



Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs



Coreness of cooperative games with truncated submodular profit functions[☆]

Wei Chen^c, Xiaohan Shan^{d,*}, Xiaoming Sun^{a,b}, Jialin Zhang^{a,b}

^a CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

^b University of Chinese Academy of Sciences, Beijing, China

^c Microsoft Research Asia, Beijing, China

^d Department of Computer Science and Technology, Tsinghua University, Beijing, China

ARTICLE INFO

Article history:

Received 23 February 2020

Accepted 7 April 2020

Available online xxxx

Keywords:

Cooperative game

Core

Submodular function

Social network

ABSTRACT

Coreness represents solution concepts related to core in cooperative games, which captures the stability of players. Motivated by the scale effect in social networks, economics and other scenario, we study the coreness of cooperative game with truncated submodular profit functions. Specifically, the profit function $f(\cdot)$ is defined by a truncation of a submodular function $\sigma(\cdot)$: $f(\cdot) = \sigma(\cdot)$ if $\sigma(\cdot) \geq \eta$ and $f(\cdot) = 0$ otherwise, where η is a given threshold. In this paper, we study the core and three core-related concepts of truncated submodular profit cooperative game. We first prove that whether core is empty can be decided in polynomial time and an allocation in core also can be found in polynomial time when core is not empty. When core is empty, we show hardness results and approximation algorithms for computing other core-related concepts including *relative least-core value*, *absolute least-core value* and *least average dissatisfaction value*.

© 2020 Published by Elsevier B.V.

1. Introduction

With the wide popularity of social media and social network sites such as Facebook, Twitter, WeChat, etc., social networks have become a powerful platform for spreading information among individuals. Thus, influential users always play important role in a social network. Motivated by this background, influence diffusion in social networks has been extensively studied [1–3].

Most of previous works focus on exploring influential nodes. To the best of our knowledge, there is no study about the “stability” of influential nodes (seed set) when they are treated as a coalition.

Consider the following scenario. A group of influential people in a social network are considering forming a coalition so that they can better serve many advertisers through viral marketing in the social network. To make the coalition stable, we need to design a fair profit allocation scheme among the members of the coalition, such that no individual or a subset of people have incentive to deviate from this coalition, thinking that the allocation to them is unfair and they could earn

[☆] An extended abstract of this paper appears at the proceedings of the 11th International Symposium on Algorithmic Game Theory (SAGT'2018) [Chen et al. (2018)]. This work was supported in part by the 973 Program of China Grant No. 2016YFB1000201, the National Natural Science Foundation of China Grants No. 61832003, 61761136014, 61872334, 61433014, K. C. Wong Education Foundation, NSFC for Distinguished 1062 Young Scholars under Grant No. 61525202 and Beijing 1063 Academy of Artificial Intelligence (BAAI).

* Corresponding author.

E-mail address: shanxiaohan@tsinghua.edu.cn (X. Shan).

<https://doi.org/10.1016/j.tcs.2020.04.004>

0304-3975/© 2020 Published by Elsevier B.V.

more by the deviation and forming an alliance by themselves. A useful and mature framework of studying such incentives for stable coalition formation is the cooperative game theory, and in particular the coreness (core and its related concepts) of the cooperative games [4,5].

First we will motivate our consideration of the truncated submodular functions here. In the above social influence scenario, the typical way of measuring the contribution of any set S of influential people is by its influence spread function $\sigma(S)$, which measures the expected number of people in the social network that could be influenced by S under some stochastic diffusion model. Extensive researches have been done on stochastic diffusion models, and it has been shown that under a large class of models $\sigma(S)$ is both monotone and submodular¹ [2,6,3]. However, the advertisers would only be interested in the coalition as a viral marketing platform when the influence spread reaches certain scale level. In other words, the coalition can only receive profit after the influence spread is above a certain scale threshold η . Therefore, the true profit function for the coalition is $f(S) = \sigma(S)$ when $\sigma(S) \geq \eta$, and $f(S) = 0$ otherwise. We call such f truncated submodular functions.

Both submodularity and scale effect are common in economic behaviors beyond the above example of viral marketing in social networks. Therefore, considering truncated submodular functions as the profit functions is reasonable. In this paper, we study the computational issues related to the coreness of cooperative games with truncated submodular profit functions.

Solution Concepts in Cooperative Games. A cooperative game $\Gamma = (V, \gamma)$ consists of a player set $V = \{1, 2, \dots, n\}$ and a profit function $\gamma : 2^V \rightarrow \mathbb{R}$ with $\gamma(\emptyset) = 0$. A subset of players $S \subseteq V$ is called a *coalition* and V is called the *grand coalition*. For each coalition S , $\gamma(S)$ represents the profit obtained by S without help of other players. An allocation over the players is denoted by a vector $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^{|V|}$ whose components are one-to-one associated with players in V , where $x_i \in \mathbb{R}$ is the value received by player $i \in V$ under allocation x . For any player set $S \subseteq V$, we use the shorthand notation $x(S) = \sum_{i \in S} x_i$. A set of all allocations satisfying some specific requirements is called a *solution concept*.

The *core* [7,8] is one of the earliest and most attractive solution concepts that directly addresses the issue of stability. The core of a game is the set of allocations ensuring that no coalition would have an incentive to split from the grand coalition, and do better on its own. More precisely, the core of a game Γ (denoted by $\mathcal{C}(\Gamma)$), is the following set of allocations: $\mathcal{C}(\Gamma) = \{x \in \mathbb{R}^{|V|} : x(V) = \gamma(V), x(S) \geq \gamma(S), \forall S \subseteq V\}$. In practice, core is very strict and may be even empty in some cases. When $\mathcal{C}(\Gamma)$ is empty, there must be some coalitions becoming dissatisfaction since they can obtain more benefits if they leave the grand coalition and work as a separated team. In this case, we use the dissatisfaction degree (or dissatisfaction value), defined as $dv(S, x) = \max\{\gamma(S) - x(S), 0\}$, to capture the instability of player set S with respect to the allocation x . Then, the overall stability of the game can be measured as either the worst-case or average-case dissatisfaction degree, for which we consider the following three versions.

The first one is the *relative least-core value* (\mathcal{RLCV}) [9], which reflects the relative stability, i.e. the minimum value of the maximum proportional difference between the profits and the payoffs among all coalitions.

Definition 1. Given a cooperative game Γ , the *relative least-core value* of Γ ($\mathcal{RLCV}(\Gamma)$) is $\min_x \max_S \frac{dv(S, x)}{\gamma(S)}$. Technically, $\mathcal{RLCV}(\Gamma)$ is the optimal solution of the following linear programming:

$$\begin{aligned} \min \quad & r \\ \text{s.t.} \quad & \begin{cases} x(V) = \gamma(V) \\ x(S) \geq (1-r)\gamma(S) & \forall S \subseteq V \\ x(\{i\}) \geq 0 & \forall i \in V \end{cases} \end{aligned} \quad (1)$$

The second one is the *absolute least-core value* (\mathcal{ALCV}) [10] which reflects the absolute stability, i.e. the minimum value of the maximum difference between the profits and the payoffs among all coalitions. The formal definition is as follows.

Definition 2. Given an cooperative game Γ , the *absolute least-core value* of Γ ($\mathcal{ALCV}(\Gamma)$) is $\min_x \max_S dv(S, x)$. Technically, $\mathcal{ALCV}(\Gamma)$ is the optimal solution of the following linear programming:

$$\begin{aligned} \min \quad & \varepsilon \\ \text{s.t.} \quad & \begin{cases} x(V) = \gamma(V) \\ x(S) \geq \gamma(S) - \varepsilon & \forall S \subseteq V \\ x(\{i\}) \geq 0 & \forall i \in V \end{cases} \end{aligned} \quad (2)$$

The above two classical least-core values capture the stability from the perspective of the most dissatisfied coalition i.e. the worst case of stability. Sometimes the worst case is too extreme to reflect the real stability. Thus, we introduce the *least average dissatisfaction value* (\mathcal{LADV}) which reflects the minimum value of average dissatisfaction degree among all coalitions.

¹ A set function f is monotone if $f(S) \leq f(T)$ for all $S \subseteq T$, and is submodular if $f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$ for all $S \subseteq T$ and $u \notin T$.

Definition 3. Given a cooperative game Γ , the *least average dissatisfaction value* of Γ ($\mathcal{LADV}(\Gamma)$) is $\min_x \mathbb{E}_S(dv(S, x))$. Technically, $\mathcal{LADV}(\Gamma)$ is the optimal value of the following linear programming:

$$\begin{aligned} \min \quad & \frac{1}{2^n} \sum_{S \subseteq V} \max\{\gamma(S) - x(S), 0\} \\ \text{s.t.} \quad & \begin{cases} x(V) = \gamma(V) \\ x(\{i\}) \geq 0 \end{cases} \quad \forall i \in V \end{aligned} \quad (3)$$

In this paper, we consider the following computational problems in the context of truncated submodular functions: (a) Whether the core of a given cooperative game is empty? (b) How to find an allocation in core if the core is not empty? (c) If the core is empty, how to compute the relative least-core value, the absolute least-core value and the least average dissatisfaction value of a cooperative game?

Contributions. We study coreness (solution concepts related to core) of truncated submodular profit cooperative game Γ_f . We consider computational properties of the core, the relative least-core value, the absolute least-core value and the least average dissatisfaction value of Γ_f , which are denoted by $\mathcal{C}(\Gamma_f)$, $\mathcal{RLCV}(\Gamma_f)$, $\mathcal{ALCV}(\Gamma_f)$ and $\mathcal{LADV}(\Gamma_f)$, respectively.

We first prove that checking the non-emptiness of $\mathcal{C}(\Gamma_f)$ can be done in polynomial time. Moreover, we can find an allocation in the core if the core is not empty. Next, we consider the case when the core is empty. For the problem of computing the relative least-core value ($\mathcal{RLCV}(\Gamma_f)$), we show that it is in general NP-hard, but when truncation threshold $\eta = 0$, there is a polynomial time algorithm. Along the way, we also find an interesting partial result showing that there is no polynomial time separation oracle for the $\mathcal{RLCV}(\Gamma_f)$'s linear program unless $P=NP$, which is of independent interest since it reveals close connections with a new class of combinatorial problems. For the absolute least-core value problem $\mathcal{ALCV}(\Gamma_f)$, we prove that finding $\mathcal{ALCV}(\Gamma_f)$ is APX-hard even when $\sigma(\cdot)$ is defined as the influence spread under the classical independent cascade (IC) model in social network. We also prove that there exists a polynomial time algorithm which can guarantee an additive term approximation. Finally, for the least average dissatisfaction value problem $\mathcal{LADV}(\Gamma_f)$, we show that we can use the stochastic gradient descent algorithm to compute $\mathcal{LADV}(\Gamma_f)$ to an arbitrary small additive error.

Related Work. Cooperative game theory is a branch of (micro-)economics that studies the behavior of self-interested agents in strategic settings where binding agreements between agents are possible [11]. Numerous classical studies about cooperative game provide rich mathematical framework to solve issues related to cooperation in multi-agent systems [12–14]. [15] studies the approximation of the absolute least core value of supermodular cost cooperative games, the results in this paper can be generalized to submodular profit cooperative games. An important application of our study is to analyze the stability of influential people in social networks. Almost all the existing studies focus on selecting seed set [16–20]. To the best of our knowledge, there is no literature considering the stability of the selected seed set. We utilize cooperative game theory to analyse the stability of seed set, and generalize it to a generic cooperative game with truncated submodular functions. The truncated operation represents the “threshold effect” which has been studied widely in literature [21,22].

2. Model and problems

2.1. Cooperative games with truncated submodular profit functions

A truncated submodular profit cooperative game is denoted by $\Gamma_f = (V, f(\cdot))$.

In Γ_f , V is the player set and $f(\cdot)$ is the profit function which is defined as follows:

$$f(S) = \begin{cases} \sigma(S), & \text{if } \sigma(S) \geq \eta \\ 0, & \text{if } \sigma(S) < \eta \end{cases}$$

Note that $\sigma(\cdot)$ is a nonnegative monotone increasing submodular function with $\sigma(\emptyset) = 0$ and $0 \leq \eta \leq \sigma(V)$ is a nonnegative threshold. To express clearly, in the left of this paper, a truncated submodular profit cooperative game is denoted by a triple form $(V, \sigma(\cdot), \eta)$.

Note that the explicit representation of $\sigma(\cdot)$ might be exponential in the size of V . The standard way to bypass this difficulty is to assume that $\sigma(\cdot)$ is given by a value oracle.

2.2. Computational problems on the coreness

Given an truncated submodular profit cooperative game Γ_f , we focus on the following problems:

CORE: Is $\mathcal{C}(\Gamma_f) \neq \emptyset$ and how to find an allocation in $\mathcal{C}(\Gamma_f)$ when $\mathcal{C}(\Gamma_f) \neq \emptyset$?

ALCV: When $\mathcal{C}(\Gamma_f) = \emptyset$, how to compute $\mathcal{ALCV}(\Gamma_f)$?

RLCV: When $\mathcal{C}(\Gamma_f) = \emptyset$, how to compute $\mathcal{RLCV}(\Gamma_f)$?

LADV: When $\mathcal{C}(\Gamma_f) = \emptyset$, how to compute $\mathcal{LADV}(\Gamma_f)$?

Before we analyze the above problems, we introduce a specific instance of truncated submodular profit cooperative game (see Section 2.3).

2.3. Influence cooperative game (Γ_{inf})

As the description in our introduction, an important motivation of our model is influence in social networks. In this section, we introduce a specific instance of truncated submodular profit cooperative game, *influence cooperative game*.

Social graph. A social graph is a directed graph $G = (V \cup U, E; P)$, where $V \cup U$ is the vertex set and E is the edge set. $P = \{p_e\}_{e \in E}$ and p_e is the influence probability on each edge $e \in E$. Note that, V and U denote the vertex set of influential people and target people in G , respectively.

Influence diffusion model. The information diffusion process follows the independent cascade (IC) model proposed by [2]. In the IC model, discrete time steps $t = 0, 1, 2, \dots$ are used to model the diffusion process. Each node in G has two states: inactive or active. At step 0, nodes in seed set S are active and other nodes are inactive. For any step $t \geq 1$, if a node u is newly active at step $t - 1$, u has a single chance to influence each of its inactive out-neighbor v with independent probability p_{uv} to make v active. Once a node becomes active, it will never return to the inactive state. The diffusion process stops when there is no new active nodes at a time step. For any $S \subseteq V$, we use $\sigma^{\text{IC}}(S)$ to denote the influence spread of S , the expected number of activated nodes in U from seed set $S \subseteq V$, at the end of an IC diffusion. According to [2], $\sigma^{\text{IC}}(\cdot)$ is a monotone submodular function.

Definition 4. An *influence cooperative game* $\Gamma_{\text{inf}} = (V, \sigma^{\text{IC}}(\cdot), \eta)$ is a special form of the truncated cooperative game, with V as the player set, and the truncation of influence spread function $\sigma^{\text{IC}}(\cdot)$ as the profit function.

In the rest of this paper, we analyze problems defined in Section 2.2 one by one. Note that our positive results (properties and algorithms) could apply to all truncated submodular profit cooperative games including influence cooperative game. Our hardness results are established for the influence cooperative games, so it is stronger than the hardness results for general truncated submodular cooperative games.

3. Computing core

We start by considering the core of Γ_f ($\mathcal{C}(\Gamma_f)$). In Γ_f , we say a player $i \in V$ is a *veto player* if $\sigma(S) < \eta$ for any $S \subseteq V \setminus \{i\}$. That is to say, a successful coalition must include all veto players.

Lemma 1. $\mathcal{C}(\Gamma_f) \neq \emptyset$ if and only if:

- (i) There exists at least one veto player in Γ_f , or
- (ii) $\sigma(S) = \sum_{i \in S} \sigma(\{i\})$, for any $S \subseteq V$.

Proof. Suppose the player set of Γ_f is $V = \{1, 2, \dots, n\}$. We first prove the sufficiency of Lemma 1. On one hand, suppose i is a veto player of Γ_f , then we can find a trivial allocation x in $\mathcal{C}(\Gamma_f)$: $x(\{i\}) = \sigma(V)$ and $x(\{j\}) = 0$, $\forall j \in V \setminus \{i\}$. On the other hand, $x(\{i\}) = \sigma(\{i\})$ ($\forall i \in V$) is an allocation in $\mathcal{C}(\Gamma_f)$ if $\sigma(S) = \sum_{i \in S} \sigma(\{i\})$.

Now we prove the necessity. Suppose $\mathcal{C}(\Gamma_f) \neq \emptyset$ and $x \in \mathcal{C}(\Gamma_f)$. Let $\sigma(V) = \sum_{i=1}^n M_i$, where $M_i = \sigma(\{1, 2, \dots, i\}) - \sigma(\{1, 2, \dots, i-1\})$ is the marginal increasing of player i . If there is no veto player, then for any $i \in V$, $\sigma(V \setminus \{i\}) \geq \eta$ since $\sigma(S)$ is monotone. Thus, $f(V \setminus \{i\}) = \sigma(V \setminus \{i\})$, $\forall i \in V$. Suppose $\sigma(V \setminus \{i\}) = \sum_{j=1}^{i-1} M_j + \sum_{j=i+1}^n M'_j$, where $M'_{ij} = \sigma(\{1, 2, \dots, i-1, i+1, \dots, j\}) - \sigma(\{1, 2, \dots, i-1, i+1, \dots, j-1\})$. Note that $M'_{ij} \geq M_j$ since $\sigma(S)$ is submodular. By the definition of the core, for any $i \in \{1, 2, \dots, n\}$, we have: $x(V \setminus \{i\}) \geq f(V \setminus \{i\}) = \sigma(V \setminus \{i\})$. That is, $x(V) - x(\{i\}) \geq \sum_{j=1}^{i-1} M_j + \sum_{j=i+1}^n M'_j$, $\forall i \in V$.

Summing up these inequalities for all $i \in V$, we have, $(n-1) \sum_{i=1}^n x(\{i\}) \geq \sum_{i=1}^n (\sum_{j=1}^{i-1} M_j + \sum_{j=i+1}^n M'_{ij}) \geq \sum_{i=1}^n (\sum_{j=1}^{i-1} M_j + \sum_{j=i+1}^n M_j) = \sum_{i=1}^n (\sigma(V) - M_i) = (n-1) \sigma(V)$.

We have known that $\sum_{i=1}^n x(\{i\}) = \sum_{j=1}^n M_j = \sigma(V)$ and then $M_j = M'_{ij}$, $\forall i, j \in V$. Thus, $\sigma(S) = \sum_{i \in S} \sigma(\{i\})$. \square

An important application of Lemma 1 is Theorem 1.

Theorem 1. Deciding whether $\mathcal{C}(\Gamma_f)$ is empty can be done in polynomial time and an allocation in $\mathcal{C}(\Gamma_f)$ can be computed in polynomial time if $\mathcal{C}(\Gamma_f)$ is not empty.

Proof (Sketch). First, it takes polynomial time to check the non-emptiness of $\mathcal{C}(\Gamma_f)$. When $\mathcal{C}(\Gamma_f)$ is not empty, then $(x_j = \sigma(V), \mathbf{0}_{\{i: i \neq j\}}) \in \mathcal{C}(\Gamma_f)$ when j is a veto player and $(\sigma(\{1\}), \dots, \sigma(\{n\})) \in \mathcal{C}(\Gamma_f)$ when (ii) satisfies. \square

The detail proof of Theorem 1 is shown in the appendix.

4. Computing relative least-core value

From Lemma 1, $\mathcal{C}(\Gamma_f)$ may be empty in many cases. It is obvious that $\mathcal{RLCV}(\Gamma_f) > 0$ if $\mathcal{C}(\Gamma_f) = \emptyset$ and $\mathcal{RLCV}(\Gamma_f) = 0$ otherwise. In this section, we study computational properties of RLCV problem. The linear programming corresponding to $\mathcal{RLCV}(\Gamma_f)$ (LP-RLCV) is as follows:

$$\begin{aligned} \min \quad & r \\ \text{s.t.} \quad & \begin{cases} x(V) = \sigma(V) \\ x(S) \geq (1-r)\sigma(S) \quad \forall S \subseteq V, \sigma(S) \geq \eta \\ x(\{i\}) \geq 0 \quad \forall i \in V \end{cases} \end{aligned} \quad (4)$$

A special case of computing $\mathcal{RLCV}(\Gamma_f)$ is when $\eta = 0$. It captures the scenario that the profit of any coalition exactly equals to its influence spread under influence cooperative game. In Theorem 2 we show that, although there are exponential number of constraints, LP-RLCV can be solved in polynomial time by providing a polynomial time separation oracle when $\eta = 0$. A separation oracle for a linear program is an algorithm that, given a putative feasible solution, checks whether it is indeed feasible, and if not, outputs a violated constraint. It is known that a linear program can be solved in polynomial time by the ellipsoid method as long as it has a polynomial time separation oracle [23].

Theorem 2. *There exists a polynomial time separation oracle of LP-RLCV when $\eta = 0$. Therefore, RLCV can be solved in polynomial time when $\eta = 0$.*

Proof. Given any solution candidate of LP-RLCV (x', r') , we need to either assert (x', r') is a feasible solution or find a constraint in LP-RLCV such that (x', r') violates it. Note that, checking $x'(V) = \sigma(V)$ and $x'(\{i\}) \geq 0$ ($\forall i \in V$) can be done in polynomial time. Thus, we only need to check whether $g(S) \triangleq 1 - x'(S)/\sigma(S) \leq r'$, $\forall S \subseteq V$.

An important property is $g(S)$ achieves its maximum value when S contains only one single player. This is because $g(S) = 1 - \frac{x'(S)}{\sigma(S)} \leq 1 - \frac{\sum_{i \in S} x'_i}{\sum_{i \in S} \sigma(i)} \leq 1 - \min_{i \in S} \{ \frac{x'_i}{\sigma(i)} \} = \max_{i \in S} \{ g(\{i\}) \}$. The first inequality is due to the submodularity of $\sigma(S)$ and the second inequality is due to $\min_{i \in [n]} \{ \frac{a_i}{b_i} \} \leq \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i}$, $\forall a_i, b_i \in \mathbb{R}$. Thus, the exponential number of constraints can be simplified to n constraints on all single players. Then, we can find a polynomial time separation oracle of LP-RLCV directly. \square

When $\eta = 0$, RLCV can be solved in polynomial time is mainly because the most dissatisfaction coalition is a single player. However, when $\eta \neq 0$, it becomes intractable to find the most dissatisfaction coalition.

Theorem 3. *There is no polynomial time separation oracle of LP-RLCV for some $\eta > 0$, unless $P=NP$.*

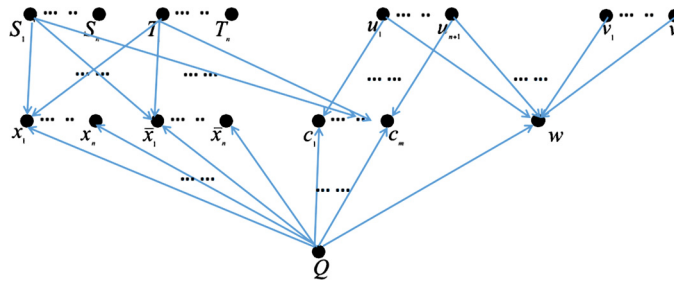
Theorem 3 can not imply the NP-hardness of RLCV. However, the proof of Theorem 3 reveals an interesting connection between RLCV problem and a series of well defined combinatorial problems. We will report the proof of Theorem 3 and the generalized combinatorial problems in the appendix.

In the left of this section, we prove the NP-hardness of RLCV, a stronger hardness result than which in Theorem 3.

Theorem 4. *It is NP-hard to compute $\mathcal{RLCV}(\Gamma_f)$, even under influence cooperative game.*

Proof (Sketch). We construct a reduction from the SAT problem. A boolean formula is in conjunctive normal form (CNF) if it is expressed as an AND of clauses, each of which is the OR of one or more literals. The SAT problem is defined as follows: given a CNF formula F , determine whether F has a satisfiable assignment. Let F be a CNF formula with m clauses C_1, C_2, \dots, C_m , over n literals z_1, z_2, \dots, z_n . Without loss of generality, we set $m > 4n$.

We construct a social graph G as follows: $G = (V_1 \cup V_2 \cup V_3, E)$ is a tripartite graph (see the sketch graph in Fig. 1). In the first layer (V_1), there are two nodes S_i and T_i corresponding to each $i \in \{1, 2, \dots, n\}$, $n+1$ dummy nodes labeled as u_1, u_2, \dots, u_{n+1} and n dummy nodes labeled as v_1, v_2, \dots, v_n . In the second layer (V_2), there are two nodes x_i and \bar{x}_i corresponding to each $i \in \{1, 2, \dots, n\}$, one node c_j for each $j \in \{1, 2, \dots, m\}$ and a dummy node w . The third layer (V_3) contains only node Q . Edges exist only between the adjacent layers. For each $i \in \{1, 2, \dots, n\}$, S_i sends an edge to every node in $\{x_i, \bar{x}_i\} \cup \{c_j: \text{clause } C_j \text{ contains literal } z_i, j \in \{1, 2, \dots, m\}\}$. Similarly, for each $i \in \{1, 2, \dots, n\}$, T_i sends an edge to every node in $\{x_i, \bar{x}_i\} \cup \{c_j: \text{clause } C_j \text{ contains literal } \bar{z}_i, j \in \{1, 2, \dots, m\}\}$. The probabilities on edges sent from S_i and T_i are 1. There is an edge with influence probability 1 from u_i to c_i for any $i \in \{1, 2, \dots, n\}$ and $m-n$ edges from u_{n+1} to $c_{n+1}, c_{n+2}, \dots, c_m$. There is an edge from u_i to w with influence probability $1 - \sqrt[n+1]{1/2}$ for any $i \in \{1, 2, \dots, n+1\}$. There is also exists an edge from v_i to w with influence probability $1 - \sqrt[n]{1/2}$ for any $i \in \{1, 2, \dots, n\}$. The left edges are from Q to all nodes in the second layer. The influence probability on edge (Q, w) is $1/2$ and all other probabilities on edges sent from Q is 1. The influence cooperative game defined on G is $\Gamma(G) = (V = V_1 \cup V_3, \sigma^{IC}(\cdot), \eta = 2n + m + 1/2)$. For convenient, we set $N = 2n + m$.

Fig. 1. The reduction from SAT to $\mathcal{RLCV}(\Gamma_f)$.

Suppose r^* is the optimal solution of the relative least-core value of $\Gamma(G)$. We can prove that $r^* \geq 1 - \frac{1}{3}(N + \frac{7}{8})/(N + \frac{1}{2})$ if F is satisfiable and $r < 1 - \frac{1}{3}(N + \frac{7}{8})/(N + \frac{1}{2})$ if F is unsatisfiable. The proof of this part is shown in the appendix. \square

5. Computing absolute least-core value

5.1. Hardness of ALCV

Theorem 5. *ALCV problem of influence cooperative game cannot be approximated within 1.139 under the unique games conjecture.*

Proof (Sketch). We construct a reduction from MAX-CUT problem. Under our construction, for any instance of MAX-CUT problem, we can construct an instance of ALCV problem such that the optimal solution of these two instances are equal. The detail proof is shown in our appendix. \square

5.2. Approximating $\mathcal{ALCV}(\Gamma)$

In this section, we approximate $\mathcal{ALCV}(\Gamma_f)$ by approximating the following linear programming (LP-PRIME):

$$\begin{aligned} \min \quad & \varepsilon \\ \text{s.t.} \quad & \begin{cases} x(V) = \sigma(V) \\ x(\{S\}) \geq \sigma(\{S\}) - \varepsilon & \forall S \subseteq V, \sigma(S) \geq \eta \\ x(\{u\}) \geq 0 & \forall u \in V \end{cases} \end{aligned}$$

The intractability of LP-PRIME lies on the exponential number of constraints and the hardness of identifying all successful coalitions. We use a relaxed version LP-RE and a strengthen version LP-STR of LP-PRIME to design an approximation algorithm of $\mathcal{ALCV}(\Gamma_f)$. (5) and (6) are formal definitions of LP-RE and LP-STR, respectively.

$$\begin{aligned} \min \quad & \varepsilon \\ \text{s.t.} \quad & \begin{cases} x(V) = \sigma(V) \\ x(S) \geq \eta - \varepsilon & \forall S \subseteq V, \sigma(S) \geq \eta \\ x(\{u\}) \geq 0 & \forall u \in V \end{cases} \end{aligned} \quad (5)$$

$$\begin{aligned} \min \quad & \varepsilon \\ \text{s.t.} \quad & \begin{cases} x(V) = \sigma(V) \\ x(S) \geq \sigma(S) - \varepsilon & \forall S \subseteq V \\ x(\{u\}) \geq 0 & \forall u \in V \end{cases} \end{aligned} \quad (6)$$

Intuitively, LP-RE and LP-STR denote absolute least-core values of two cooperative games with new profit functions. Specifically, LP-RE relaxes the constraints in LP-PRIME by reducing the profits of all successful coalitions excepting V to η . Formally, the profit function in LP-RE is $g(S)$: $g(V) = \sigma(V)$, $\forall S \subset V$, $g(S) = \eta$ if $\sigma(S) \geq \eta$ and $g(S) = 0$ otherwise. The profit function in LP-STR is $h(S) = \sigma(S)$, $\forall S \subseteq V$. Clearly, LP-STR strengthens LP-PRIME by increasing the profits of all unsuccessful coalitions.

Our main result in this section is shown in Theorem 6.

Theorem 6. $\forall \delta > 0$, there exists an approximate algorithm \mathcal{A} of the $\mathcal{ALCV}(\Gamma_f)$ problem with running time in $\text{poly}(n, 1/\delta, \log \sigma(V))$, \mathcal{A} outputs ε'_p such that $\varepsilon_p^* \leq \varepsilon'_p \leq \min\{\varepsilon_p^* + \sigma(V) - \eta + 2\delta, \max\{3\varepsilon_p^*, \eta\}\}$.

We prove Theorem 6 by show Lemma 2, Lemma 3 and Lemma 4 in order.

Lemma 2. Suppose the optimal value of LP-PRIME, LP-RE and LP-STR are ε_p^* , ε_r^* and ε_s^* , respectively. Then, we have

$$\varepsilon_p^* \leq \varepsilon_r^* + (\sigma(V) - \eta) \leq \varepsilon_p^* + (\sigma(V) - \eta), \quad (7)$$

$$\varepsilon_p^* \leq \varepsilon_s^* \leq \max\{\varepsilon_p^*, \eta\}. \quad (8)$$

Lemma 3. There exists a polynomial time approximate algorithm of LP-STR outputting ε'_s such that $\varepsilon_s^* \leq \varepsilon'_s \leq 3\varepsilon_s^*$.

Lemma 4. $\forall \delta > 0$, there exists an algorithm of LP-RE outputting ε'_r such that $\varepsilon_r^* \leq \varepsilon'_r \leq \varepsilon_r^* + 2\delta$, with runs time in $\text{poly}(n, 1/\delta, \log \sigma(V))$.

The proofs of Lemma 2 to Lemma 4 rely heavily on mathematical computation and we report them in the appendix.

6. Computing least average dissatisfaction value

Based on Definition 3, $\mathcal{LADV}(\Gamma_f)$ equals the optimal value of the following linear programming:

$$\begin{aligned} \min \quad & F(x) = \frac{1}{2^n} \sum_{S \subseteq V} \max\{f(S) - x(S), 0\} \\ \text{s.t.} \quad & \begin{cases} x(V) = \sigma(V) \\ x(\{i\}) \geq 0 \end{cases} \quad \forall i \in V \end{aligned} \quad (9)$$

Where $f(S) = \sigma(S)$ if $\sigma(S) \geq \eta$ and $f(S) = 0$ otherwise. There are exponential terms in $F(x)$, however, we can utilize stochastic gradient algorithm to approximate the optimal solution of (9). This is because the object function $F(x)$ is a convex function (Lemma 5) and the feasible solution area in (9) is a convex set.

Lemma 5. $F(x)$ is a convex function.

The proof of Lemma 5 is shown in our appendix. The stochastic gradient descent algorithm (cf. [24]) can be used to compute $\mathcal{LADV}(\Gamma_f)$ (see Algorithm 1).

Algorithm 1 Stochastic gradient descent for LADV.

```

1: Parameters: Scalar  $\alpha > 0$ , integer  $T > 0$ 
2: Initialize:  $\mathbf{X}^1 = \mathbf{0}$ ,  $t = 0$ .
3: Set  $D = \{\mathbf{X} : \mathbf{X}_i \geq 0 (\forall i \in V), \sum_{i \in V} \mathbf{X}_i = \sigma(V)\}$ .
4: for  $t = 1$  to  $T$  do
5:   /*choose a random  $\mathbf{Y}^t$  such that  $\mathbb{E}[\mathbf{Y}^t | \mathbf{X}^t]$  is a subgradient of  $F$ .*/
6:   Uniformly at random choose a set  $S \in 2^V$ .
7:   if  $f(S) \geq \mathbf{X}^t(S)$  then
8:     Set  $\mathbf{Y}^t = (-\mathbf{1}_S, \mathbf{0}_{V \setminus S})$ .
9:   else
10:    Set  $\mathbf{Y}^t = \mathbf{0}$ .
11:   end if
12:   update  $\mathbf{X}^{t+\frac{1}{2}} = \mathbf{X}^t - \alpha \mathbf{Y}^t$ .
13:   /*Project  $\mathbf{X}^{t+\frac{1}{2}}$  to  $D$ */
14:    $\mathbf{X}^{t+1} = \arg \min_{\mathbf{X} \in D} \|\mathbf{X} - \mathbf{X}^{t+\frac{1}{2}}\|^2$ .
15: end for
16: return  $\hat{F} = \min\{F(\mathbf{X}^t)\}_{t \in \{1, 2, \dots, T\}}$ .
```

Let F^* be the optimal solution of $\mathcal{LADV}(\Gamma_f)$, \hat{F} be the output of Algorithm 1 and the profit of grand coalition $\sigma(V) = V$. Then, the performance of Algorithm 1 can be formalized in the following theorem.

Theorem 7. $\forall \varepsilon > 0$, $\mathbb{E}[\hat{F}] - F^* \leq \varepsilon$ if $T \geq \frac{\sigma(V)^4 n^4}{\varepsilon^2}$ and $\alpha = \sqrt{\frac{\sigma(V)^4}{T n^4}}$ in Algorithm 1.

Following the analysis in Chapter 14 of [24], Theorem 7 holds since it is easy to check that $\mathbb{E}[\mathbf{Y}^t | \mathbf{X}^t]$ is a subgradient of $F(\mathbf{X})$ at node \mathbf{X}^t , for any $t \in [T]$ (line 6 - line 11 in Algorithm 1).

7. Conclusion and future work

In this paper, we study the core related solution concepts of truncated submodular profit cooperative game. One possible future work is to change the way of truncating a function. For example, we can set $f(S) = \sigma(S)$ if $|S| \geq k$ and $f(S) = 0$ otherwise. This setting is a special case of the setting in our paper and maybe we can try to design algorithms for it. In this paper, we prove that computing the relative least-core value is NP-hard. We also prove that the relative least-core value can

be solved in polynomial time in a special case. A directly future work is to design an approximate algorithm of RLCV under general case.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Appendix of section 3

Proof of Theorem 1. We can design the following polynomial time process to check the emptiness of $\mathcal{C}(\Gamma_f)$ and find an allocation in $\mathcal{C}(\Gamma_f)$ when $\mathcal{C}(\Gamma_f) \neq \emptyset$.

Step 1: Query $\sigma(V \setminus \{i\})$ for all $i \in V$ from value oracle. If there exists $j \in V$ such that $\sigma(V \setminus \{j\}) < \eta$, go to Step 2, otherwise, go to Step 3.

Step 2: Return $x = (0, \dots, x_j = \sigma(V), 0, \dots, 0) \in \mathcal{C}(\Gamma_f)$.

Step 3: Query $\sigma(V)$ and $\sigma(\{i\})$ for all $i \in V$ from value oracle. If $\sigma(V) = \sum_{i \in V} \sigma(\{i\})$, go to Step 4, otherwise, go to Step 5.

Step 4: Return $x = (\sigma(\{1\}), \dots, \sigma(\{n\})) \in \mathcal{C}(\Gamma_f)$.

Step 5: Assert that $\mathcal{C}(\Gamma_f) = \emptyset$. \square

Appendix B. Appendix of Section 4

Proof of Theorem 3. We construct a reduction from NP-complete problem DOMINANT-SET [25]. Given an undirected graph $G = (V, E)$ and an integer $k \in \mathbb{N}$, the DOMINANT-SET problem concerns testing whether there exists a dominant set of G with size no more than k . A dominant set is a subset $S \subseteq V$ such that each vertex in $V \setminus S$ is adjacent to at least one vertex in S .

Given any instance of DOMINANT-SET problem $(G = (V, E); k)$, we construct a social graph G' as follows: The vertex set in G' is $V' = V_1 \cup V_2$, where $V_1 = V_2 = V$. For each node $i \in V_1$ and $j \in V_2$, there is a directed edge (i, j) in G' if and only if either $(i, j) \in E$ in G or $i = j$. The influence probability on each edge is 1.

The influence cooperative game defined on G' is $\Gamma(G') = (V_1, \sigma^{\text{IC}}(\cdot), \eta = |V_1| = n)$. Thus, the linear programming corresponding to the relative least-core value of $\Gamma(G')$ is:

$$\begin{aligned} \min \quad & r \\ \text{s.t.} \quad & \begin{cases} x(V_1) = n \\ x(S) \geq (1-r)n & \forall S \subseteq V_1, \sigma^{\text{IC}}(S) = n \\ x(\{i\}) \geq 0 & \forall i \in V_1 \end{cases} \end{aligned} \quad (\text{B.1})$$

Now we prove that the DOMINANT-SET problem can be solved in polynomial time if there exists a polynomial time separation oracle of (B.1). Given a candidate solution (x', r') , where $x'_i = 1$ for any $i \in V_1$ and $r' = 1 - (k+1)/n$. Suppose there exists a polynomial time separation oracle \mathcal{O} of (B.1). Then $\forall S \subseteq V_1, \sigma^{\text{IC}}(S) = n$, we can decide whether $|S| \geq (\frac{k+1}{n})n = k+1$ in polynomial time. Note that $\{S : S \subseteq V_1, \sigma^{\text{IC}}(S) = n\}$ is the set of all dominant sets of G . Thus, for any G 's dominant set S , (x', r') is a feasible solution if and only if $|S| \geq k+1$. In other words, having \mathcal{O} , we can decide whether there exists a dominant set with size no more than k . \square

In Remark 1, we introduce a class of combinatorial optimization problems inspired from the proof process of Theorem 3.

Remark 1. We define an *adversarial* version of the classical weighted set cover problem: Given a ground set U , a collection of subsets $S \subseteq 2^U$, a weight budget M . The objective of the *adversarial* weighted set cover problem is to allocate weight among subsets in \mathcal{S} such that the minimum weight of all set covers is maximum.

Formally, the objective of the *adversarial* weighted set cover problem is:

$$\max_{w: \sum_{S \in \mathcal{S}} w(S) \leq M} \min_{\mathcal{C}: \mathcal{C} \text{ is a set cover}} \sum_{S \in \mathcal{C}} w(S),$$

where w is a nonnegative allocation vector.

Similarly, we can define *adversarial* weighted vertex cover problem, *adversarial* weighted dominant set problem, and so on.

The following argument shows that the *adversarial* weighted set cover (dominant set, vertex cover, etc.) problem is a special instance of RLCV. When $\eta = \sigma(V) = M$, $\mathcal{RLCV}(\Gamma_f)$ can be denoted more compactly:

$$\mathcal{RLCV}(\Gamma_f) = \min_{x: x(V) = M, x \geq 0} \max_{S: \sigma(S) = M} (1 - \frac{x(S)}{M}).$$

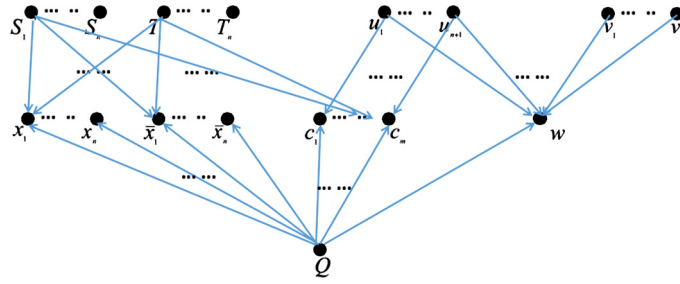


Fig. 2. The reduction from SAT to $\mathcal{RLCV}(\Gamma_f)$.

Thus, it is enough to compute

$$\max_{x: x(V)=M, x \geq 0} \min_{S: \sigma(S)=M} x(S).$$

Given any instance of set cover problem, similar to the construction in the proof of Theorem 3, it is not difficult to construct a social graph such that $\sigma^{\text{LC}}(S)$ equals the number of elements covered by S , for any collection S . Thus, the adversarial weighted set cover problem is a special instance of RLCV problem.

Proof of Theorem 4. We reduce from the SAT problem. A boolean formula is in conjunctive normal form (CNF) if it is expressed as an AND of clauses, each of which is the OR of one or more literals. The SAT problem is defined as follows: given a CNF formula F , determine whether F has a satisfiable assignment. Let F be a CNF formula with m clauses C_1, C_2, \dots, C_m , over n literals z_1, z_2, \dots, z_n . Without loss of generality, we set $m > 4n$.

We construct a social graph G as follows: $G = (V_1 \cup V_2 \cup V_3, E)$ is a tripartite graph (see the sketch graph in Fig. 2). In the first layer (V_1), there are two nodes S_i and T_i corresponding to each $i \in \{1, 2, \dots, n\}$, $n+1$ dummy nodes labelled as u_1, u_2, \dots, u_{n+1} and n dummy nodes labelled as v_1, v_2, \dots, v_n . In the second layer (V_2), there are two nodes x_i and \bar{x}_i corresponding to each $i \in \{1, 2, \dots, n\}$, one node c_j for each $j \in \{1, 2, \dots, m\}$ and a dummy node w . The third layer (V_3) contains only node Q . Edges exist only between the adjacent layers. For each $i \in \{1, 2, \dots, n\}$, S_i sends an edge to every node in $\{x_i, \bar{x}_i\} \cup \{c_j : \text{clause } C_j \text{ contains literal } z_i, j \in \{1, 2, \dots, m\}\}$. Similarly, for each $i \in \{1, 2, \dots, n\}$, T_i sends an edge to every node in $\{x_i, \bar{x}_i\} \cup \{c_j : \text{clause } C_j \text{ contains literal } \bar{z}_i, j \in \{1, 2, \dots, m\}\}$. The probabilities on edges sent from S_i and T_i are 1. There is an edge with influence probability 1 from u_i to c_i for any $i \in \{1, 2, \dots, n\}$ and $m-n$ edges from u_{n+1} to $c_{n+1}, c_{n+2}, \dots, c_m$. There is an edge from u_i to w with influence probability $1 - \frac{n+1}{\sqrt{1/2}}$ for any $i \in \{1, 2, \dots, n+1\}$. There is also exists an edge from v_i to w with influence probability $1 - \frac{n}{\sqrt{1/2}}$ for any $i \in \{1, 2, \dots, n\}$. The left edges are from Q to all nodes in the second layer. The influence probability on edge (Q, w) is $1/2$ and all other probabilities on edges sent from Q is 1. The influence cooperative game defined on G is $\Gamma(G) = (V = V_1 \cup V_2 \cup V_3, \sigma^{\text{LC}}(\cdot), \eta = 2n + m + 1/2)$. For convenient, we set $N = 2n + m$.

Under the above construction, suppose F is satisfiable and the corresponding assignment is $\{y_1, y_2, \dots, y_n\}$. Let $A = \{S_i : y_i = 1, i \in \{1, 2, \dots, n\}\} \cup \{T_i : y_i = 0, i \in \{1, 2, \dots, n\}\}$, $B = \{S_i : y_i = 0, i \in \{1, 2, \dots, n\}\} \cup \{T_i : y_i = 1, i \in \{1, 2, \dots, n\}\}$. Thus, A can active all nodes in the second layer except w and B can active all nodes in $\{x_1, x_2, \dots, x_n\} \cup \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$. Therefore, there are three disjoint successful coalitions $A' = A \cup \{v_1, v_2, \dots, v_n\}$, $B' = B \cup \{u_1, u_2, \dots, u_{n+1}\}$ and Q which means $\sigma^{\text{LC}}(A') \geq \eta$, $\sigma^{\text{LC}}(B') \geq \eta$ and $\sigma^{\text{LC}}(Q) \geq \eta$. Suppose r^* is the optimal solution of the relative least-core value of $\Gamma(G)$ and x^* is an optimal allocation. We can prove $r^* \geq 1 - \frac{1}{3}(N + \frac{7}{8})/(N + \frac{1}{2})$. This conclusion can be derived by separately considering cases $x^*(Q) > \frac{1}{3}(N + \frac{7}{8})$, $x^*(Q) < \frac{1}{3}(N + \frac{7}{8})$ and $x^*(Q) = \frac{1}{3}(N + \frac{7}{8})$.

When F is un-satisfiable, it is sufficient for our proof if we can find a solution (x, r) such that $r < 1 - \frac{\frac{1}{3}(N + \frac{7}{8})}{N + \frac{1}{2}}$. Note that when F is un-satisfiable, then for any $S \subseteq V$, $|S| \geq 2n + 1$ if $\sigma(S) \geq \eta$. Otherwise, we can construct an assignment such that F is satisfiable. Let $x(Q) = \frac{1}{3}(N + \frac{7}{8}) + \alpha$ and for any $v \in V$, $x(v) = \frac{\frac{2}{3}(N + \frac{7}{8}) - \alpha}{4n + 1}$.

The left is to prove that there exists a positive α such that $\frac{x(Q)}{\sigma^{\text{LC}}(Q)} \leq \frac{x(S)}{\sigma^{\text{LC}}(S)}$ for any S satisfying $\sigma^{\text{LC}}(S) \geq \eta$. We prove the above inequality by considering the following two cases:

(i) $Q \in S$: In this case,

$$\begin{aligned} \frac{x(S)}{\sigma^{\text{LC}}(S)} &\geq \frac{x(Q) + x(S \setminus \{Q\})}{\sigma^{\text{LC}}(\{Q\}) + 1} \\ &\geq \min\left\{\frac{x(Q)}{\sigma^{\text{LC}}(\{Q\})}, x(S \setminus \{Q\})\right\} \\ &\geq \min\left\{\frac{\frac{1}{3}(N + \frac{7}{8}) + \alpha}{N + \frac{1}{2}}, \frac{\frac{2}{3}(N + \frac{7}{8}) - \alpha}{4n + 1}\right\}. \end{aligned}$$

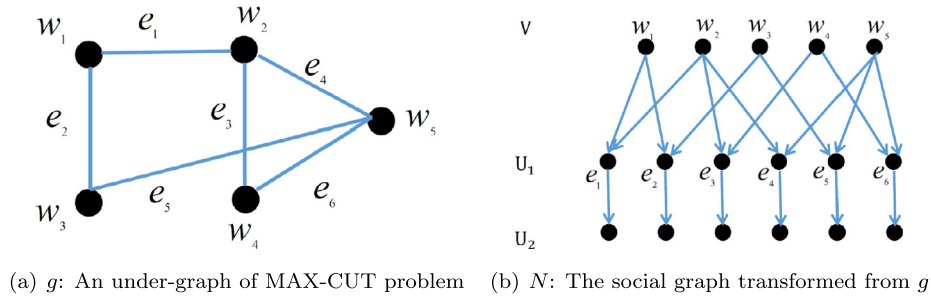


Fig. 3. Constructing a social graph from a MAX-CUT instance.

There exists $\alpha > 0$ such that $\frac{\frac{1}{3}(N+\frac{7}{8})+\alpha}{N+\frac{1}{2}} < \frac{\frac{2}{3}(N+\frac{7}{8})-\alpha}{4n+1}$ since $\frac{\frac{1}{3}(N+\frac{7}{8})}{N+\frac{1}{2}} < \frac{\frac{2}{3}(N+\frac{7}{8})}{4n+1}$. Thus,

$$\frac{x(S)}{\sigma^{\text{IC}}(S)} \geq \frac{x(Q)}{\sigma^{\text{IC}}(\{Q\})}.$$

(ii) $Q \notin S$: In this case, there exists an $\alpha > 0$ such that

$$\begin{aligned} \frac{x(S)}{\sigma^{\text{IC}}(S)} &= \frac{|S|[\frac{2}{3}(N+\frac{7}{8})-\alpha]}{(4n+1)(N+\frac{3}{4})} \\ &\geq \frac{(2n+1)[\frac{2}{3}(N+\frac{7}{8})-\alpha]}{(4n+1)(N+\frac{3}{4})} \\ &> \frac{[\frac{1}{3}(N+\frac{7}{8})+\alpha/2]}{N+\frac{1}{2}} \\ &= \frac{x(Q)}{\sigma^{\text{IC}}(Q)}. \quad \square \end{aligned}$$

Appendix C. Appendix of Section 5

Proof of Theorem 5. We construct a reduction from MAX-CUT problem. The input of MAX-CUT is an undirected graph $g = (W, E)$, where W is the node set and E is the edge set. The question is to compute the value of the maximum size of all edge cuts in g . Given an instance of MAX-CUT problem A_1 , we construct an instance of influence cooperative game A_2 as follows:

The social graph is a one-way layer graph $N = (V \cup U_1 \cup U_2, D)$, where each node in V corresponds to a node in W one-to-one and each node in U_1 corresponds to an edge in E one-to-one. There is a directed edge from $v \in V$ to $u \in U_1$ if and only if the node in W corresponding to v is one of the endpoints of the edge in E corresponding to u . U_2 is a copy of U_1 and there is a directed edge from each node in U_1 to its copy node in U_2 . The probabilities on all edges equal 1. Fig. 3 shows an example of the above construction.

The influence cooperative game defined on N is $A_2 = (V, \sigma^{\text{IC}}(\cdot), \eta = 0)$. Thus, $\mathcal{ALCV}(A_2)$ equals the optimal solution of the following linear programming:

$$\begin{aligned} \min \quad & \varepsilon \\ \text{s.t.} \quad & \begin{cases} x(V) = \sigma^{\text{IC}}(V) \\ x(S) \geq \sigma^{\text{IC}}(S) - \varepsilon & \forall S \subseteq V \\ x(\{u\}) \geq 0 & \forall u \in V \end{cases} \end{aligned} \quad (\text{C.1})$$

In $g = (W, E)$, given any subset $S \subseteq W$, let $\mathcal{I}(S) = |\{(i, j) \in E : i \in S, j \in S\}|$ and $\mathcal{C}(S) = |\{(i, j) \in E : i \in S, j \in W \setminus S\}|$. Clearly, $\mathcal{I}(S)$ is the number of edges induced by S and $\mathcal{C}(S)$ is the size of the cut between S and $W \setminus S$. To prove the hardness of computing $\mathcal{ALCV}(A_2)$, it is sufficient to prove $\mathcal{ALCV}(A_2) = c^*$, where c^* is the size of the maximum cut of g . Let (x^*, ε^*) be the optimal solution of (C.1). Therefore,

$$\begin{cases} x^*(S) \geq \sigma^{\text{IC}}(S) - \varepsilon^*, \forall S \subseteq V \\ x^*(V \setminus S) \geq \sigma^{\text{IC}}(V \setminus S) - \varepsilon^*, \forall S \subseteq V \end{cases}$$

Summing up these two inequalities, we have, $2\varepsilon^* \geq \sigma^{\text{IC}}(S) + \sigma^{\text{IC}}(V \setminus S) - x^*(V) = 2\mathcal{I}(S) + 2\mathcal{C}(S) + 2\mathcal{I}(V \setminus S) + 2\mathcal{C}(V \setminus S) - \sigma^{\text{IC}}(V) = 2\mathcal{I}(V) + 2\mathcal{C}(V \setminus S) - \sigma^{\text{IC}}(V) = 2\mathcal{C}(V \setminus S) = 2\mathcal{C}(S)$. Then $\varepsilon^* \geq c^*$ since $\varepsilon^* \geq \mathcal{C}(S)$ for any $S \subseteq V$.

Let $x = (x_1, x_2, \dots, x_n)$ and x_i be the degree of i for any $i \in W$. It is obvious that $x(V) = 2|E| = \sigma^{\text{LC}}(V)$. For any coalition $S \subseteq V$, $\sigma^{\text{LC}}(S) = 2(\mathcal{I}(S) + \mathcal{C}(S)) = (2\mathcal{I}(S) + \mathcal{C}(S)) + \mathcal{C}(S) = x(S) + \mathcal{C}(S) \leq x(S) + c^*$. That is to say (x, c^*) is a feasible solution of (C.1). Thus, $c^* \geq \varepsilon^*$. Combining with $c^* \leq \varepsilon^*$, we have $c^* = \varepsilon^*$.

If there is a ρ -approximate ($\rho \geq 1$) algorithm \mathcal{A} for computing $\mathcal{ALCV}(A_2)$ and \mathcal{A} outputs ε . Therefore, $\varepsilon^* \leq \varepsilon \leq \rho \varepsilon^*$. Which means $c^* \leq \varepsilon \leq \rho c^*$. Thus, ε/ρ is a $(1/\rho)$ -approximate value of c^* .

If the unique games conjecture is true, [26] proved that MAX-CUT cannot be approximated within $\alpha = \frac{2}{\pi} \min_{0 \leq \theta \leq \pi} \frac{\theta}{1 - \cos \theta}$ ($\alpha \approx 0.878$). Thus, ALCV problem of influence cooperative game can not be approximated within $\frac{1}{\alpha} \approx 1.139$. \square

Proof of Lemma 2. Suppose the optimal solutions of LP-PRIME, LP-RE and LP-STR are (x_p^*, ε_p^*) , (x_r^*, ε_r^*) and (x_s^*, ε_s^*) , respectively.

We first prove inequality (7). It obvious that (x_p^*, ε_p^*) is a feasible solution of LP-RE. Thus, $\varepsilon_r^* \leq \varepsilon_p^*$. In LP-RE, for any successful coalition S , we have $x_r^*(S) \geq \eta - \varepsilon_r^* \geq \sigma(S) - (\varepsilon_r^* + \sigma(V) - \eta)$. Therefore, $(x_r^*, \varepsilon_r^* + \sigma(V) - \eta)$ is a feasible solution of LP-PRIME. Thus, $\varepsilon_p^* \leq \varepsilon_r^* + \sigma(V) - \eta$ and then (7) sets up. Now we prove (8). Similar to the proof of inequality (7), (x_s^*, ε_s^*) is a feasible solution of LP-PRIME. Thus, $\varepsilon_p^* \leq \varepsilon_s^*$. In LP-PRIME, for any successful coalition S , we have $x_p^*(S) \geq \sigma(S) - \varepsilon_p^*$. For any unsuccessful coalition S' , we have $\sigma(S') - x_p^*(S') < \eta$. Thus, $(x_p^*(S), \max(\varepsilon_p^*, \eta))$ is a feasible solution of LP-STR. Thus, $\varepsilon_s^* \leq \varepsilon_s^* \leq \max(\varepsilon_p^*, \eta)$. \square

Proof of Lemma 3. The formula LP-STR captures cooperative games with submodular profit function. [15] proposed a framework to approximate the least-core value of this kind of cooperative games.

They defined an optimization problem names as *x-maximum dissatisfaction problem for cooperative game* (V, σ) (x-MD), where V is the player set and σ is the submodular profit function. The definition of x-MD is: Given any allocation x such that $x(V) = \sigma(V)$, find a coalition S^* whose dissatisfaction is maximum. i.e. $\max_{S \subseteq V} \{\sigma(S) - x(S)\}$. Under their framework, a ρ -approximation algorithm of x-MD implies a $1/\rho$ -approximation algorithm of the least-core value of the cooperative game (N, σ) . Moreover, ones can find an allocation in this ρ -approximation least-core.

Note that, given an allocation $x(V) = \sigma(V)$, finding the maximum value of $\sigma(S) - x(S)$ falls into the *submodular function maximization problem* since $\sigma(S) - x(S)$ is submodular. In [27], the authors design a deterministic $1/3$ -approximate algorithm of submodular function maximization problem when the function value on \emptyset and V is nonnegative. That is to say, there exists an $1/3$ -approximation algorithm of LP-STR since $\sigma(\emptyset) - x(\emptyset) = \sigma(V) - x(V) = 0$. \square

Proof of Lemma 4. The dynamic scheme is a generalization of the process in [28] In LP-RE, let $\alpha = \eta - \varepsilon$, then LP-RE can be transformed to the following linear programming (LP-RELAX-NEW):

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & \begin{cases} x(V) = \sigma(V) \\ x(\{S\}) \geq \alpha & \forall S \subseteq V, \sigma(S) \geq \eta \\ x(\{u\}) \geq 0 & \forall u \in V \\ 0 \leq \alpha \leq \eta \end{cases} \end{aligned} \quad (\text{C.2})$$

We break LP-RELAX-NEW into a family of linear feasibility programs $\mathcal{F} = \{LFP_1, LFP_2, \dots, LFP_t\}$, where $t = \lceil \frac{\eta}{\delta} \rceil$. The k -th linear feasibility program LFP_k is:

$$\begin{cases} x(V) = \sigma(V) \\ x(\{S\}) \geq k\delta & \forall S \subseteq V, \sigma(S) \geq \eta \\ x(\{u\}) \geq 0 & \forall u \in V \end{cases}$$

Let $k^* = \max\{k : LFP_k \text{ has a feasible solution}\}$ and α^* be the optimal value of LP-RELAX-NEW. Thus, $k^*\delta \leq \alpha^* < (k^* + 1)\delta$.

To solve these linear feasibility programs, we need a polynomial separation oracle since there are exponential constraints in each program. However, we do not know how to construct such an oracle. Following the idea in [28], we also construct a “partial” separation oracle \mathcal{O}_k for each program LFP_k , $k = 1, \dots, t$. \mathcal{O}_k has the following property, given a candidate solution x to LFP_k , the output of \mathcal{O}_k falls into one of the following cases:

1. \mathcal{O}_k successes, i.e. it can assert that x is a feasible solution for LFP_k or output a violate constraint in LFP_k .
2. \mathcal{O}_k fails, then it outputs a feasible solution for LFP_{k-1} .

Now we introduce how to use this partial separation oracle. Suppose we run the ellipsoid algorithm for each LFP_k , $k = 1, \dots, t$, using \mathcal{O}_k instead of a proper separation oracle. If \mathcal{O}_k successes, we can obtain a feasible solution to LFP_k or assert that $k^* = k - 1$. If \mathcal{O}_k fails, we can obtain a feasible solution to LFP_{k-1} . Thus, when we work to $k = k^*$, we can obtain a feasible solution of LFP_{k^*} or LFP_{k^*-1} . Let k' be the largest value of k for which our procedure finds a feasible solution for LFP_k . It holds that $k^* - 1 \leq k' \leq k^*$. Now, it is not difficult to derive that $\varepsilon_r^* \leq \eta - k'\delta \leq \varepsilon_r^* + 2\delta$.

A crucial problem is how to design \mathcal{O}_k . Indeed the main idea of \mathcal{O}_k is to apply dynamic programming such that we can decide whether there exists a coalition S with $\sigma(S) \geq \eta$ but $x(S) < k\delta$ under the given x . That is to say, given $x(V) = \sigma(V)$,

we want to compute $\max\{\sigma(S) | x(S) < k\delta\}$. To use dynamic programming, we need to discretized $x(S)$. For each $i \in V$, let $x'_i = \max\{j\delta' \leq x_i\}$, where $\delta' = \frac{\delta}{M}$. Now, for each $j \in \{1, \dots, n\}$ and $l \in \mathcal{L}\{0, \dots, (k-1)M-1\}$, let $z[j, l] = \max\{\sigma(S) | S \subseteq \{1, \dots, j\}, x'(S) = l\delta'\}$. For each $l \in \mathcal{L}$, we initialize $z[j, l]$ as:

$$z[1, l] = \begin{cases} \sigma(\{1\}) & l = \lfloor \frac{x'_1}{\delta'} \rfloor \\ -\infty & \text{otherwise} \end{cases}$$

The iteration rule is:

$$z[j, l] = \max\{z[j-1, l], z[j-1, \frac{l\delta' - x'_j}{\delta'}] + \sigma(\{1, \dots, j\}) - \sigma(\{1, \dots, j-1\})\}.$$

Let $U = \max\{z[n, i], i \in \mathcal{L}\}$, then we can find a violated constraints if $U \geq \eta$. Otherwise, if $U \leq \eta$, then for any $S \in V$ with $\sigma(S) \geq \eta$, we have $x'(S) \geq (k-1)n\delta' - \delta'$, thus, $x(S) \geq (k-1)\delta$. That is to say, x is a feasible solution to LFP_{k-1} . The running time of \mathcal{O}_k is polynomial time in $\log \sigma(V)$, n and $1/\delta$. \square

Appendix D. Appendix of Section 6

Proof of Lemma 5.

$$\begin{aligned} & tF(x^{(1)}) + (1-t)F(x^{(2)}) \\ &= \frac{1}{2^n} \sum_{S \subseteq V} (\max(t(f(S) - x^{(1)}(S)), 0) + \\ & \quad \max((1-t)(f(S) - x^{(2)}(S)), 0)) \\ &\geq \frac{1}{2^n} \sum_{S \subseteq V} \max(t(f(S) - x^{(1)}(S)) + (1-t)(f(S) - x^{(2)}(S)), 0) \\ &= F(tx^{(1)} + (1-t)x^{(2)}). \quad \square \end{aligned}$$

References

- [1] P. Domingos, M. Richardson, Mining the network value of customers, in: KDD, ACM, 2001, pp. 57–66.
- [2] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: KDD, ACM, 2003, pp. 137–146.
- [3] W. Chen, L.V.S. Lakshmanan, C. Castillo, Information and Influence Propagation in Social Networks, Morgan & Claypool Publishers, 2013.
- [4] G. Demange, On group stability in hierarchies and networks, J. Polit. Econ. 112 (2004) 754–778.
- [5] R. Meir, J.S. Rosenschein, E. Malizia, Subsidies, stability, and restricted cooperation in coalitional games, in: IJCAI, 2011, pp. 301–306.
- [6] E. Mossel, S. Roch, Submodularity of influence in social networks: from local to global, SIAM J. Comput. 39 (6) (2010) 2176–2188.
- [7] D. Gillies, Some theorems on n-person games, Ph.D. thesis, Princeton University, 1953.
- [8] L.S. Shapley, Markets as cooperative games, in: IJCAIR and Corporation Memorandum, 1955.
- [9] U. Faigle, W. Kern, On some approximately balanced combinatorial cooperative games, Math. Methods Oper. Res. 38 (1993) 141–152.
- [10] M. Maschler, B. Peleg, L.S. Shapley, Geometric properties of the kernel, nucleolus, and related solution concepts, Math. Oper. Res. 4 (4) (1979) 303–338.
- [11] G. Chalkiadakis, E. Elkind, M. Wooldridge, Computational aspects of cooperative game theory, Synth. Lect. Artif. Intell. Mach. Learn. 5 (6) (2011) 1–168.
- [12] X. Deng, C.H. Papadimitriou, On the complexity of cooperative solution concepts, Math. Oper. Res. 19 (2) (1994) 257–266.
- [13] S. Ieong, Y. Shoham, Marginal contribution nets: a compact representation scheme for coalitional games, in: Proceedings of the 6th ACM Conference on Electronic Commerce, ACM, 2005, pp. 193–202.
- [14] V. Conitzer, T. Sandholm, Complexity of constructing solutions in the core based on synergies among coalitions, Artif. Intell. 170 (6–7) (2006) 607–619.
- [15] A.S. Schulz, N.A. Uhan, Approximating the least core value and least core of cooperative games with supermodular costs, Discrete Optim. 10 (2) (2013) 163–180.
- [16] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: KDD, ACM, 2009, pp. 199–208.
- [17] A. Goyal, F. Bonchi, L.V. Lakshmanan, S. Venkatasubramanian, On minimizing budget and time in influence propagation over social networks, Soc. Netw. Anal. Min. (2012) 1–14.
- [18] Y. Tang, Y. Shi, X. Xiao, Influence maximization in near-linear time: a martingale approach, in: SIGMOD, ACM, 2015, pp. 1539–1554.
- [19] X. Shan, W. Chen, Q. Li, X. Sun, J. Zhang, Cumulative activation in social networks, Sci. China Inf. Sci. 62 (2019).
- [20] Z. Yang, S. He, J. Chen, GotU: leverage social ties for efficient user localization, Sci. China Inf. Sci. 63 (2019).
- [21] M. Granovetter, Threshold models of collective behavior, Am. J. Sociol. (1978) 489–515.
- [22] A.C. Bemmaor, Testing alternative econometric models on the existence of advertising threshold effect, J. Mark. Res. 21 (3) (1984) 298–308.
- [23] M. Grötschel, L. Lovász, A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Vol. 2, Springer Science & Business Media, 2012.
- [24] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.
- [25] R.G. Michael, S.J. David, Computers and Intractability: A Guide to the Theory of np-Completeness, WH Free. Co., San Fr., 1979, pp. 90–91.
- [26] S. Khot, G. Kindler, E. Mossel, R. O'Donnell, Optimal inapproximability results for max-cut and other 2-variable csps?, SIAM J. Comput. 37 (1) (2007) 319–357.
- [27] N. Buchbinder, M. Feldman, J. Seffi, R. Schwartz, A tight linear time (1/2)-approximation for unconstrained submodular maximization, SIAM J. Comput. 44 (5) (2015) 1384–1402.
- [28] E. Elkind, L.A. Goldberg, P. Goldberg, M. Wooldridge, Computational complexity of weighted threshold games, in: Proceedings of the National Conference on Artificial Intelligence, vol. 22, AAAI Press/MIT Press, Menlo Park, CA/Cambridge, MA/London, 1999, 2007, p. 718.