

# Game Theoretic Clustering for Finding Strong Communities

Chao Zhao <sup>1</sup>, Ali Al-Bashabsheh <sup>2</sup> and Chung Chan <sup>1,\*</sup><sup>1</sup> Department of Computer Science, City University of Hong Kong, Hong Kong, China; ao.ao@my.cityu.edu.hk<sup>2</sup> School of General Engineering, Beihang University, Beijing 100191, China; entropyali@gmail.com

\* Correspondence: chung.chan@cityu.edu.hk

**Abstract:** We address the challenge of identifying meaningful communities by proposing a model based on convex game theory and a measure of community strength. Many existing community detection methods fail to provide unique solutions, and it remains unclear how the solutions depend on initial conditions. Our approach identifies strong communities with a hierarchical structure, visualizable as a dendrogram, and computable in polynomial time using submodular function minimization. This framework extends beyond graphs to hypergraphs or even polymatroids. In the case when the model is graphical, a more efficient algorithm based on the max-flow min-cut algorithm can be devised. Though not achieving near-linear time complexity, the pursuit of practical algorithms is an intriguing avenue for future research. Our work serves as the foundation, offering an analytical framework that yields unique solutions with clear operational meaning for the communities identified.

**Keywords:** game theory; community detection; hierarchical clustering

## 1. Introduction

Community detection is a fundamental problem in various fields, such as biological study and social network analysis. The definition of a community can vary based on the specific problem and objective at hand, but the definitions provided in [1–3] are generally considered widely accepted. In broad terms, a community is commonly understood as a group of individuals with stronger connections among its members than with individuals outside the group.

In the process of conducting community detection, real-world problems are typically translated into graphs where nodes represent individuals and edges represent relations. Numerous community detection methods have been developed based on diverse principles and objective functions. Surveys of community detection methods can be found in [4–10].

Game theory has emerged as a technique applied in community detection [8,11–15]. Its applications extend to identifying disjoint, overlapping, and hierarchical communities. As a systematic framework, game theory models and studies the decisions and outcomes of players in a game [16,17]. Broadly, game theory can be categorized into two main types: non-cooperative game theory and cooperative game theory. Non-cooperative game theory focuses on the competition between individual players, emphasizing their strategies and payoffs. Cooperative game theory, on the other hand, focuses on the cooperation between players and addresses the allocation of payoffs to players based on the worth of the coalitions formed. Within cooperative game theory, there are two main types: non-transferable utility cooperative games, where the payoff for a player within a coalition cannot be transferred to another player in the same coalition, and transferable utility cooperative games, where payoffs are considered transferable among players in the same coalition. Solution concepts such as the core, kernel, nucleolus, Shapley value, egalitarian, etc., play crucial roles in cooperative game theory [18,19].

The community detection method based on cooperative game theory typically identifies the coalition with the highest score determined by a measure evaluated on the coalitions. However, due to the use of approximations, non-unique results are common. Zhou et al. [20]



**Citation:** Zhao, C.; Al-Bashabsheh, A.; Chan, C. Game Theoretic Clustering for Finding Strong Communities. *Entropy* **2024**, *26*, 268. <https://doi.org/10.3390/e26030268>

Academic Editor: Boleslaw K. Szymanski

Received: 17 January 2024

Revised: 12 March 2024

Accepted: 15 March 2024

Published: 18 March 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

presented a community detection method using cooperative game theory and the Shapley value. The study focused on a social network where nodes are linked to relationships in various finite topics. The Shapley value represents a node's contribution to the connection closeness of a coalition. The algorithm forms hierarchical and overlapping coalitions by iteratively adding each node to one of the coalitions formed in the previous iteration, where the newly added node obtains the largest Shapley value. Despite running in polynomial time, the algorithm relies on approximation. Another related approach for overlapping and hierarchical community detection [21] also employs cooperative game theory, and the hierarchical structure of coalitions is obtained through a greedy agglomerative method, potentially yielding non-unique results.

The Naming Game [22–24] presents another game theoretic approach applicable to community detection, where the community structure emerges from the dynamic interactions between pairs of nodes within the game. However, empirical evidence suggests that the solution is generally not unique [24]. The convergence and computational costs of the method are analyzed through extensive empirical experiments [22,24], while it remains unclear regarding the theoretical bounds. Furthermore, the Naming Game relies on pairwise connections and does not capture higher-order statistics among nodes beyond pairwise relationships, therefore limiting its scope of applications in community detection.

We introduce a notion of strength derived from cooperative game theory to identify strong communities that are interpretable. Moreover, the strong communities are unique, computable in polynomial time with recursive procedures, and can be represented by a dendrogram. The scope of consideration encompasses a set of individuals with a super-modular function for evaluating the communities, which means our approach is applicable to community detection tasks beyond graphical models. Our framework focuses on elucidating the theoretical properties of the strong communities and can provide the foundation for future research on empirical algorithms for large-scale datasets.

This paper is structured as follows: In Section 2, we present the relevant concepts in cooperative games. Section 3 outlines the derivation of the objective function based on convex games. We also formulate the definitions for community strength and strong communities in this section. Moving on to Section 4, we delve into the discussion of the properties of strong communities, laying the foundation for their computation. Section 5 details the solution to the problem through submodular function minimization and, in certain cases, introduces the use of the max-flow min-cut algorithm as a more efficient method in practice. In Section 6, concrete examples are provided to demonstrate the computation of strong communities and the representation of the dendrogram of these communities. Finally, in Section 7, we conclude our work.

## 2. Cooperative Game

A cooperative game [17] is characterized by  $(V, g)$ , where

- $V$  is a finite set of players with  $|V| \geq 2$ , and
- $g : 2^V \rightarrow \mathbb{R}$  is a set function called the characteristic function, where  $g(C)$  is the worth of the coalition  $C \subseteq V$ , assuming players in  $C$  cooperate to form such coalition.

Denote the payoff allocation for the players as a vector

$$r_V = (r_1, r_2, \dots, r_{|V|}) \in \mathbb{R}^{|V|},$$

with  $r_i$  being the  $i$ -th element in  $r_V$  as the payoff allocated for  $i$ -th player.

The total payoff in the coalition  $C \subseteq V$  is denoted as

$$r(C) := \sum_{i \in C} r_i. \quad (1)$$

Furthermore, when  $g$  is a supermodular function, the game is called convex [17]. In this case, for  $\forall B, C \subseteq V$ ,

$$g(B) + g(C) \leq g(B \cup C) + g(B \cap C). \tag{2}$$

Or equivalently, for  $\forall B \subseteq C \subseteq V, i \in V \setminus B$ ,

$$g(B \cup \{i\}) - g(B) \leq g(C \cup \{i\}) - g(C), \tag{3}$$

where both sides are the increases in worth when a player  $i$  is added to a coalition. (3) means that the increase in worth, when a player adds to a coalition, is equal or larger than that for a larger superset coalition, i.e., the marginal worth is non-diminishing for convex games. For simplicity,  $g$  is thought to be normalized, i.e.,  $g(\emptyset) = 0$ .

As for the payoff allocation, the transferable utility is considered here, i.e., the payoffs can be transferred between players in the same coalition. The core [18] is one of the relevant solution concepts in cooperative games, which is about the feasible allocation of payoffs to players.

The core of a game  $(V, g)$  is defined as [17]:

$$\text{Core}(V, g) := \{r_V \in \mathbb{R}^{|V|} \mid r(V) = g(V), r(C) \geq g(C), \forall C \subseteq V\}. \tag{4}$$

In the definition of the core,  $r(V) = g(V)$  means the payoff allocation exactly splits the total worth of the grand coalition  $V$ . The inequality  $r(C) \geq g(C)$  says that no other coalition  $C \subsetneq V$  can have a worth larger than the payoff  $C$  can receive by cooperating in  $V$ , and hence will not deviate from the grand coalition  $V$ . The core can be viewed as the stable payoff allocation. For a convex game, the core is always nonempty [17,25].

### 3. Problem Formulation

By regarding the set  $V$  of nodes as players, we consider a convex game with the characteristic function  $g$  being a supermodular function on  $2^V$ .

In particular, consider a weighted digraph on the set  $V$  of nodes. Such a graph can be characterized by the weight of the directed edges described using the weight function  $w$ :

$$w(B, C) = \sum_{i \in B, j \in C} a_{ij},$$

where  $a_{i,j}$  is the weight of the edge from node  $i$  to node  $j$ . This covers the undirected graphs special cases when  $a_{ij} = a_{ji}$  for all  $i, j \in V$ .

Consider the function  $g$  defined in the form of

$$g(B) = \gamma \cdot \beta \cdot w(B, B) - \gamma \cdot (1 - \beta) \cdot w(V \setminus B, B), \tag{5}$$

where  $\beta \in [0, 1]$  and  $\gamma > 0$ . The function  $g$  in (5) is supermodular [26]. When  $\beta = 0$ , (5) reduces to the total weight of edges in  $B$  scaled by  $\gamma$ ; when  $\beta = 1$ , (5) reduces to the negative of the total weight of incoming edges from outside to  $B$  scaled by  $\gamma$ .

We want to identify strong communities based on the convex game using the following measure of community strength.

**Definition 1.** For  $C \subseteq V : |C| > 1$ , define

$$\sigma(C) := \min_{\substack{B \subseteq C \\ B \neq \emptyset}} \max_{r_C \in \text{Core}(C, g)} \min_{i \in B} r_i, \tag{6}$$

which is referred to as the strength of community  $C$ .

The inner maximization in (6) is the stable payoff guaranteed to any player in  $B$ , which we termed the community support to  $B$ . The outside minimization in (6) gives the strength of  $C$ , which is the minimum community support over  $B \subsetneq C : B \neq \emptyset$ .

The following example illustrates the interpretation of the strength in (6) more concretely.

**Example 1.** Consider the unweighted graph in Figure 1a with  $V = \{1, 2, 3\}$  and characteristic function  $g$  in (5) with  $\beta = 1$  and  $\gamma = 1$ , i.e., for  $B \subseteq V : B \neq \emptyset$ ,  $g(B) = \frac{1}{2}w(B, B)$ , which calculates the total number of internal edges inside  $B$ .

We are going to show how to obtain the strength of  $\{1, 2\}$ , which requires us to calculate the minimum community support over  $B \subsetneq \{1, 2\} : B \neq \emptyset$  according to (6). By definition, the community support to  $B$  from  $\{1, 2\}$  is the stable payoff that is guaranteed to each player in  $B$ . For a payoff allocation  $r_{\{1,2\}}$  to be stable, it should be in  $\text{Core}(\{1, 2\}, g)$ , which is calculated to be

$$\text{Core}(\{1, 2\}, g) = \{(r_1, r_2) \mid r_1 + r_2 = 1, r_1 \geq 0, r_2 \geq 0\}. \tag{7}$$

Then, we consider the guaranteed stable payoff to each player in  $B$ . For instance, when  $B = \{1\}$ , the guaranteed stable payoff to players in  $B$  is 1, which is achieved with the payoff allocation  $r_{\{1,2\}} = (1, 0)$ ; when  $B = \{2\}$ , the stable guaranteed payoff to players in  $B$  is 1, which is achieved with the payoff allocation  $r_{\{1,2\}} = (0, 1)$ . Therefore, we know that the minimum community support to any non-empty proper subset of  $\{1, 2\}$  is 1, i.e., the strength of  $\{1, 2\}$  is 1.

Additionally, there is 1 unit of payoff that is transferable between players 1 and 2 based on the constraint for the core. Such a transferable payoff tends to improve the guaranteed payoff for players in non-empty subsets of  $C$ . As a result,  $\{1, 2\}$  intuitively forms a meaningful community.

Similarly, we can show that the strength of  $V$  is 0. For a payoff allocation  $r_V$  to be stable, it has to be in  $\text{Core}(V, g)$  which is calculated to be (see Appendix A.1)

$$\text{Core}(V, g) = \{r_V := (r_1, r_2, r_3) \mid r_1 + r_2 \geq 1, r_1 \geq 0, r_2 \geq 0, r_3 = 0\}. \tag{8}$$

Then, we consider the community support to  $B \subsetneq V : B \neq \emptyset$ . For instance, when  $B = \{1\}$ , the community support to  $B$  is 1, which is achieved with the payoff allocation  $r_V = (1, 0, 0)$ . By enumeration over  $B \subsetneq V : B \neq \emptyset$ , we can obtain that when  $B = \{3\}$ , the community support to  $B$  is 0, which is the minimum value of such community support. Hence, the strength of  $V$  is 0.

There is another equivalent definition for  $\sigma(C)$  with (1) where we consider the average payoff allocated to a set  $B \subsetneq C : B \neq \emptyset$  instead of the inner minimization term  $\min_{i \in B} r_i$  in (1), as stated in the following result.

**Proposition 1.** For  $C \subseteq V : |C| > 1$ ,

$$\sigma(C) = \min_{\substack{B \subsetneq C \\ B \neq \emptyset}} \max_{r_C \in \text{Core}(C, g)} \frac{r(B)}{|B|}, \tag{9}$$

**Proof.** See Appendix A.2.  $\square$

Our goal is to identify strong communities defined using  $\sigma$  as follows.

**Definition 2.** For any threshold  $\alpha$ , define the collection of strong communities in  $V$  as

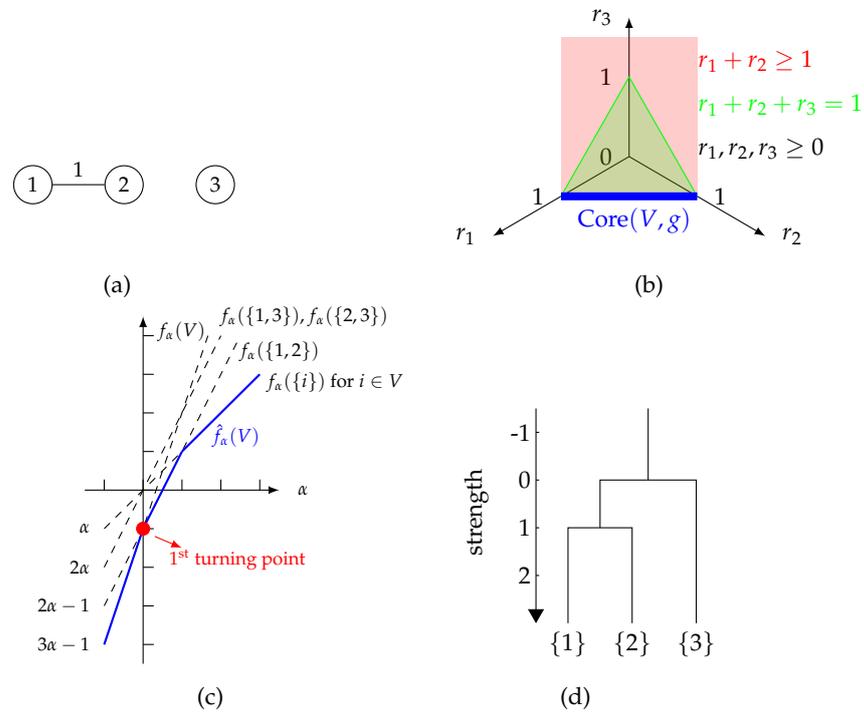
$$\mathcal{C}_\alpha(V) := \text{maximal}\{C \subseteq V \mid |C| > 1, \sigma(C) > \alpha\}. \tag{10}$$

The maximal  $\mathcal{F}$  means inclusion-wise maximal subsets in  $\mathcal{F}$ , i.e.,

$$\text{maximal}\mathcal{F} := \{B \in \mathcal{F} \mid \nexists C \in \mathcal{F}, B \subsetneq C\}. \tag{11}$$

Similarly, minimal  $\mathcal{F}$  means inclusion-wise minimal subsets in  $\mathcal{F}$ , i.e.,

$$\text{minimal}\mathcal{F} := \{B \in \mathcal{F} \mid \nexists C \in \mathcal{F}, B \supsetneq C\}. \tag{12}$$



**Figure 1.** An illustrative example of an unweighted graph with  $g(B) := \frac{1}{2}w(B, B)$  for  $B \subseteq V : B \neq \emptyset$ . (a) The unweighted graph; (b) Visualization of  $\text{Core}(V, g)$ ; (c) The curve  $\hat{f}_\alpha(V)$ ; (d) The dendrogram.

**Example 2.** In Example 1, we already get  $\sigma(V) = 0$ . Similarly, we can get  $\sigma(\{1, 2\}) = 1$ ,  $\sigma(\{1, 3\}) = 0$  and  $\sigma(\{2, 3\}) = 0$ .

According to (10), the strong communities in  $V$  given by our approach are

$$C_\alpha(V) = \begin{cases} \{V\}, & \alpha < 0, \\ \{\{1, 2\}\}, & \alpha \geq 0. \end{cases} \tag{13}$$

### 4. Main Results

#### 4.1. Characterization of Community Strength

The community strength defined in (9) takes a simpler form for the convex game as shown in Theorem 1.

**Theorem 1.** For any  $C \subseteq V : |C| > 1$ ,

$$\sigma(C) = \min_{\substack{B \subseteq C: \\ B \neq \emptyset}} \frac{g(C) - g(B)}{|C \setminus B|}, \tag{14}$$

Furthermore, the set of optimal solutions to (9) is given by

$$\{C \setminus B \mid B \in \mathcal{S}(C)\}, \tag{15}$$

where  $\mathcal{S}(C)$  is the set of optimal solutions to the minimization in (14).

**Proof.** See Appendix A.3. □

Equation (14) is the basic formula of community strength that we will utilize to derive the properties of the strong communities and investigate how to calculate the strong subsets.

The following example shows the equivalent value of the strength of  $V$  calculated by (9) and (14).

**Example 3.** Consider  $V$  as in Example 1, follow (14) to calculate the strength of  $V$ ,

$$\sigma(V) = \min_{\substack{B \subseteq V: \\ B \neq \emptyset}} \frac{g(V) - g(B)}{|C \setminus B|} \tag{16}$$

$$= 0 \tag{17}$$

with  $\mathcal{S}(V) = \{\{1, 2\}\}$ . The value of  $\sigma(V)$  calculated here according to (14) is consistent with that calculated in Example 1.

Define for  $\alpha \in \mathbb{R}$  and  $C \subseteq V : |C| > 1$  that

$$\hat{f}_\alpha(C) := \min_{\substack{B \subseteq C: \\ B \neq \emptyset}} f_\alpha(B), \quad \text{where} \tag{18}$$

$$f_\alpha(B) := \alpha|B| - g(B), \quad \text{for } B \subseteq V. \tag{19}$$

Denote the optimal solution set to (18) as  $\mathcal{B}_\alpha(C)$ , and the collection of inclusion-wise minimal sets among  $\mathcal{B}_\alpha(C)$  as  $\mathcal{B}_\alpha^*(C)$ , i.e.,

$$\mathcal{B}_\alpha^*(C) := \text{minimal } \mathcal{B}_\alpha(C). \tag{20}$$

$\mathcal{B}_\alpha(C)$  is the set that we use to analyze the relation between  $\sigma(C)$  and the curve  $\hat{f}_\alpha(C)$ , and  $\mathcal{B}_\alpha^*(C)$  is the set we use for showing the computation of  $\mathcal{C}_\alpha(V)$  in the latter part.

The following example shows the curve of  $\hat{f}_\alpha(V)$  for the set  $V$  in Example 1.

**Example 4.** Consider  $V$  as in Example 1, according to (18),

$$\hat{f}_\alpha(V) = \begin{cases} 3\alpha - 1, & \alpha < 0, \\ 2\alpha - 1, & 0 \leq \alpha < 1, \\ \alpha, & \alpha \geq 1, \end{cases} \tag{21}$$

and the inclusion-wise minimal solution set to (18) is given by

$$\mathcal{B}_\alpha^*(V) = \begin{cases} \{V\}, & \alpha < 0, \\ \{\{1, 2\}\}, & 0 \leq \alpha < 1, \\ \{\{i\} | i \in V\}, & \alpha \geq 1, \end{cases} \tag{22}$$

as illustrated in Figure 1c, where the result for  $\hat{f}_\alpha(V)$  and  $\mathcal{B}_\alpha^*(V)$  can be obtained directly after we draw every curve of  $f_\alpha(B)$  for  $B \subseteq V : B \neq \emptyset$ .

For instance,

- when  $\alpha \in (0, 1)$ ,  $\hat{f}_\alpha(V)$  is given by  $f_\alpha(\{1, 2\})$ , hence  $\mathcal{B}_\alpha^*(V) = \{\{1, 2\}\}$ .
- when  $\alpha = 0$ , both  $\{1, 2\}$  and  $V$  are solutions to (18) with respect to  $\hat{f}_\alpha(V)$ , while  $\{1, 2\}$  is the inclusion-wise minimal solution, hence  $\mathcal{B}_\alpha^*(V) = \{1, 2\}$ .

From (19), it can be seen that  $f_\alpha(B)$  is a linear function of  $\alpha$  with slope  $|B|$ . Therefore,  $\hat{f}_\alpha(C)$  in (18) is a piece-wise linear function since it is a minimization of linear functions. With  $C \subseteq V : |C| > 1$ , the curve must have at least one turning point since the slope of  $f_\alpha(\{i\}), i \in V$  is different from  $f_\alpha(C)$ . Figure 1c is the curve of  $\hat{f}_\alpha(V)$  for  $V$  in Example 1.

The following result shows that  $\sigma(C)$  can be obtained from the curve. It will be used for deriving the representation and computation of the strong communities defined in Definition 2.

**Proposition 2.** For the curve  $\hat{f}_\alpha(C)$  against  $\alpha \in \mathbb{R}$ :

(1)  $\sigma(C)$  is the  $\alpha$ -coordinate of the first turning point. More precisely,

$$\min_{\substack{B \subseteq C: \\ B \neq \emptyset}} f_\alpha(B) = f_\alpha(C) \iff \sigma(C) = \alpha, \tag{23}$$

$$\min_{\substack{B \subseteq C: \\ B \neq \emptyset}} f_\alpha(B) > f_\alpha(C) \iff \sigma(C) > \alpha. \tag{24}$$

(2) The collection  $\mathcal{B}_\alpha(C)$  of optimal solution to (18) satisfies

$$\mathcal{B}_\alpha(C) \not\supseteq C, \text{ for } \alpha > \sigma(C), \tag{25}$$

$$\mathcal{B}_\alpha(C) = \mathcal{S}(C) \cup \{C\}, \text{ for } \alpha = \sigma(C), \tag{26}$$

$$\mathcal{B}_\alpha(C) = \{C\}, \text{ for } \alpha < \sigma(C). \tag{27}$$

**Proof.** See Appendix A.4.  $\square$

The following example can further illustrate the property of  $\hat{f}_\alpha(V)$ .

**Example 5.** In Example 4, the first turning point of the curve  $\hat{f}_\alpha(V)$  is  $(0, -1)$ , whose  $\alpha$ -coordinate is exactly  $\sigma(V)$ .

#### 4.2. Representation of Strong Communities

The strong communities defined in Definition 2 form a hierarchy and can be represented by a dendrogram.

**Theorem 2.** For any  $C_1 \in \mathcal{C}_{\alpha_1}(V), C_2 \in \mathcal{C}_{\alpha_2}(V)$  where  $\alpha_1 \leq \alpha_2$ , we have

$$C_1 \supseteq C_2, \text{ or } C_1 \cap C_2 = \emptyset.$$

Furthermore,

$$\text{if } C_1 \supsetneq C_2, \text{ then } \alpha_1 < \alpha_2.$$

**Proof.** See Appendix A.6.  $\square$

The following example shows that the strong communities in Figure 1a as in Example 1 with respect to two different  $\alpha$ 's have a containment relationship.

**Example 6.** Let  $\alpha_1 := -1$  and  $\alpha_2 := 1$ . By the calculation results in Example 2,  $C_1 := V \in \mathcal{C}_{\alpha_1}(V)$  and  $C_2 := \{1, 2\} \in \mathcal{C}_{\alpha_2}(V)$ . Then  $C_1 \supsetneq C_2$ , which means the communities in  $\mathcal{C}_{\alpha_1}(V)$  are contained by those in  $\mathcal{C}_{\alpha_2}(V)$ . This shows the hierarchical structure of the strong communities with respect to the specific  $\alpha_1$  and  $\alpha_2$ .

Theorem 2 follows from the following lemma.

**Lemma 1.** For all  $C_1, C_2 \subseteq V : C_1 \cap C_2 \neq \emptyset$ ,

$$\sigma(C_1 \cup C_2) \geq \min\{\sigma(C_1), \sigma(C_2)\}. \tag{28}$$

**Proof.** See Appendix A.5.  $\square$

**Example 7.** As an example for showing Lemma 1, consider Figure 1a as in Example 1, let  $C_1 := \{1, 2\}$  and  $C_2 := \{2, 3\}$ , then  $C_1 \cap C_2 \neq \emptyset$ . By the calculation results in Example 2,

$$\sigma(C_1 \cup C_2) = 0 \geq \min\{\sigma(C_1), \sigma(C_2)\},$$

i.e., (28) holds for  $C_1$  and  $C_2$ .

Lemma 1 establishes that the strength of the union of any two overlapping non-empty sets is lower bound by the smaller strength among the two sets, and this is the basis for Theorem 2.

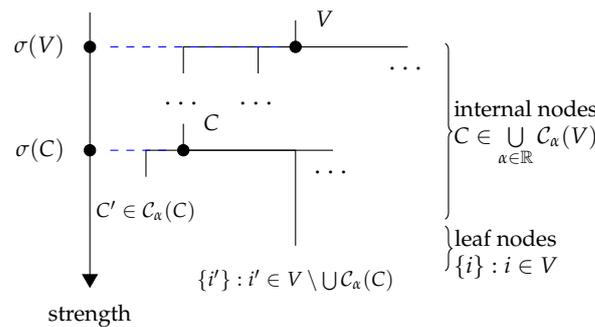
The family  $\bigcup_{\alpha \in \mathbb{R}} C_\alpha(V)$  is said to be laminar and can be shown to contain at most  $|V| - 1$  elements. More precisely, we will show that the family of communities, together with their levels of strength, can be represented by the following dendrogram with  $\sigma$ , meaning the cophenetic similarity.

**Definition 3.** The dendrogram for the set of communities is defined as follows:

- (1) Every  $C \in \bigcup_{\alpha \in \mathbb{R}} C_\alpha(V)$  is an internal node annotated with the value  $\sigma(C)$ ;
- (2) Every singleton  $\{i\}$  for  $i \in V$  is a leaf node (annotated with the value  $+\infty$ );
- (3) The parent of each node  $B \subsetneq V : B \neq \emptyset$  is defined as the minimum

$$\text{parent}(B) := \min\{C \in C_\alpha(V) \mid B \subsetneq C, \alpha \in \mathbb{R}\}. \tag{29}$$

As illustrated in Figure 2, the dendrogram forms a tree because each node (except the root node  $V$ ) has a unique parent node.



**Figure 2.** Dendrogram of the communities.

As a result of Theorem 2, the following corollary states that the parent of each strong community except  $V$  exists and is unique.

**Corollary 1.** For every  $B \subsetneq V : B \neq \emptyset$ , the minimum element  $\text{parent}(B)$  exists.

**Proof.** See Appendix A.7.  $\square$

Using the following result, we can show that the set of children for each node  $C \in \bigcup_{\alpha \in \mathbb{R}} C_\alpha(V)$  is

$$C_{\sigma(C)}(C) \cup \{\{i\} \mid i \in V \setminus \bigcup C_{\sigma(C)}(C)\},$$

which is also illustrated in Figure 2.

Analogous to Corollary 1, a community  $B$  has a parent  $C$  in the dendrogram if and only if  $B$  is in  $C_{\sigma(C)}(C)$ , and the strength of  $B$  is larger than that of  $C$ , as stated in the following corollary.

**Corollary 2.** For any nodes  $C \in \bigcup_{\alpha \in \mathbb{R}} C_\alpha(V)$  of the dendrogram,

$$\text{parent}(B) = C \iff B \in C_{\sigma(C)}(C), \tag{30}$$

which implies  $\sigma(B) > \sigma(C)$ .

**Proof.** See Appendix A.8.  $\square$

**Example 8.** For Figure 1a as in Example 1, by the calculation results in Example 2, the dendrogram that corresponds to  $\bigcup_{\alpha \in \mathbb{R}} \mathcal{C}_\alpha(V)$  is shown in Figure 1d.

We defined the community strength in (6) by modeling the problem based on the convex game in game theory, gave its alternative forms in (9) and (14), and showed that the community strength and the solutions to the minimization of (14) are related to the first turning point of the curve defined by (18) against the parameter  $\alpha$ . We also showed that the collection of strong communities defined in (10) form a hierarchy and can be represented by a dendrogram. These motivate the methods for computing strong communities, as described in the following section.

### 5. Computation of Strong Communities

In this section, we will show how to calculate the strong communities in  $\mathcal{C}_\alpha(V)$  at a threshold  $\alpha$ , and how to calculate all the strong communities.

The following result shows that  $\mathcal{C}_\alpha(V)$  can be calculated with a recursive procedure.

**Theorem 3.** For  $\alpha \in \mathbb{R}$ ,  $\mathcal{C}_\alpha(V)$  can be calculated with the following recurrence relation

$$\mathcal{C}_\alpha(V) = \begin{cases} \emptyset, & \text{if } |V| \leq 1, \\ (\mathcal{B}_\alpha^*(V) \setminus \{\{i\} | i \in V\}) \cup \mathcal{C}_\alpha(U), & \text{otherwise,} \end{cases} \quad (31)$$

where

$$U := V \setminus \bigcup \mathcal{B}_\alpha^*(V), \quad (32)$$

$\mathcal{B}_\alpha^*(V)$  is defined in (20), and (31) is the base case.

**Proof.** See Appendix A.9.  $\square$

Theorem 2 shows that  $\mathcal{C}_\alpha(V)$  can be computed in a divisive way. In the first recursive step,  $V$  is the ground set, if  $|V| \leq 1$ , we directly calculate  $\mathcal{C}_\alpha(V) = \emptyset$  by the base case (31) and stop the recursion; otherwise we calculate  $\mathcal{B}_\alpha^*(V)$ , then  $\mathcal{B}_\alpha^*(V) \setminus \{\{i\} | i \in V\}$  is the set of newly found strong subsets, and we enter the next recursive step. The new recursive step is similar to the first recursive step, but we use  $U$  given in (33) as the ground set.

The following example shows how to run the recursive procedure in Theorem 3 for computing  $\mathcal{C}_\alpha(V)$ .

**Example 9.** Consider Figure 1a as in Example 1 and we calculate  $\mathcal{C}_\alpha(V)$  at  $\alpha = \frac{1}{2}$  by following Theorem 3.

(1) The first recursive step:

- $|V| > 1$ , which corresponds to the case in (32).
- Then we need to compute  $\mathcal{B}_\alpha^*(V)$ . By the calculation in Example 4, we know  $\mathcal{B}_\alpha^*(V) = \{\{1, 2\}\}$ .
- By (32), the elements in  $\mathcal{B}_\alpha^*(V) \setminus \{\{i\} | i \in V\} = \{\{1, 2\}\}$  are in  $\mathcal{C}_\alpha(V)$ , and computing  $\mathcal{C}_\alpha(U)$  will provide us the remaining strong subsets in  $\mathcal{C}_\alpha(V)$ , where  $U$  is given by (33). Here,  $U = \{3\}$ .

(2) The second recursive step:

- Regard  $U$  as the ground set and compute  $\mathcal{C}_\alpha(U)$  according to (31) and (32).
- Since  $|U| = 1$ , the base case (31) applies, which means  $\mathcal{C}_\alpha(U) = \emptyset$ , and the recursive procedure ends.

According to the recursive steps,  $\mathcal{C}_\alpha(V) = \{\{1, 2\}\}$ .

Notice that in this example, there are two recursive steps in total. For some other examples where the  $U$  obtained in the first recursive step has a cardinality larger than 1, i.e., the case (32) applies, then the following recursive step will be similar to the first recursive step except that  $U$  instead of  $V$  is regarded as the ground set.

Additionally, we use the  $\mathcal{B}_\alpha^*(V)$  from the calculation in Example 4, which employs a brute force method enumerating all  $B \subseteq V : B \neq \emptyset$ . We will discuss how to compute  $\mathcal{B}_\alpha^*(V)$  in polynomial time later.

The following example illustrates why there are strong communities not in  $\mathcal{B}_\alpha^*(V)$  and why the recursive procedure in Theorem 3 can identify those strong communities.

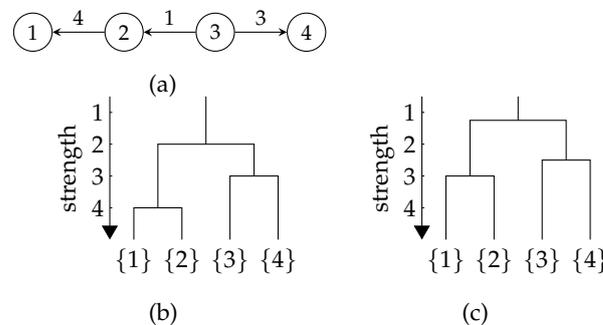
**Example 10.** Consider Figure 3a on  $V = \{1, 2, 3, 4\}$  with function  $g$  defined by (5) with  $\gamma = 1$  and  $\beta = \frac{1}{2}$ . Then  $g(C)$  is the total weight of internal edges in  $C$ .

In the graph,  $C_1 := \{1, 2\}$  and  $C_2 := \{3, 4\}$  have relatively large total weights of internal edges compared with other subsets of  $V$  hence they are meaningful communities that are expected to be identified.

Let  $\alpha = 2$ .  $\mathcal{B}_\alpha^*(V)$  in (20) contains the minimal non-empty subsets of  $V$  that leads to the minimum value of  $f_\alpha(\cdot)$  in (19).

$C_1$  is identified by  $\mathcal{B}_\alpha^*(V)$  because  $f_\alpha(C_1)$  achieves the minimum value among all the non-empty subsets of  $V$ . However, the other meaningful community  $C_2$  is not in  $\mathcal{B}_\alpha^*(V)$  because  $C_2$  will never be a subsets of  $V$  that leads to the minimum value of  $f_\alpha(\cdot)$ , as  $f_\alpha(C_1) < f_\alpha(C_2)$  always holds. In other words,  $C_1$  dominates  $C_2$ .

To identify  $C_2$ , we remove the nodes that appeared in the communities in (32) from  $V$ , as described in (33), and then start a new recursive step to identify strong communities within the remaining nodes. Since the community  $C_1$  that dominated  $C_2$  in  $V$  was removed,  $C_2$  can now be identified with  $\mathcal{B}_\alpha^*(\cdot)$ . In this way, the recursive procedure in Theorem 3 works to identify all the strong communities in  $\mathcal{C}_\alpha(V)$ .



**Figure 3.** A simple digraph and the dendrogram when  $g$  is defined by (46) with different  $\beta$ . (a) The digraph; (b) The dendrogram when  $\beta = 1$ ; (c) The dendrogram when  $\beta = \frac{1}{2}$ .

For the recurrence relation in Theorem 3 to be applicable, the recursive procedure in Theorem 3 finishes with finite recursive steps. The following results imply that  $U$  in (31) always has a smaller size than  $V$ .

**Proposition 3.** For  $\alpha \in \mathbb{R}$  and the set  $V$  with  $|V| \geq 2$ ,

$$|\mathcal{B}_\alpha^*(V)| \geq 1. \tag{34}$$

**Proof.** See Appendix A.10.  $\square$

As a result of Proposition 3, the number of recursive steps in Theorem 3 needed is bounded by  $|V|$ .

**Proposition 4.** For a non-empty set  $V$ , it takes at most  $|V|$  recursive steps to calculate  $\mathcal{C}_\alpha(V)$  by Theorem 3.

**Proof.** See Appendix A.11.  $\square$

The following property is the basis of Theorem 3, which ensures that the recursive procedure in Theorem 3 does not leave out any strong communities in  $\mathcal{C}_\alpha(V)$  for a chosen  $\alpha$ .

**Proposition 5.** For any  $B \in \mathcal{B}_\alpha^*(V) : |C| > 1, \forall C \subseteq V,$

$$B \cap C \neq \emptyset \text{ and } C \not\subseteq B \Rightarrow \sigma(C) \leq \alpha, \tag{35}$$

or the contrapositive

$$\sigma(C) > \alpha \Rightarrow B \cap C = \emptyset \text{ or } C \subseteq B. \tag{36}$$

**Proof.** See Appendix A.12.  $\square$

Equation (36) implies that any other strong subset with strength larger than  $\alpha$  is either disjoint with the elements in  $\mathcal{B}_\alpha^*(V)$ , or is a subset of an element in  $\mathcal{B}_\alpha^*(V)$ . This ensures that when we continue the computation with  $U$  in (33) as the ground set after we obtained the strong subset with strength larger than  $\alpha$  in  $\mathcal{B}_\alpha^*(V) \setminus \{\{i\} | i \in V\}$ , the remaining strong subsets will be captured by  $\mathcal{C}_\alpha(U)$ .

To obtain  $\mathcal{C}_\alpha(V)$ , calculating  $\mathcal{B}_\alpha^*(V)$  is a basic step that requires optimization of (18), which can be done based on the method in [26] as described in the following.

### 5.1. Divide-and-Conquer

We rewrite the minimization of  $\hat{f}_\alpha(V)$  in (18) in a similar way as that in [26]:

$$\hat{f}_\alpha(V) = \min_{t \in V} \hat{f}_\alpha^{(t)}(V), \text{ where} \tag{37}$$

$$\hat{f}_\alpha^{(t)}(V) := \min_{\substack{B \subseteq V: \\ t \in B}} f_\alpha(B), \tag{38}$$

which is a two-step optimization problem, and denote  $\mathcal{B}_\alpha^{(t)}(V)$  as the minimal optimal solution set to (38).

Since  $\hat{f}_\alpha$  defined in (18) is a submodular function,  $\mathcal{B}_\alpha^{(t)}(V)$  can be solved with submodular function minimization (SFM) algorithms, and it has a unique element since the feasible domain  $\{B \subseteq V | j \in B\}$  is a lattice ([27] Proposition 10.1).

Let  $T_\alpha^*(V)$  be the set of optimal solutions  $t$  to (37), we have the following result that indicates how  $\mathcal{B}_\alpha^*(V)$  can be calculated.

**Proposition 6** ([26] Proposition 2). For  $\alpha \in \mathbb{R}, \mathcal{B}_\alpha^*(V)$  in (20) can be obtained from  $\mathcal{B}_\alpha^{(t)}(V), t \in V$  and  $T_\alpha^*(V)$  by

$$\mathcal{B}_\alpha^*(V) = \text{minimal} \bigcup_{t \in T_\alpha^*(V)} \mathcal{B}_\alpha^{(t)}(V). \tag{39}$$

**Proof.** See Appendix A.13.  $\square$

According to Theorem 3, computing  $\mathcal{C}_\alpha(V)$ , all the strong subset with a strength larger than  $\alpha$ , can be done with the following steps:

- (1) Calculate  $\mathcal{B}_\alpha^{(t)}(V)$  for  $t \in V$  by optimizing (38) with SFM algorithms;
- (2) Calculate  $T_\alpha^*(V)$  by optimizing (38);
- (3) Calculate  $\mathcal{B}_\alpha^*(V)$  according to (39);
- (4) Calculate  $U$  by (33);
- (5)  $\mathcal{M}$  is the newly found strong subsets that have a strength larger than  $\alpha$  in this recursive step. If  $|U| \leq 1$ , then stop; otherwise, regard  $U$  as  $V$  and go to (1) to start a new recursive step.

The union of the set of strong subsets calculated in all the recursive steps is  $\mathcal{C}_\alpha(V)$ . To calculate all the strong subsets, i.e.,  $\bigcup_{\alpha \in \mathbb{R}} \mathcal{C}_\alpha(V)$ , for each  $t \in V$ , define

$$g^{(t)}(A) := g(A \cup \{t\}) - g(\{t\}) \text{ for } A \subseteq (V \setminus \{t\}), \tag{40}$$

then

$$B' \subseteq (V \setminus \{t\}) \mapsto \alpha|B'| - g^{(t)}(B') \tag{41}$$

is a normalized submodular function.

We need to obtain the minimal optimal solution to (41) for all  $\alpha \in \mathbb{R}$ . Luckily, with SFM algorithms such as Wolfe’s minimum norm point algorithm [28], for (41), we can obtain for some  $N'_t \in \mathbb{N}$  the sequence of  $\alpha_1^{(t)}, \alpha_2^{(t)}, \dots, \alpha_{N'_t}^{(t)}$  and the corresponding sequence of sets  $A_1^{(t)}, A_2^{(t)}, \dots, A_{N'_t}^{(t)}$  that satisfies ([27] Proposition 8.6)

$$-\infty = \alpha_0^{(t)} < \alpha_1 < \alpha_2 < \dots < \alpha_{N'_t}^{(t)} < \alpha_{N'_t+1}^{(t)} \infty, \tag{42}$$

$$A_0^{(t)} := V \supseteq A_1^{(t)} \supseteq A_2^{(t)} \supseteq \dots \supseteq A_{N'_t}^{(t)} := \emptyset, \tag{43}$$

and for any  $\alpha \in [\alpha_i^{(t)}, \alpha_{i+1}^{(t)}], i \in \{0, 1, \dots, N'_t\}$ ,

$$A_i^{(t)} \text{ is the minimal minimizer to (41)}. \tag{44}$$

Equation (44) means with the sequences (42) and (43), we can obtain the minimum solution to (41) for all  $\alpha \in \mathbb{R}$ .

For any  $\alpha \in \mathbb{R}$ , if  $A^{(t)}$  is the unique minimal solution to (41), then  $B^{(t)} = A^{(t)} \cup \{t\}$  will be the unique minimal solution to (37), or in another word,

$$\mathcal{B}_\alpha^{(t)}(V) = \{B^{(t)}\}, \tag{45}$$

since  $g(B') - g^{(t)}(B') = g(\{t\})$  is a constant for all  $C \subseteq V \setminus \{t\}$ . This means the minimal solution set  $\mathcal{B}_\alpha^{(t)}(V)$  to (37) for all  $\alpha \in \mathbb{R}$  can be obtained from the solutions to (41) with sequences (42) and (43).

Therefore, with sequences (42) and (44) for all  $t \in V$ ,  $T_\alpha^*(V)$  can be obtained for all  $\alpha \in \mathbb{R}$ . Then calculating  $\mathcal{C}_\alpha(V)$  for all  $\alpha \in \bigcup_{t \in V} \{a_0^{(t)}, \alpha_1, \dots, \alpha_{N'_t+1}^{(t)}\}$  based on Theorem 3, Proposition 6 and  $T_\alpha^*(V)$  is sufficient for us to obtain  $\bigcup_{\alpha \in \mathbb{R}} \mathcal{C}_\alpha(V)$ .

With  $\text{MNB}(n)$  denoting the complexity of the minimum norm base algorithm for SFM on the ground set of size  $n$ , we have the following result.

**Proposition 7.**  $\bigcup_{\alpha \in \mathbb{R}} \mathcal{C}_\alpha(V)$  can be computed in  $\mathcal{O}(|V|^3 \text{MNB}(|V|))$  time.

**Proof.** See Appendix A.14.  $\square$

### 5.2. Using Max-Flow Min-Cut Algorithm

For the step of optimizing (38) in computing  $\mathcal{C}_\alpha(V)$ , SFM algorithms are used. However, SFM algorithms are generally computationally expensive. There are works on improving the efficiency of SFM problems by max-flow min-cut algorithms [29,30]. We discuss a category of choices for  $g$  when the max-flow min-cut algorithm can be utilized for computing  $\mathcal{C}_\alpha(V)$ .

Consider the function  $g$  defined in the form of ([26] Definition 2)

$$g(B) = \beta \cdot w(B, B) - (1 - \beta) \cdot w(V \setminus B, B), \tag{46}$$

which is a special case of (5) with  $\gamma = 1$ .

Following the method in [26], we can construct an augmented digraph and run a max-flow min-cut algorithm [31–34] to obtain the solution to (38). With  $\alpha$  in (19) and  $\beta$  in (46), the  $(\alpha, \beta, t)$ -augmented digraph [26] is a digraph on  $\{V\} \cup \{s\}$  where  $s \notin V$  is an additional node, with the edge weight  $w_{\alpha, \beta, t} : (V \cup \{s\}) \rightarrow \mathbb{R}_{\geq 0}$  defined as

$$w_{\alpha, \beta, t}(i, j) := \begin{cases} w(i, j), & i \in V, j \in V \setminus \{t\} \\ w(i, j) + \beta d_i, & i \in V, j = t \\ \alpha, & i = s, j \in V \\ 0, & \text{otherwise.} \end{cases} \tag{47}$$

**Proposition 8** ([26] Theorem 3). *The  $\mathcal{B}_\alpha^{(t)}(V)$  contains a unique minimum set  $C$  such that  $(\{s\} \cup V \setminus C, C)$  is a minimum  $s$ - $t$  cut of the  $(\alpha, \beta, t)$ -augmented digraph.*

Proposition 8 implies that  $\mathcal{B}_\alpha^{(t)}(V)$  can be solved by max-flow min-cut algorithm. Moreover, with the parametric max-flow min-cut algorithms [34], we can obtain  $\mathcal{B}_\alpha^{(t)}(V)$  for all  $\alpha \geq 0$ . Hence, when  $g$  has the form of (46), computing  $\mathcal{C}_\alpha(V)$  for a certain  $\alpha$  or all  $\alpha$  follows the same procedure in Section 5.1, except that we can use max-flow min-cut algorithms to calculate  $\mathcal{B}_\alpha^{(t)}(V)$  instead of SFM algorithms.

### 6. Discussions

To illustrate the dendrogram of strong communities found by our approach, the digraph in Figure 3a is used as an example, with the function  $g$  given in (46) for different choices of  $\beta$  for experiments.

The result for the cases  $\beta = 1$  and  $\beta = 0.5$  are shown in Figure 3b and Figure 3c, respectively. The example of the calculation procedures based on Theorem 3 for the case  $\beta = 1$  is in Appendix A.15.

We can obtain the collection of strong communities  $\mathcal{C}_\alpha(V)$  in (10) for  $\alpha$  from the dendrogram. For instance, the strong communities for  $\alpha = \frac{5}{2}$  is,

$$\mathcal{C}_{\frac{5}{2}}(V) = \{\{1, 2\}, \{3, 4\}\}.$$

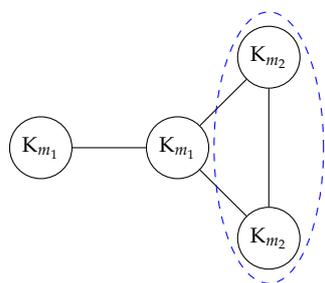
The parameter  $\beta$  in (46) is a balancing factor between the total weight of internal edges and the negative total weight of incoming edges, and when  $\beta = 1$ , it can be used for the problem of finding the minimal densest subgraphs.

In [35], another kind of augmented graph is constructed, and an algorithm is given for quickly increasing  $\alpha$  value based on the current community that has already been found and then conducting max-flow min-cut algorithms. We want to point out that, although the method there is similar to solving  $\mathcal{B}_\alpha(V)$  in our work, the algorithm there for calculating the next critical  $\alpha$ , as the author also said, needs more calculation steps if we want to obtain more solutions for intermediate  $\alpha$ 's. In other words, not all critical  $\alpha$ 's are found, while our approach calculates all the critical  $\alpha$ 's and the solutions directly. Additionally, our approach goes beyond just finding  $\mathcal{B}_\alpha(V)$ , and we considered digraphs, which can be generalized to undirected graphs directly.

The work in [26] extends the notion of web communities [36] to digraphs and calculates the communities in polynomial time, which is closely related to our approach. In fact,  $\mathcal{B}_\alpha^*(V) \setminus \{\{i\} | i \in V\}$  is the set of web communities, which is included in the strong communities defined in (10) in this work. For a set  $C \subsetneq V : C \neq \emptyset$ , subsets of  $V \setminus C$  can prevent  $C$  from being a web community in [26], even if  $C$  has a strength larger than  $\alpha$ . Nevertheless, such a phenomenon does not exist for strong communities detected by our approach. Whether  $C$  is a strong community in (10) or not is independent of subsets of  $V \setminus C$  according to (10). In Example 10,  $C_1$  is a web community, and  $C_2$  is not, since  $C_2$  is dominated by  $C_1$ . However,  $C_2$  is a meaningful community that is expected to be identified.

The web community method fails to identify  $C_2$ , while our approach can identify it as a strong community, as we have calculated in the example.

Our approach also addresses some known issues associated with existing community detection methods. For instance, Modularity [37], a common community detection method, is NP-hard and suffers from the limitation of resolution limit [38]. There are works such as [39,40] aiming at resolving the resolution limit issue, yet both rely on heuristics to obtain solutions. It is worth mentioning that Modularity is a measure applied over partitions, while our strength measure is on individual communities. In Figure 4, there are four complete graphs of two sizes,  $m_1 = 20$  and  $m_2 = 5$ . Despite the two complete graphs of size  $m_2$  being smaller than the other two, they should be identified as communities since they are the maximal complete graphs. However, Modularity fails to recognize the two smaller complete graphs, and instead, it merges them into a single community [38]. In contrast, our approach successfully identifies the two smaller complete graphs of size  $m_2$  as strong communities.



**Figure 4.** An unweighted graph with  $m_1 = 20$ ,  $m_2 = 5$  where  $K_m$  denotes the complete graph with  $m$  nodes [38]. As for the two smaller complete graphs denoted by  $K_{m_2}$ , Modularity [37] will merge the two into a single community as indicated by the blue dashed ellipse due to resolution limit [38], while our approach can identify each of them as strong communities.

The strong communities are derived from game theory, where the strength can be interpreted as the support inside the community that can be shared with a part of individuals in need in the same community. Application in real-world problems is promising, such as finding small groups of advertisers and keywords in sponsored auctions, where the community strength means the average money inside the groups [35,41].

## 7. Conclusions

We introduced a novel concept of strength for community detection using a convex game model in cooperative game theory. It can be applied to networks with a supermodular characteristic function. Theorem 1 establishes the dual objective function, based on which we conducted a comprehensive analysis of strong community properties. The laminar structure demonstrated in Theorem 2 reveals that strong communities form a hierarchy and can be represented by a dendrogram.

To compute strong subsets, Theorem 3 introduces a recurrence relation, enabling polynomial time calculations through submodular function minimization. For specific characteristic function choices in the convex game on the graphical model, an augmented digraph can be constructed to apply the max-flow min-cut algorithm to improve computation efficiency.

Unlike many existing community detection methods, which often rely on approximation, are non-deterministic, and lack guarantees on complexity, our approach for community detection is deterministic, computable in polynomial time, and supported by a rigorous theoretical analysis of its properties. Since our approach captures high-order statistics through the supermodular characteristic function, the primary limitation of our approach lies in its computational complexity. This complexity presents a challenge when applying the method to large-scale real-world datasets. Nevertheless, our work proposes an analytical framework for community detection that yields unique solutions and provides theoretical foundations for future research aimed at improving the complexity and empirical applications.

## Appendix A

### Appendix A.1

We show the computation and visualization of  $\text{Core}(V, g)$  in Example 1. According to (4),

$$\text{Core}(V, g) = \{r_V := (r_1, r_2, r_3) \mid r_1 + r_2 + r_3 = g(V) = 1, \tag{A1}$$

$$r_1 + r_2 \geq g(\{1, 2\}) = 1, \tag{A2}$$

$$r_1 + r_3 \geq g(\{1, 3\}) = 0, \tag{A3}$$

$$r_2 + r_3 \geq g(\{2, 3\}) = 0, \tag{A4}$$

$$r_1, r_2, r_3 \geq 0 \tag{A5}$$

$$= \{(r_1, r_2, r_3) \mid r_1 + r_2 \geq 1, r_1 \geq 0, r_2 \geq 0, r_3 = 0\}, \tag{A6}$$

where  $r_1, r_2, r_3$  are the payoff allocated to players 1, 2 and 3, respectively.

Equation (A1) ensures that the payoff vector  $r_V$  exactly splits the total worth of  $V$ , (A2) ensures that the total payoff for  $\{1, 2\}$  is no less than the worth of  $\{1, 2\}$  so that players in  $\{1, 2\}$  will not deviate from  $V$ . The interpretation for (A3)–(A5) is similar to (A2). The intersection defined by (A1) through (A5) is the  $\text{Core}(V, g)$  which is visualized in Figure 1b, where

- (A1) is the region given by the plane containing  $(1, 0, 0)$ ,  $(0, 1, 0)$  and  $(0, 0, 1)$ , as indicated by green color;
- (A2) is the region that includes the plain and on the opposite side of the origin relative to the plane, which contains  $(1, 0, 0)$  and  $(0, 1, 0)$  and parallel to  $r_3$  axis, as indicated by red color;
- (A5) restricts each element of  $r_V$  to be non-negative, which includes the region represented by (A3) and (A4); and
- the line segment in blue between  $(1, 0, 0)$  and  $(0, 1, 0)$  is  $\text{Core}(V, g)$ .

Equation (A6) is the simplified algebraic form for  $\text{Core}(V, g)$ .

### Appendix A.2

**Proof of Proposition 1.** Given  $C \subseteq V : |C| > 1$ , for any  $B \subsetneq C : B \neq \emptyset$  and payoff vector  $r_C$ ,

$$\min_{i \in B} r_i \leq \frac{r(B)}{|B|}. \tag{A7}$$

Then, we show that the equality in (A7) holds. Suppose node  $j \in B$  is a solution to the left-hand side of (A7), i.e.,

$$\min_{i \in B} r_i = r_j. \tag{A8}$$

Let  $B = \{j\}$ , then the right-hand side of (A7) is

$$\frac{r(B)}{|B|} = r_j. \tag{A9}$$

Equations (A8) and (A9) imply that the equality in (A7) holds.

Since the equality in (A7) holds for  $B \subsetneq C : B \neq \emptyset$  and payoff vector  $r_C$ , then

$$\min_{\substack{B \subsetneq C \\ B \neq \emptyset}} \max_{r_C \in \text{Core}(C, g)} \min_{i \in B} r_i = \min_{\substack{B \subsetneq C \\ B \neq \emptyset}} \max_{r_C \in \text{Core}(C, g)} \frac{r(B)}{|B|}, \tag{A10}$$

which establishes (9) according to (6). □

Appendix A.3

**Proof of Theorem 1.** For  $\forall B \subsetneq C, B \neq \emptyset$ , by the relationship between  $r(\cdot)$  and  $g(\cdot)$  when  $r_C \in \text{Core}(C, g)$ , we have

$$r(B) = r(C) - r(C \setminus B) \leq g(C) - g(C \setminus B),$$

where the equality is achieved when  $r(C \setminus B) = g(C \setminus B)$ . Hence,

$$\max_{r_C \in \text{Core}(C, g)} \frac{r(B)}{|B|} \leq \frac{g(C) - g(C \setminus B)}{|B|} \tag{A11}$$

Next, we will show that  $\exists r_C \in \text{Core}(C, g)$  s.t.  $r(C \setminus B) = g(C \setminus B)$  so that the equality is achieved in (A11).

To ease the notation, denote

$$T := C \setminus B. \tag{A12}$$

With  $T$ , (A11) becomes

$$\max_{r_C \in \text{Core}(C, g)} \frac{r(B)}{|B|} \leq \frac{g(C) - g(T)}{|C \setminus T|} \tag{A13}$$

For the convex subgame  $(T, g)$ , the core of  $(T, g)$  is non-empty since the game is convex [17,25]. Suppose  $r_T \in \text{Core}(T, g)$ , which means

$$r(T) = g(T), \text{ and} \tag{A14}$$

$$r(T') \geq g(T'), \forall T' \subseteq T. \tag{A15}$$

Then we show the steps to construct a vector  $r_C$  from  $r_T$  with  $r_C \in \text{Core}(C, g)$ .

Step 1: Select an  $i$ , with

$$i \in \arg \max_{j: j \in C \setminus T} g(T \cup \{j\}) - g(\{j\}). \tag{A16}$$

Step 2: Assign

$$r_i = g(T \cup \{i\}) - g(\{i\}), \tag{A17}$$

and construct  $r_{T \cup \{i\}}$  by values from  $r_T$  and  $r_i$ .

For  $\forall T' \subseteq T$ , by supermodularity of  $g$ ,

$$g(T' \cup \{i\}) - g(T') \leq g(T \cup \{i\}) - g(T). \tag{A18}$$

Then,

$$r(T' \cup \{i\}) = r(\{i\}) + r(T') \tag{A19}$$

$$= g(T \cup \{i\}) - g(\{i\}) + r(T') \tag{A20}$$

$$\geq g(T \cup \{i\}) - g(\{i\}) + g(T') \tag{A21}$$

$$\geq g(T' \cup \{i\}), \tag{A22}$$

where (A19) is by (1), (A20) is by (A17), (A21) is by (A15), and (A22) is by (A18).

Equations (A22), (A14) and (A15) indicate that

$$r_{T \cup \{i\}} \in \text{Core}(T \cup \{i\}, g). \tag{A23}$$

Update  $T$  by  $T \cup \{i\}$ , and continue the Step 1 and Step 2 above, and finally we will have a constructed

$$r_C \in \text{Core}(C, g), \tag{A24}$$

with  $r(T) = g(T)$  preserved.

This means, for  $\forall B \subsetneq C, B \neq \emptyset$ , the formula (A11), or its equivalent formula (A13), can achieve equality with the constructed  $r_C$  obtained in (A24). Then (14) is implied considering (9).

Therefore, the minimization over the non-empty proper subset of  $C$  for the right-hand side of (A13) will lead to a value equal to the minimization over the non-empty proper subset of the left-hand side of (A13), which is the community strength for  $C$  defined in (9). And based on the above proof, we know that if a non-empty set  $B^*$  is a solution for the minimization in (9), the set  $T^* := C \setminus B^*$  will be a solution for the minimization in (14).

Hence, Theorem 1 is established.  $\square$

Appendix A.4

**Proof of Proposition 2.** We first show the second property in Proposition 2, which will imply the first property.

Denote the sign of a number  $x$  by

$$\text{sgn}(x) := \begin{cases} 1, & x > 0 \\ -1, & x < 0 \\ 0, & x = 0. \end{cases} \tag{A25}$$

We have

$$\begin{aligned} & \text{sgn}(\sigma(C) - \alpha) \\ &= \text{sgn}\left(\min_{\substack{B \subsetneq C: \\ B \neq \emptyset}} \frac{g(C) - g(B)}{|C \setminus B|} - \alpha\right) \end{aligned} \tag{A26a}$$

$$= \text{sgn}\left(\min_{\substack{B \subsetneq C: \\ B \neq \emptyset}} g(C) - g(B) - \alpha|C \setminus B|\right) \tag{A26b}$$

$$= \text{sgn}\left(\min_{\substack{B \subsetneq C: \\ B \neq \emptyset}} f_\alpha(B) - f_\alpha(C)\right), \tag{A26c}$$

where

- (A26a) is by (14);
- (A26b) is because  $|C \setminus B| > 0$  and  $B \neq C$  and  $\text{sgn}(x) = \text{sgn}(ax)$  if  $a > 0$ ;
- (A26c) is by the definition of  $f_\alpha$  in (19).

Note also that the sets of optimal solutions to the minimization/maximization in each step are the same. Hence,

$$\sigma(C) < \alpha \iff \text{sgn}(\sigma(C) - \alpha) < 0 \tag{A27}$$

$$\iff \text{sgn}\left(\min_{\substack{B \subsetneq C: \\ B \neq \emptyset}} f_\alpha(B) - f_\alpha(C)\right) < 0 \tag{A28}$$

$$\iff \min_{\substack{B \subsetneq C: \\ B \neq \emptyset}} f_\alpha(B) < f_\alpha(C) \tag{A29}$$

$$\iff C \notin \mathcal{B}_\alpha(C), \tag{A30}$$

where

- (A27) is by (A25);
- (A28) is by (A26c);
- (A29) is by (A25);
- (A30) is by definition of  $\mathcal{B}_\alpha(C)$ .

Equation (A30) implies (25).

Similarly, with the inequalities “<” replaced by “>”, and simply by definition of  $\mathcal{B}_\alpha(C)$ , we have

$$\begin{aligned} \sigma(C) > \alpha &\iff \min_{\substack{B \subseteq C; \\ B \neq \emptyset}} f_\alpha(B) > f_\alpha(C) \\ &\iff \{C\} = \mathcal{B}_\alpha(C), \end{aligned}$$

which implies (24) and (27).

With the inequalities replaced by equalities, we have

$$\sigma(C) = \alpha \iff \min_{\substack{B \subseteq C; \\ B \neq \emptyset}} f_\alpha(B) = f_\alpha(C),$$

which implies (23) and (26). This completes the proof.  $\square$

Appendix A.5

**Proof of Lemma 1.** Assume

$$\sigma(C_1) \geq \sigma(C_2) = \alpha \tag{A31}$$

It suffices to show that  $\sigma(C_1 \cup C_2) \geq \alpha$ , or equivalently, by (23) and (24),

$$f_\alpha(B) \geq f_\alpha(C_1 \cup C_2), \forall B \subseteq C_1 \cup C_2 : B \neq \emptyset. \tag{A32}$$

Consider the case  $B \subseteq C_1$  first,

$$\begin{aligned} f_\alpha(B) &\stackrel{(a)}{\geq} f_\alpha(C_1) \\ &\stackrel{(b)}{\geq} f_\alpha(C_1 \cup C_2) - \underbrace{f_\alpha(C_1) + f_\alpha(C_1 \cap C_2)}_{\stackrel{(c)}{\geq} 0} \end{aligned}$$

which implies (A32) as desired, where

- (a) and (c) are by (23) and (24), since both  $\sigma(C_1)$  and  $\sigma(C_2)$  are at least  $\alpha$  by (A31);
- (b) is because  $f_\alpha$  is submodular due to the supermodularity of  $g$  and modularity of the cardinality function.

Then consider the remaining case  $B \not\subseteq C_1$ , i.e.,  $B \subseteq C_1 \cup C_2$  and  $B \cap C_2 \neq \emptyset$ ,

$$\begin{aligned} f_\alpha(B) &\stackrel{(d)}{\geq} f_\alpha(B \cup C_2) - \underbrace{f_\alpha(C_2) + f_\alpha(B \cap C_2)}_{\stackrel{(e)}{\geq} 0} \\ &\stackrel{(f)}{\geq} f_\alpha(\underbrace{(B \cup C_2) \cup C_1}_{\stackrel{(g)}{=} C_1 \cup C_2}) - \underbrace{f_\alpha(C_1) + f_\alpha((B \cup C_2) \cap C_1)}_{\stackrel{(h)}{\geq} 0}, \end{aligned}$$

which implies (A32) as desired, where

- (d) and (f) are by the submodularity of  $f_\alpha$  explained above;
- (h) is by (23) and (24) since  $(B \cup C_2) \cap C_1 \subseteq C_1$ ,  $(B \cup C_2) \cap C_1 \supseteq C_1 \cap C_2$  and  $C_1 \cap C_2 \neq \emptyset$  by the assumption stated in the lemma;
- (g) is because  $B \subseteq C_1 \cup C_2$ .

$\square$

Appendix A.6

**Proof of Theorem 2.** Suppose on the contrary that  $C_1 \not\supseteq C_2$  and  $C_1 \cap C_2 \neq \emptyset$ . By Lemma 1,

$$\sigma(C_1 \cup C_2) \geq \sigma(C_1) > \alpha_1.$$

This contradicts  $C_1 \in \mathcal{C}_{\alpha_1}(V)$ , in particular, the inclusion-wise maximality of  $C_1$ , since  $C_1 \cup C_2 \supsetneq C_1$ , where the strict inclusion is because  $C_1 \not\supseteq C_2$ .  $\square$

Appendix A.7

**Proof of Corollary 1.** Consider  $B \subsetneq V : B \neq \emptyset$ . An inclusion-wise minimal element in the set in its parent  $p(B)$  exists since  $V$  is in the set. Suppose on the contrary that there can be multiple minimal elements, say  $C_1 \in \mathcal{C}_{\alpha}(V)$  and  $C_2 \in \mathcal{C}_{\alpha}(V)$  with  $\alpha_1 \leq \alpha_2$  w.l.o.g.

By Theorem 2, since  $C_1 \cap C_2 \supseteq B \neq \emptyset$ , we have  $C_2 \subseteq C_1$ , contradicting the minimality of  $C_1$  as desired. Hence,  $p(B)$  exists and is unique.  $\square$

Appendix A.8

**Proof of Corollary 2.** To prove the “if” case, consider any  $B \in \mathcal{C}_{\sigma(C)}(C)$ , on one hand,  $B \subsetneq C$ ; on the other hand, by definition of  $\mathcal{C}_{\alpha}(\cdot)$  in (14), we have  $\sigma(B) > \sigma(C)$ . Hence,  $C$  satisfies the condition in (29). To show minimality, note that the maximality of  $B \in \mathcal{C}_{\sigma(C)}(C)$  according to the definition of  $\mathcal{C}_{\alpha}(\cdot)$  in (14) indicates that

$$\forall C' \subseteq C : B \subsetneq C', \sigma(C') \leq \sigma(C).$$

Hence,  $p(B) = C$ .

Consider the reverse case, i.e.,  $p(B) = C$  and any  $\alpha' \in \mathbb{R}$  s.t.  $B \in \mathcal{C}_{\alpha'}(V)$ . Then

$$\sigma(B) > \alpha' \geq \sigma(C),$$

where the last inequality is by Theorem 2 because  $C \notin \mathcal{C}_{\alpha'}(V)$  as  $C \not\supseteq B$ . To show  $B \in \mathcal{C}_{\sigma(C)}(C)$ , it suffices to show maximality of  $B$  in  $\mathcal{C}_{\sigma(C)}(C)$ , i.e.,

$$\forall C' \subsetneq C : C' \not\supseteq B, \sigma(C') \leq \sigma(C).$$

Suppose, on the contrary, that there exists

$$C' \subsetneq C : C' \not\supseteq B, \sigma(C') > \sigma(C).$$

Then, there must also exist

$$C'' \in \mathcal{C}_{\sigma(C)}(C) : C'' \supseteq C'.$$

By Theorem 2, since  $C'' \cap C \supseteq B \neq \emptyset$ , we must have  $C'' \subsetneq C$ , contradicting the minimality of  $C$  by definition of parent in (29) as desired.

Hence, Corollary 2 is established.  $\square$

Appendix A.9

**Proof of Theorem 3.** When  $|U| \leq 1$ , we have  $\mathcal{C}_{\alpha}(U) = \emptyset$  by the definition of  $\mathcal{C}_{\alpha}(\cdot)$  in (10), then (31) is implied by (32).

Then we prove the equality in (32) by showing that for  $|V| \geq 2$ ,

$$(\mathcal{B}_{\alpha}^*(V) \setminus \{\{i\} | i \in V\}) \subseteq \mathcal{C}_{\alpha}(V). \tag{A33}$$

and

$$\mathcal{C}_{\alpha}(V) \setminus \mathcal{B}_{\alpha}^*(V) = \mathcal{C}_{\alpha}(U). \tag{A34}$$

To show (A33), consider any  $B \in \mathcal{B}_\alpha^*(V) : |B| \geq 2$ . By minimality of  $B$  in  $\mathcal{B}_\alpha(V)$ ,

$$f_\alpha(B) > \min_{\substack{T \subseteq B \\ T \neq \emptyset}} f_\alpha(T),$$

which implies

$$\sigma(B) > \alpha$$

by (23) and (24) since  $|B| > 1$ . Then by Proposition 5,  $B$  is maximal among

$$\{B' \mid B' \subseteq V, \sigma(B') > \alpha\}, \tag{A35}$$

which means  $B \in \mathcal{C}_\alpha(V)$ .

It remains to show (A34).

First, we show that any element of the left-hand side of (A34) is on the right-hand side of (A34).

Consider any element  $C$  of the left-hand side of (A34), i.e.,

$$C \in \mathcal{C}_\alpha(V), \text{ and} \tag{A36}$$

$$C \notin \mathcal{B}_\alpha^*(V). \tag{A37}$$

Since  $C \in \mathcal{B}_\alpha^*(V)$ , we have

$$\sigma(C) > \alpha. \tag{A38}$$

For any  $B \in \mathcal{B}_\alpha^*(V)$ , we deduce that

$$C \not\subseteq B. \tag{A39}$$

This is because,

- when  $|B| = 1$ , (A39) trivially, otherwise (A36) does not hold by (10).
- when  $|B| > 1$ , by (A33),

$$B \in \mathcal{C}_\alpha(V). \tag{A40}$$

Then by (A37),

$$C \neq B,$$

which further implies (A39) by the maximality of  $C \in \mathcal{C}_\alpha(V)$  according to (10).

By (A36), (A37), (A39) and Proposition 5,

$$B \cap C = \emptyset.$$

Then

$$C \subseteq V \setminus \bigcup \mathcal{B}_\alpha^*(V),$$

which is equivalent to  $C \subseteq U$  by definition of  $U$ , and hence by (A36),

$$C \in \mathcal{C}_\alpha(U). \tag{A41}$$

Next, we show that any element of the left-hand side of (A34) is in the right-hand side of (A34).

Consider any element  $C$  of the right-hand side of (A34), i.e.,

$$C \in \mathcal{C}_\alpha(U), \tag{A42}$$

then by (10), we have

$$\sigma(C) > \alpha. \tag{A43}$$

For any

$$C' \subseteq V : C' \supsetneq C, \tag{A44}$$

we deduce that

$$\sigma(C') \leq \alpha. \tag{A45}$$

This is because

- when  $C' \subseteq U$ , (A45) holds by the maximality of  $C$  in  $\mathcal{C}_\alpha(U)$  indicated by (10).
- when  $C' \not\subseteq U$ ,

$$\exists B \in \mathcal{B}_\alpha^*(V) \text{ s.t.} \tag{A46a}$$

$$B \cap C' \neq \emptyset \tag{A46b}$$

since the way  $U$  is defined in (33) means any elements not in  $U$  must appear in subsets in  $\mathcal{B}_\alpha^*(V)$ . Additionally,

$$C' \not\subseteq B, \tag{A47}$$

otherwise (A44) implies  $C \subsetneq B$ , which further implies (A42) must not hold. According to Proposition 5, (A46a), (A46b) and (A47) implies (A45) holds.

(A45) implies  $C$  is maximal among subsets in

$$\{C'' | C'' \subseteq V, \sigma(C'') > \alpha\},$$

which further implies

$$C \in \mathcal{C}_\alpha(V). \tag{A48}$$

According to (10), (A42) implies

$$C \subseteq U. \tag{A49}$$

By (33), any element appeared in subsets in  $\mathcal{B}_\alpha^*(V)$  is not in  $U$ , then (A49) implies all the elements in  $C$  is not in any subset in  $\mathcal{B}_\alpha^*(V)$ . Hence

$$C \notin \mathcal{B}_\alpha^*(V) \tag{A50}$$

As a result, (A48) and (A50) implies

$$C \subseteq (\mathcal{C}_\alpha(V) \setminus \mathcal{B}_\alpha^*(V)).$$

Hence, (A34) is established.

The combination of (A33) and (A34) establish (32) in Theorem 3.

Hence, the equality in the recurrence relation holds.

By Proposition 4, the recursive procedure ends in finite steps. Hence, Theorem 3 is valid for calculating  $\mathcal{C}_\alpha(V)$  recursively.  $\square$

Appendix A.10

**Proof of Proposition 3.** When  $|V| \geq 2$ , the feasible domain of (18) is non-empty, then its solution set  $\mathcal{B}_\alpha(V)$  is non-empty, and hence  $\mathcal{B}_\alpha^*(V)$  is non-empty, i.e., (34) holds.  $\square$

Appendix A.11

**Proof of Proposition 4.** When  $|V| = 1$ , it corresponds to the base case in (31), hence only need 1 recursive step to calculate  $\mathcal{C}_\alpha(V)$ .

When  $|V| \geq 2$ , for  $N' \in \mathbb{N}$ , suppose  $U_1, U_2, \dots, U_{N'-1}$  is the sequence of the ground set for computing  $\mathcal{C}_\alpha(\cdot)$  used in the recursive steps that corresponds to (32), and  $U_{N'}$  is the ground set for computing  $\mathcal{C}_\alpha(\cdot)$  by the base case (31). Then we know  $U_1 = V, |U_{N'}| \in \{0, 1\}$  and  $N'$  is the number of recursive steps. Proposition 3 implies that  $|U_{i+1}| \leq |U_i| - 1$  for  $i = 1, 2, \dots, N' - 1$ . Hence  $N' \leq |V|$ .

Hence, Proposition 4 holds.  $\square$

Appendix A.12

**Proof of Proposition 5.** Suppose on the contrary that there exists  $C \subseteq V$  s.t.  $\sigma(C) > \alpha, B \cap C = \emptyset$  and  $C \not\subseteq B$ . Consider an inclusion-wise maximal  $C$ . By Lemma 1 we have

$$\begin{aligned} \sigma(C \cup B) &\geq \min\{\sigma(C), \sigma(B)\} \\ &> \alpha, \end{aligned}$$

and so, by maximality of  $C$  we have  $C \supseteq B$ . Then we have

$$\begin{aligned} f_\alpha(C) &\stackrel{(a)}{<} \min_{\substack{T \subseteq C: \\ T \neq \emptyset}} f_\alpha(T) \\ &\stackrel{(b)}{\leq} f_\alpha(B), \end{aligned}$$

where (a) is by (23) and (24) in Proposition 2, and (b) is by  $C \supseteq B$  as obtained above with the assumption. However, this contradicts the optimality of  $B \in \mathcal{B}_\alpha^*(V)$ .  $\square$

Appendix A.13

**Proof of Proposition 6.**

$$\mathcal{B}_\alpha^*(V) \stackrel{(a)}{=} \text{minimal} \bigcup_{t \in T_\alpha^*(V)} \arg \min_{B \subseteq C: t \in B} f_\alpha(B) \tag{A51}$$

$$\stackrel{(b)}{=} \text{minimal} \bigcup_{t \in T_\alpha^*(V)} \left( \text{minimal} \arg \min_{B \subseteq C: t \in B} f_\alpha(B) \right) \tag{A52}$$

$$\stackrel{(c)}{=} \text{minimal} \bigcup_{t \in T_\alpha^*(V)} \mathcal{B}_\alpha^{(t)}(V), \tag{A53}$$

where (a) is by definition of  $\mathcal{B}_\alpha^*(V)$ , (20) and (37); (b) is by the fact that for a  $t \in T_\alpha^*(V)$  and any  $A \in \arg \min_{B \subseteq C: t \in B} f_\alpha(B)$ , if  $A$  is not minimal, then  $A$  is not in the right-hand side due to the outer minimal applied to the union; and (c) is by definition of  $\mathcal{B}_\alpha^{(t)}(V)$ .  $\square$

Appendix A.14

**Proof of Proposition 7.** When  $|V| \leq 1$ , the base case (31) applies, and it takes  $\mathcal{O}(1)$  time.

When  $|V| \geq 2$ , the recursive case (32) applies. Consider the recursive step with ground set  $V$  in Theorem 3:

- For a  $t \in V$ , calculating  $\mathcal{B}_\alpha^{(t)}(V)$  for all  $\alpha \in \mathbb{R}$  takes  $\text{MNB}(|V|)$  time by the minimum norm base algorithm, hence calculating  $\mathcal{B}_\alpha^{(t)}(V)$  for all  $t \in V, \alpha \in \mathbb{R}$  takes  $\mathcal{O}(|V|^2 \text{MNB}(|V|))$  time;
- Calculating  $T_\alpha^*(V)$  for an  $\alpha \in \mathbb{R}$  takes  $\mathcal{O}(|V|)$  time, hence calculating  $T_\alpha^*(V)$  for all  $\alpha \in \mathbb{R}$  takes  $\mathcal{O}(|V|^3)$  time, since there can be a number  $\mathcal{O}(|V|)$  of  $\alpha$  values where we need to conduct calculation;

Hence it takes  $\mathcal{O}(|V|^2 \text{MNB}(|V|))$  time for this recursive step.

According to Proposition 4, the recursive procedure ends with at most  $|V|$  recursive steps, hence it takes  $\mathcal{O}(|V|^3 \text{MNB}(|V|))$  time to calculate  $\bigcup_{\alpha \in \mathbb{R}} \mathcal{C}_\alpha(V)$ .  $\square$

Appendix A.15. Calculation of Strong Communities in Figure 3b

As an example, we show how to obtain the dendrogram in Figure 3b and then obtain the strong communities below.

Following the recursive procedure in Theorem 3, to solve  $\mathcal{C}_\alpha(V)$ , first seek to solve  $\mathcal{B}_\alpha^*(V)$ , then for each critical  $\alpha$  value at turning point, solve for  $\mathcal{C}_\alpha(U)$  with similar procedure, where  $U$  is the complement set of  $V$  given by (33). Hence, the procedure to calculate  $\mathcal{B}_\alpha^*(V)$  is representative, which can be done according to Proposition 6.

When  $j = 1$  is the sink node,

$$\begin{aligned} f_\alpha(\{1\}) &= \alpha \cdot |(\{1\})| - g(\{1\}) = \alpha, \\ f_\alpha(\{1, 2\}) &= \alpha \cdot |(\{1, 2\})| - g(\{1, 2\}) = 2\alpha - 4, \\ f_\alpha(\{1, 2, 3\}) &= 3\alpha - 5, \\ f_\alpha(\{1, 2, 3, 4\}) &= 4\alpha - 8, \end{aligned}$$

and  $f_\alpha(\{1, 2, 4\}), f_\alpha(\{1, 3\}), f_\alpha(\{1, 3, 4\}), f_\alpha(\{1, 4\})$  can be written out similarly and we omit them here. Then drawing  $f_\alpha$  against  $\alpha$ , the lowest curve formed is  $\hat{f}_\alpha^{(1)}(V)$ , and the corresponding solution  $\mathcal{B}_\alpha^{(1)}(V)$  can be obtained.

$$\hat{f}_\alpha^{(1)}(V) = \begin{cases} 4\alpha - 8, & \alpha < 2, \\ 2\alpha - 4, & 2 \leq \alpha < 4, \\ \alpha, & \alpha \geq 4, \end{cases}$$

and correspondingly,

$$\mathcal{B}_\alpha^{(1)}(V) = \begin{cases} \{\{1, 2, 3, 4\}\}, & \alpha < 2, \\ \{\{1, 2\}\}, & 2 \leq \alpha < 4, \\ \{\{1\}\}, & \alpha \geq 4, \end{cases}$$

Then, for all  $j \in V$ , calculate  $\hat{f}_\alpha^{(j)}(V)$  and  $\mathcal{B}_\alpha^{(j)}(V)$ , and finally obtain  $T_\alpha^*(V)$  and  $\mathcal{B}_\alpha^*(V)$ . The result is

$$T_\alpha^*(V) = \begin{cases} \{1, 2, 3, 4\}, & \alpha < 2, \\ \{1, 2\}, & 2 \leq \alpha < 4, \\ \{1, 2, 3, 4\}, & \alpha \geq 4, \end{cases}$$

$$\mathcal{B}_\alpha^*(V) = \begin{cases} \{\{1, 2, 3, 4\}\}, & \alpha < 2, \\ \{\{1, 2\}\}, & 2 \leq \alpha < 4, \\ \{\{1\}, \{2\}, \{3\}, \{4\}\}, & \alpha \geq 4, \end{cases}$$

and

$$\mathcal{B}_\alpha^*(V) \setminus \{\{i\} | i \in V\} = \begin{cases} \{\{1, 2, 3, 4\}\}, & \alpha < 2, \\ \{\{1, 2\}\}, & 2 \leq \alpha < 4, \\ \emptyset, & \alpha \geq 4, \end{cases}$$

When  $\alpha < 2$  or  $\alpha \geq 4$ , the  $U$  in (33) satisfies (31), hence the base case applies, no new strong subset is found and the recursion finishes.

For  $2 \leq \alpha < 4$ , we need to continue to calculate  $\mathcal{C}_\alpha(U)$  with  $U = \{3, 4\}$ . With a similar calculation procedure, we can obtain for  $2 \leq \alpha < 4$ ,

$$\mathcal{B}_\alpha^*(U) = \begin{cases} \{\{3, 4\}\}, & 2 \leq \alpha < 3, \\ \{\{3\}, \{4\}\}, & 3 \leq \alpha < 4, \end{cases}$$

and

$$\mathcal{B}_\alpha^*(U) \setminus \{\{i\} | i \in U\} = \begin{cases} \{\{3, 4\}\}, & 2 \leq \alpha < 3, \\ \emptyset, & 3 \leq \alpha < 4. \end{cases}$$

Then the remaining recursions to be done satisfy the base case in (31). Hence all the strong communities are obtained. The dendrogram of the strong communities is shown in Figure 3b.

**Author Contributions:** Conceptualization, C.C.; Formal analysis, C.C.; Investigation, A.A.-B.; Project administration, C.C.; Software, C.Z.; Supervision, C.C.; Visualization, C.Z.; Writing—original draft, C.Z. and C.C.; Writing—review and editing, A.A.-B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Flake, G.W.; Lawrence, S.; Giles, C.L. Efficient identification of web communities. In Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2000), Boston, MA, USA, 20–23 August 2000; Association for Computing Machinery: New York, NY, USA 2000; Volume 2000, pp. 150–160.
2. Radicchi, F.; Castellano, C.; Cecconi, F.; Loreto, V.; Parisi, D. Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 2658–2663. [[CrossRef](#)] [[PubMed](#)]
3. Newman, M.E. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **2004**, *69*, 066133. [[CrossRef](#)] [[PubMed](#)]
4. Javed, M.A.; Younis, M.S.; Latif, S.; Qadir, J.; Baig, A. Community detection in networks: A multidisciplinary review. *J. Netw. Comput. Appl.* **2018**, *108*, 87–111. [[CrossRef](#)]
5. Chintalapudi, S.R.; Prasad, M.K. A survey on community detection algorithms in large scale real world networks. In Proceedings of the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 11–13 March 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1323–1327.
6. Cai, Q.; Ma, L.; Gong, M.; Tian, D. A survey on network community detection based on evolutionary computation. *Int. J. Bio-Inspired Comput.* **2016**, *8*, 84–98. [[CrossRef](#)]
7. Pizzuti, C. Evolutionary computation for community detection in networks: A review. *IEEE Trans. Evol. Comput.* **2017**, *22*, 464–483. [[CrossRef](#)]
8. Jonnalagadda, A.; Kuppasamy, L. A cooperative game framework for detecting overlapping communities in social networks. *Phys. A Stat. Mech. Appl.* **2018**, *491*, 498–515. [[CrossRef](#)]
9. Su, X.; Xue, S.; Liu, F.; Wu, J.; Yang, J.; Zhou, C.; Hu, W.; Paris, C.; Nepal, S.; Jin, D.; et al. A Comprehensive Survey on Community Detection with Deep Learning. *arXiv* **2021**, arXiv:2105.12584.
10. Liu, F.; Xue, S.; Wu, J.; Zhou, C.; Hu, W.; Paris, C.; Nepal, S.; Yang, J.; Yu, P.S. Deep Learning for Community Detection: Progress, Challenges and Opportunities. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, Yokohama, Japan, 11–17 July 2020; pp. 4981–4987.
11. Athey, S.; Calvano, E.; Jha, S. A Theory of Community Formation and Social Hierarchy. *SSRN Electron. J.* **2016** 1–53. [[CrossRef](#)]
12. Gilles, R.P. *The Cooperative Game Theory of Networks and Hierarchies*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2010; Volume 44.

13. Zhou, X.; Zhao, X.; Liu, Y.; Sun, G. A game theoretic algorithm to detect overlapping community structure in networks. *Phys. Lett. A* **2018**, *382*, 872–879. [[CrossRef](#)]
14. Torkaman, A.; Badie, K.; Salajegheh, A.; Bokaei, M.H.; Ardestani, S.F.F. A Four-Stage Algorithm for Community Detection Based on Label Propagation and Game Theory in Social Networks. *AI* **2023**, *4*, 255–269. [[CrossRef](#)]
15. Ferdowsi, F.; Aghababaei Samani, K. Detecting overlapping communities in complex networks using non-cooperative games. *Sci. Rep.* **2022**, *12*, 11054. [[CrossRef](#)]
16. Morgenstern, O.; Von Neumann, J. *Theory of Games and Economic Behavior*; Princeton University Press: Princeton, NJ, USA, 1953.
17. Chalkiadakis, G.; Elkind, E.; Wooldridge, M. *Computational Aspects of Cooperative Game Theory*; Synthesis Lectures on Artificial Intelligence and Machine Learning Series; Springer: Cham, Switzerland, 2011; 168p.
18. Myerson, R.B. *Game Theory: Analysis of Conflict*; Harvard University Press: Cambridge, MA, USA, 1997.
19. Jonnalagadda, A.; Kuppusamy, L. A survey on game theoretic models for community detection in social networks. *Soc. Netw. Anal. Min.* **2016**, *6*, 83. [[CrossRef](#)]
20. Zhou, L.; Lü, K.; Cheng, C.; Chen, H. A game theory based approach for community detection in social networks. In *Big Data, Proceedings of the 29th British National Conference on Databases, Oxford, UK, 8–10 July 2013*; Springer: Berlin/Heidelberg, Germany, 2013, pp. 268–281.
21. Zhou, L.; Lü, K.; Yang, P.; Wang, L.; Kong, B. An approach for overlapping and hierarchical community detection in social networks based on coalition formation game theory. *Expert Syst. Appl.* **2015**, *42*, 9634–9646. [[CrossRef](#)]
22. Lu, Q.; Korniss, G.; Szymanski, B.K. The naming game in social networks: Community formation and consensus engineering. *J. Econ. Interact. Coord.* **2009**, *4*, 221–235. [[CrossRef](#)]
23. Baronchelli, A. A gentle introduction to the minimal naming game. *Belg. J. Linguist.* **2016**, *30*, 171–192. [[CrossRef](#)]
24. Uzun, T.G.; Ribeiro, C.H.C. Detection of communities with Naming Game-based methods. *PLoS ONE* **2017**, *12*, e0182737. [[CrossRef](#)] [[PubMed](#)]
25. Shapley, L.S. Cores of convex games. *Int. J. Game Theory* **1971**, *1*, 11–26. [[CrossRef](#)]
26. Chan, C.; Al-Bashabsheh, A.; Zhao, C. Finding Better Web Communities in Digraphs via Max-Flow Min-Cut. In *Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT), Paris, France, 7–12 July 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 410–414.
27. Bach, F. Learning with submodular functions: A convex optimization perspective. *Found. Trends<sup>®</sup> Mach. Learn.* **2013**, *6*, 145–373. [[CrossRef](#)]
28. Fujishige, S.; Isotani, S. A submodular function minimization algorithm based on the minimum-norm base. *Pac. J. Optim.* **2011**, *7*, 3–17.
29. Granot, F.; McCormick, S.T.; Queyranne, M.; Tardella, F. Structural and algorithmic properties for parametric minimum cuts. *Math. Program.* **2012**, *135*, 337–367. [[CrossRef](#)]
30. Arora, C.; Banerjee, S.; Kalra, P.; Maheshwari, S. Generic cuts: An efficient algorithm for optimal inference in higher order MRF-MAP. In *Computer Vision—ECCV 2012, Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Part V 12*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 17–30.
31. Goldberg, A.V.; Hed, S.; Kaplan, H.; Tarjan, R.E.; Werneck, R.F. Maximum flows by incremental breadth-first search. In *Algorithms—ESA 2011, Proceedings of the 19th Annual European Symposium on Algorithms, Saarbrücken, Germany, 5–9 September 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 457–468.
32. Goldberg, A.V.; Hed, S.; Kaplan, H.; Kohli, P.; Tarjan, R.E.; Werneck, R.F. Faster and more dynamic maximum flow by incremental breadth-first search. In *Algorithms—ESA 2015, Proceedings of the 23th Annual European Symposium on Algorithms, Patras, Greece, 14–16 September 2015*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 619–630.
33. Kolmogorov, V. A faster algorithm for computing the principal sequence of partitions of a graph. *Algorithmica* **2010**, *56*, 394–412. [[CrossRef](#)]
34. Gallo, G.; Grigoriadis, M.D.; Tarjan, R.E. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.* **1989**, *18*, 30–55. [[CrossRef](#)]
35. Lang, K.J.; Andersen, R. Finding dense and isolated submarkets in a sponsored search spending graph. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, Lisbon, Portugal, 6–10 November 2007*; pp. 613–622.
36. Flake, G.W.; Tarjan, R.E.; Tsioutsoulis, K. Graph clustering and minimum cut trees. *Internet Math.* **2004**, *1*, 385–408. [[CrossRef](#)]
37. Newman, M.E. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [[CrossRef](#)]
38. Fortunato, S.; Barthelemy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 36–41. [[CrossRef](#)] [[PubMed](#)]
39. Chen, M.; Nguyen, T.; Szymanski, B.K. A new metric for quality of network community structure. *arXiv* **2015**, arXiv:1507.04308.
40. Lu, X.; Kuzmin, K.; Chen, M.; Szymanski, B.K. Adaptive modularity maximization via edge weighting scheme. *Inf. Sci.* **2018**, *424*, 55–68. [[CrossRef](#)]
41. Auerbach, J.; Galenson, J.; Sundararajan, M. An empirical analysis of return on investment maximization in sponsored search auctions. In *Proceedings of the 2nd International Workshop on Data Mining and Audience Intelligence for Advertising, Las Vegas, NV, USA, 24–27 August 2008*; pp. 1–9.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.