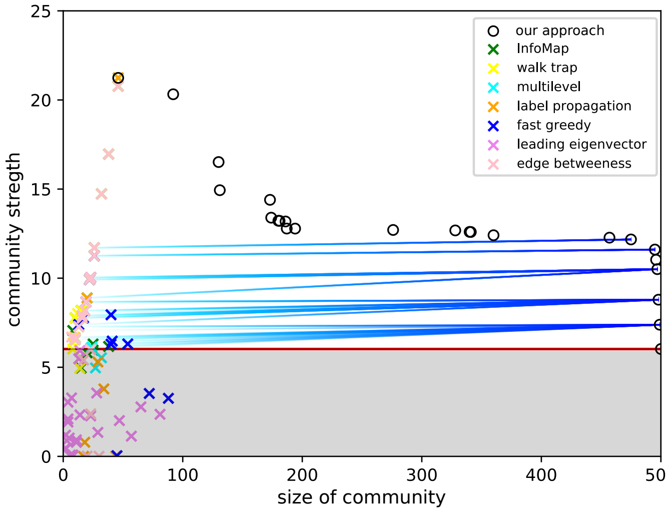


Fig. 8: Strength versus size plots showing existing algorithms fail to give strong communities of large sizes. We use an LFR benchmark network [18] generated with $\mu = 0.1$. Each black circle is a community obtained by our method at $\beta = 0.9$ and each X-mark is a community obtained by another method. A blue line connects an X-mark to a circle if the community of former has weaker strength and is contained in the latter. All points below the red line (shaded in grey) are dominated by the trivial community consisting of all the nodes.

where the inequality is strict by the minimality of C. Hence,

$$\begin{aligned}\alpha &< f(C)-f(C\backslash\{i\})\\ &=(1-\beta)\cdot\big[w(V\backslash C,i)-w(i,C)\big]-\beta\cdot\big[w(C,i)+w(i,C)\big]\\ &=w(V\backslash C,i)-w(i,C)-\beta\cdot w(V,i),\end{aligned}$$

which gives (6a) as desired. (6b) can be proved similarly by considering $C\cup\{i\}$ for i in $V\setminus C$.

Proof (Theorem 1) When $\alpha_1 \geq \alpha_2$, suppose to the contrary that

$$C_1 \not\subseteq C_2 \quad \text{and} \quad C_1\cap C_2 \neq \emptyset. \tag{22}$$

By the submodularity (10) of f, we have

$$f(C_1)-f(C_1\cap C_2) \geq f(C_1\cup C_2)-f(C_2) \tag{23}$$

Since $\alpha_1 \geq \alpha_2$, and $|C_1|-|C_1\cap C_2| = |C_1\cup C_2|-|C_2| \geq 0$, we also have

$$\alpha_1\left[|C_1|-|C_1\cap C_2|\right] \geq \alpha_2\left[|C_1\cup C_2|-|C_2|\right]. \tag{24}$$

Adding (24) to (23) side by side, we have

$$f_{\alpha_1}(C_1)-f_{\alpha_1}(C_1\cap C_2) \geq f_{\alpha_2}(C_1\cup C_2)-f_{\alpha_2}(C_2).$$

Note that the l.h.s. is strictly smaller than 0 because C_1 is a minimal solution to (2) for $\alpha = \alpha_1$. Since C_2 is optimal for $\alpha = \alpha_2$, The r.h.s. is at least 0, which is the desired contradiction.

When $C_1 \subsetneq C_2$, since C_1 is the minimal solution to (2) for $\alpha = \alpha_1$, we have

$$f_{\alpha_1}(C_1) \leq f_{\alpha_1}(C_2),$$

which means

$$f(C_1) + \alpha_1 |C_1| \leq f(C_2) + \alpha_1 |C_2|. \tag{25}$$

Since C_2 is the minimal solution to (2) for $\alpha = \alpha_2$, we have

$$f_{\alpha_2}(C_2) < f_{\alpha_2}(C_1),$$

which means

$$f(C_2) + \alpha_2 |C_2| < f(C_1) + \alpha_2 |C_1|. \tag{26}$$

Adding (25) to (26) side by side, we have

$$\alpha_1 \left[|C_1| - |C_2|\right] < \alpha_2 \left[|C_1| - |C_2|\right].$$

Since $|C_1| - |C_2| < 0$, so we get $\alpha_1 > \alpha_2$.

PROOF (THEOREM 2) By the optimality of C, we have

$$f(C) + \alpha \cdot |C| \leq f(C') + \alpha \cdot |C'|$$

for all $C' \subseteq V : |C'| \geq 1$, where, by the minimality of C, the inequality is strict if $C' \subsetneq C$. ■

PROOF (THEOREM 3) Let C be any s–t cut of the augmented graph, then from the definition of the augmented digraph, we have
$w(\{s\} \cup V \backslash C, C)$

$$= w(V \backslash C, C) + \beta \sum_{i \in V \backslash C} w(V \backslash \{i\}, \{i\}) + \alpha \cdot |C| \tag{27}$$

$$\begin{aligned} &= f(C) + \alpha \cdot |C| + \beta \big[w(C, C) \\ &\qquad + w(V \backslash C, C) + \sum_{i \in V \backslash C} w(V \backslash \{i\}, \{i\}) \big] \end{aligned} \tag{28}$$

$$= f(C) + \alpha \cdot |C| + \beta \sum_{i \in V} w(V \backslash \{i\}, \{i\}), \tag{29}$$

where the last equality follows by noting that $\sum_{i \in C} w(V \backslash \{i\}, \{i\}) = w(C, C) + w(V \backslash C, C)$. ■

APPENDIX D
DETAILED CALCULATIONS

This section details the calculations for the candidates, communities, and cut-clusters in Examples 4 and 5. It is helpful to point out the underpinning mathematical structure of the community candidate $C_{\alpha,t}$, namely the principle sequence of a submodular function [13] [14].

Proposition 4 *Given $\beta \in [0, 1]$ and $t \in V$, the candidate $C_{\alpha,t}$ of any $t \in V$ and $\alpha \in \mathbb{R}$ can be characterized as*

$$C_{\alpha,t} = \begin{cases} C_0 & \alpha \geq \alpha_1 \\ C_\ell & \alpha \in [\alpha_{\ell+1}, \alpha_\ell), \ell \in \{1, \ldots, N-1\} \\ C_N & \alpha < \alpha_N \end{cases} \tag{30}$$

for some non-negative integer N and sequences

$$\infty > \alpha_1 > \alpha_2 > \cdots > \alpha_N \geq 0 \tag{31}$$

$$\{t\} = C_0 \subsetneq C_1 \subsetneq C_2 \subsetneq \cdots \subsetneq C_N = \{V\}. \tag{32}$$

(The integer N and the two sequences depend on t and β.)□

It follows from Theorem 2 that:

Corollary 1 *Given $\beta \in [0, 1]$ and $t \in V$, with $C_0 = \{t\}$, we have for all $l \in \{1, \ldots, N\}$*

$$\alpha_\ell = \max_{C \subseteq V : C \supsetneq C_{\ell-1}} \frac{f(C_{\ell-1}) - f(C)}{|C \setminus C_{\ell-1}|} \tag{33}$$

and C_ℓ is the inclusion-wise maximum solution. □

Hence, α_ℓ and C_ℓ (i.e., the candidates together with the α values at which they change) can be computed successively by solving (33). For an algorithm that makes the computations in polynomial time, we refer the reader to [12].

We remind the reader that Examples 4 and 5 consider the graph in Fig. 1d. The desired communities, for $\beta = 0$ and 1, and the cut-clusters can be obtained by first computing the candidates and then invoking Propositions 2 and 3, respectively. To compute the candidates and the α values at which the candidates change, we rely on Corollary 1. Let us start with $\beta = 0$:
For $t = 0$, we have

$$\begin{aligned} \alpha_1 &= \max_{C \subseteq V : C \supsetneq \{0\}} \frac{w(V \backslash \{0\}, \{0\}) - w(V \backslash C, C)}{|C| - 1} \\ &= \max_{C \subseteq V : C \supsetneq \{0\}} \frac{2 - w(V \backslash C, C)}{|C| - 1}, \end{aligned}$$

which results in $\alpha_1 = \frac{2}{3}$ and the maximization is uniquely achieved by V, i.e., we have

$$C_{\alpha,0} = \begin{cases} \{0\}, & \alpha \geq \frac{2}{3} \\ \{0,1,2,3\}, & \alpha < \frac{2}{3} \end{cases} \text{ and } \hat{f}_0(\alpha) = \begin{cases} \alpha + 2, & \alpha \geq \frac{2}{3} \\ 4\alpha, & \alpha < \frac{2}{3}. \end{cases}$$

For $t = 1$, we have

$$\begin{aligned} \alpha_1 &= \max_{C \subseteq V : C \supsetneq \{1\}} \frac{w(V \backslash \{1\}, \{1\}) - w(V \backslash C, C)}{|C| - 1} \\ &= \max_{C \subseteq V : C \supsetneq \{0\}} \frac{5 - w(V \backslash C, C)}{|C| - 1}, \end{aligned}$$

which results in $\alpha_1 = 2$, uniquely achieved by $\{0, 1\}$. In the next succession, we have

$$\begin{aligned} \alpha_2 &= \max_{C \subseteq V : C \supsetneq \{0,1\}} \frac{w(V \backslash \{0,1\}, \{0,1\}) - w(V \backslash C, C)}{|C| - 2} \\ &= \max_{C \subseteq V : C \supsetneq \{0,1\}} \frac{3 - w(V \backslash C, C)}{|C| - 2}, \end{aligned}$$

which results in $\alpha_2 = 1.5$, achieved uniquely by $\{0, 1, 2, 3\}$, i.e., we have

$$C_{\alpha,1} = \begin{cases} \{1\}, & \alpha \geq 2 \\ \{0,1\}, & \alpha \in [\frac{3}{2}, 2) \\ \{0,1,2,3\}, & \alpha < 1.5. \end{cases}, \hat{f}_1(\alpha) = \begin{cases} \alpha + 5, & \alpha \geq 2 \\ 2\alpha + 3, & \alpha \in [\frac{3}{2}, 2) \\ 4\alpha, & \alpha < 1.5\ . \end{cases}$$

By symmetry, given β, we have for $t = 2$ and 3, $C_{\alpha,t} = \{3 - j \mid j \in C_{\alpha,3-t}\}$ (this set is empty by convention if $C_{\alpha,3-t}$

is empty) and $\hat{f}_t = \hat{f}_{3-t}$. Hence, the candidates are as listed in (14).

Let us first obtain the cut-clusters. By Proposition 3, taking the inclusion-wise maximal candidates for $\beta = 0$ gives the cut clusters in (19). (In Fig. 2a, this corresponds to drawing a vertical line at α, reading out the candidates $C_{\alpha,t}$ that correspond to the intersection of this vertical line with the solid lines in the figure, and finally taking the maximal among such candidates.)

To obtain the communities at $\beta = 0$, we use Proposition 2, or equivalently, compute $\hat{f}(\alpha)$ from $\hat{f}_t(\alpha)$, which gives

$$\hat{f}(\alpha) = \begin{cases} \alpha + 2, & \alpha \geq 1, \quad \text{solved by } \{0\}, \{3\} \\ 4\alpha, & \alpha < 1, \quad \text{solved by } \{0, 1, 2, 3\}. \end{cases}$$

resulting in the communities in (15). (The ones labeled with $\beta = 0$.)

Next we consider $\beta = 1$. To obtain the communities for $\beta = 1$, we repeat the same calculations that were carried out when computing the communities at $\beta = 0$, as summarized below

$$C_{\alpha,0} = \begin{cases} \{0\}, & \alpha \geq 5 \\ \{0,1,2\}, & \alpha \in [4,5) \\ \{0,1,2,3\}, & \alpha < 4. \end{cases}, \hat{f}_0(\alpha) = \begin{cases} \alpha, & \alpha \geq 5 \\ 3\alpha - 10, & \alpha \in [4,5) \\ 4\alpha - 14, & \alpha < 4 \end{cases}.$$

$$C_{\alpha,1} = \begin{cases} \{1\}, & \alpha \geq 6 \\ \{1,2\}, & \alpha \in [4,6) \\ \{0,1,2,3\}, & \alpha < 4. \end{cases}, \hat{f}_1(\alpha) = \begin{cases} \alpha, & \alpha \geq 6 \\ 2\alpha - 6, & \alpha \in [4,6) \\ 4\alpha - 14, & \alpha < 4 \end{cases}.$$

Hence,

$$\hat{f}(\alpha) = \begin{cases} \alpha, & \alpha \geq 6, \quad \text{solved by } \{0\}, \dots, \{3\} \\ 2\alpha - 6, & \alpha \in [4, 6), \quad \text{solved by } \{1, 2\} \\ 4\alpha - 14, & \alpha < 4, \quad \text{solved by } \{0, 1, 2, 3\}. \end{cases}$$