


Lipschitz Continuous Allocations for Optimization Games

Soh Kumabe 

CyberAgent, Tokyo, Japan

Yuichi Yoshida  

National Institute of Informatics, Tokyo, Japan

Abstract

In cooperative game theory, the primary focus is the equitable allocation of payoffs or costs among agents. However, in the practical applications of cooperative games, accurately representing games is challenging. In such cases, using an allocation method sensitive to small perturbations in the game can lead to various problems, including dissatisfaction among agents and the potential for manipulation by agents seeking to maximize their own benefits. Therefore, the allocation method must be robust against game perturbations.

In this study, we explore optimization games, in which the value of the characteristic function is provided as the optimal value of an optimization problem. To assess the robustness of the allocation methods, we use the Lipschitz constant, which quantifies the extent of change in the allocation vector in response to a unit perturbation in the weight vector of the underlying problem. Thereafter, we provide an algorithm for the matching game that returns an allocation belonging to the $(\frac{1}{2} - \epsilon)$ -approximate core with Lipschitz constant $O(\epsilon^{-1})$. Additionally, we provide an algorithm for a minimum spanning tree game that returns an allocation belonging to the 4-approximate core with a constant Lipschitz constant.

The Shapley value is a popular allocation that satisfies several desirable properties. Therefore, we investigate the robustness of the Shapley value. We demonstrate that the Lipschitz constant of the Shapley value for the minimum spanning tree is constant, whereas that for the matching game is $\Omega(\log n)$, where n denotes the number of vertices.

2012 ACM Subject Classification Mathematics of computing → Approximation algorithms

Keywords and phrases Cooperative Games, Lipschitz Continuity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.102

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: <https://arxiv.org/abs/2405.11889>

Funding *Soh Kumabe*: Supported by JSPS KAKENHI Grant Number JP21J20547.

Most of this work was done while the author was a student at The University of Tokyo.

Yuichi Yoshida: Supported by JSPS KAKENHI Grant Number JP24K02903.

1 Introduction

1.1 Background and Motivation

Cooperative games model decision-making scenarios in which multiple agents can achieve greater benefits through cooperation. A primary concern in cooperative game theory is the allocation of payoffs or costs provided by the grand coalition in an acceptable manner to each agent. Among the cooperative games, those defined by optimization problems corresponding to the set of agents involved are known as *optimization games* [8].

Consider the following well-known examples formulated by the matching game (MG) [9, 3] and the minimum spanning tree game (MSTG) [6]:



© Soh Kumabe and Yuichi Yoshida;

licensed under Creative Commons License CC-BY 4.0

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).

Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;

Article No. 102; pp. 102:1–102:16



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



MG: Members of a tennis club form pairs for a doubles tournament. Each pair of players has a predicted value for the prize money they would win if they teamed up. How should the total prize money won by all pairs be distributed among members?

MSTG: Multiple facilities cooperate to construct a power grid to receive electricity from a power plant. Each potential power line has a predetermined installation cost. When constructing a power grid that ensures electricity distribution to all facilities, what cost should each facility bear?

In these examples, the characteristic function of the game often contains errors, or can be manipulated through deliberate misreporting. For instance,

MG: It is challenging to accurately predict the compatibility of pairs who have never teamed up before. Moreover, pairs known to work well may hide this fact.

MSTG: Costs for power line installation might be misestimated owing to unforeseen terrain or geological conditions caused by natural disasters or inadequate surveys, land rights, or landscape regulations. Facilities may conceal these issues.

In such uncertain situations, allocations that drastically change with slight perturbations in the game can lead to problems, such as

MG: If minor estimation errors in predicted prize money significantly change the benefits for each player, players might not be satisfied with the allocation. In addition, when someone falsely reports a substantial increase in their gain, it becomes difficult to prove the intent of the manipulation if the degree of falsehood is small.

MSTG: Facilities might not accept the allocation if trivial issues in power line construction significantly increase their cost burden. Moreover, if such issues are used to significantly reduce their own costs or increase those of competing facilities, several minor construction issues may be concealed, leading to risk management problems.

To avoid these issues, it is desirable to use allocations that are robust against perturbations in real-world cooperative games.

Before proceeding, we define some terms in cooperative game theory. The *cooperative game* (V, ν) is defined as a pair consisting of a set of *agents* V and a *characteristic function* $\nu: 2^V \rightarrow \mathbb{R}_{\geq 0}$ representing the payoff or cost obtained when subsets of agents cooperate. An *optimization game* is defined by the optimization problem \mathcal{P} in a discrete structure. The \mathcal{P} -*game* (G, w) is defined from a pair of structures G consisting of *agent set* V , *edge set* E , and a weight vector $w \in \mathbb{R}_{\geq 0}^E$. In several games, G is a graph (V, E) . For each subset $S \subseteq V$, the characteristic function value $\nu(S)$ is defined as the optimum value of \mathcal{P} on the substructure corresponding to S (e.g., the subgraph induced by S) with respect to weight vector w . An optimization game is a *welfare allocation game* (resp., *cost allocation game*) if each agent intends to maximize (resp., minimize) the value allocated to it.

Now, we formally introduce matching games and minimum spanning tree games, as discussed in previous examples.

► **Definition 1** (Matching game [3, 8]). *Let $G = (V, E)$ be an undirected graph. For a vertex set $S \subseteq V$ and an edge weight vector $w \in \mathbb{R}_{\geq 0}^E$, let $\text{OPT}(S, w)$ denote the maximum matching weight in $G[S]$ with respect to weight w , where $G[S]$ represents the subgraph of G induced by S . The matching game of G with respect to weight w is defined as (V, ν) , where $\nu(S) = \text{OPT}(S, w)$. The matching game is a welfare-allocation game.*

► **Definition 2** (Minimum spanning tree game [4, 6]). *Let $G = (V \cup \{r\}, E)$ be an undirected graph. For a vertex set $S \subseteq V$ and edge weight vector $w \in \mathbb{R}_{\geq 0}^E$, let $\text{OPT}(S, w)$ denote the minimum weight of a spanning tree in $G[S \cup \{r\}]$ with respect to weight w . Notably, vertex r does not correspond to the agent. To ensure that the characteristic function has finite values, we assume that edge (r, v) exists for all $v \in V$. The minimum spanning tree game of G with respect to weight w is defined as (V, ν) , where $\nu(S) = \text{OPT}(S, w)$. The minimum spanning tree game is a cost-allocation game.*

Let us return to the discussion of the robustness of the allocation. To measure the robustness of the allocation, we introduce the concept of *Lipschitz continuity* in the allocation of these games, analogous to the introduction of Kumabe and Yoshida [23] for discrete optimization problems under the same name. Algorithm \mathcal{A} that takes a weight vector $w \in \mathbb{R}^E$ and returns an allocation $x \in \mathbb{R}_{\geq 0}^V$ is *L-Lipschitz* or has *Lipschitz constant* L if:

$$\sup_{\substack{w, w' \in \mathbb{R}_{\geq 0}^E \\ w \neq w'}} \frac{\|\mathcal{A}(w) - \mathcal{A}(w')\|_1}{\|w - w'\|_1} \leq L. \quad (1)$$

If the Lipschitz constant of the allocation method is small, the change in the allocation in response to a unit change in the weight vector is guaranteed to be small. Employing such an allocation method can resolve these problems as follows:

MG: Minor mistakes in estimating the compatibility of pairs will not significantly affect the overall distribution of prize money, making it easier for agents to accept the allocation. Additionally, substantial misreporting is necessary to increase benefits significantly through declaration adjustments, making schemes more likely to be exposed.

MSTG: Minor issues related to the installation of power lines will not cause significant fluctuations in the cost burden for each facility, making it easier for them to accept their share of the costs. Additionally, because the loss incurred by reporting such issues is small, the likelihood of these issues being concealed is reduced.

1.1.1 The core

The *core* is the most fundamental solution concept in cooperative game theory. An allocation vector $x \in \mathbb{R}^V$ is in the core of (welfare allocation) cooperative game (V, ν) if:

$$\sum_{v \in S} x_v \geq \nu(S) \quad (S \subsetneq V), \quad \sum_{v \in V} x_v = \nu(V). \quad (2)$$

Similarly, an allocation vector $x \in \mathbb{R}^V$ is in the core of (cost allocation) cooperative game (V, ν) if:

$$\sum_{v \in S} x_v \leq \nu(S) \quad (S \subsetneq V), \quad \sum_{v \in V} x_v = \nu(V). \quad (3)$$

Because the core is one of the most fundamental solution concepts, it is natural to desire a Lipschitz continuous algorithm \mathcal{A} that takes a weight vector as the input and returns an allocation belonging to the core. However, this can only be achieved in a limited number of situations for two reasons. First, a non-empty core does not necessarily exist in all games. Second, even for games in which the core exists in all instances, there is no guarantee that a vector from the core can be selected such that the Lipschitz constant of \mathcal{A} remains small.

To examine the second reason, we consider the *assignment game* introduced by Shapley and Shubik [29], which is a special case of the matching game in which the underlying graph is bipartite and known to have a non-empty core [8, 29].

► **Example 3.** Let n be an odd number greater than or equal to 5. Let us consider a path $G = (V, E)$ consisting of n vertices labeled sequentially as v_1, \dots, v_n . The weight vectors $w, w' \in \mathbb{R}_{\geq 0}^E$ are defined as follows:

$$w_{(v_i, v_{i+1})} = 1 \quad (i = 1, \dots, n-1), \quad w'_{(v_i, v_{i+1})} = \begin{cases} 1 & \text{if } 2 \leq i \leq n-2 \\ 0 & \text{otherwise} \end{cases}$$

In this case, the allocations x, x' belonging to the core of the assignment game for w and w' are both unique and obtained as follows:

$$x_i = \begin{cases} 1 & \text{if } i \text{ is even} \\ 0 & \text{otherwise} \end{cases} \quad x'_i = \begin{cases} 1 & \text{if } i \text{ is odd and } i \notin \{1, n\} \\ 0 & \text{otherwise} \end{cases}$$

If \mathcal{A} is an algorithm that takes a weight vector and returns a vector in the core of the game, then it must satisfy:

$$\frac{\|\mathcal{A}(w) - \mathcal{A}(w')\|_1}{\|w - w'\|_1} = \frac{n-2}{2} = \Omega(n)$$

Thus, algorithm \mathcal{A} must have a large Lipschitz constant, $\Omega(n)$.

Therefore, we consider providing a Lipschitz continuous algorithm that outputs allocations that satisfy some looser solution concepts. The least core [28] is one such solution concept that can be defined even for games in which a core does not exist. However, this does not resolve the problem of a large Lipschitz constant (for instance, in the game in Example 3, the core and the least core coincide, resulting in a Lipschitz constant of $\Omega(n)$). Instead, we consider the approximate core, which is a solution concept that multiplicatively relaxes the constraints of partial coalitions.

1.1.2 Approximate Core

The approximate core was introduced by Faigle and Kern [11] as a useful solution concept for games where the core is empty. Intuitively, the approximate core represents the core when the constraints for partial coalitions are relaxed by a factor α . For $\alpha \leq 1$, the allocation vector $x \in \mathbb{R}^V$ is in the α -(*approximate*) *core* of the (welfare allocation) cooperative game (V, ν) if:

$$\sum_{v \in S} x_v \geq \alpha \nu(S) \quad (S \subsetneq V), \quad \sum_{v \in V} x_v = \nu(V).$$

Similarly, for $\alpha \geq 1$, the allocation vector $x \in \mathbb{R}^V$ is in the α -(*approximate*) *core* of the (cost allocation) cooperative game (V, ν) if:

$$\sum_{v \in S} x_v \leq \alpha \nu(S) \quad (S \subsetneq V), \quad \sum_{v \in V} x_v = \nu(V).$$

In this study, we provide algorithms with small Lipschitz constants that return an α -approximate core for some constant α in several optimization games. For the matching game, we obtain the following:

► **Theorem 4.** *Let $\epsilon \in (0, \frac{1}{2}]$. For the matching game, there is a polynomial-time algorithm with a Lipschitz constant $O(\epsilon^{-1})$ that returns $(\frac{1}{2} - \epsilon)$ -approximate core allocation.*

Note that Faigle and Kern [11] and Vazirani [32] showed that the $\frac{2}{3}$ -approximate core of the matching game is non-empty, and that this is also a tight bound. However, their allocation, which is constructed using the optimal solution of the matching LP, is not Lipschitz continuous. Our allocation compromises the core approximability by $\frac{1}{6} + \epsilon$ to ensure Lipschitz continuity.

Our allocation was inspired by the Lipschitz continuous algorithm for the maximum weight-matching problem proposed by Kumabe and Yoshida [23]. Similarly, the proposed algorithm is based on the greedy method. We emphasize our results because their algorithm

reduces the approximation ratio to $\frac{1}{8} - \epsilon$ to obtain a Lipschitz constant of $O(\epsilon^{-1})$. Although the definitions of Lipschitz continuity and the approximation ratio for discrete algorithms differ from those for allocations, making a simple comparison infeasible, our analysis is simpler and offers better approximation guarantees.

For the minimum spanning tree game, we have the following.

► **Theorem 5.** *For the minimum spanning tree game, there is a polynomial-time algorithm with Lipschitz constant $O(1)$ that returns 4-approximate core allocation.*

The core of the minimum spanning tree game is non-empty; an allocation by Bird [4] is known to belong to the core and can be computed in polynomial time. However, it is not Lipschitz continuous. As in the matching game, our allocation compromises the core approximability to ensure Lipschitz continuity.

The original motivation for the approximate core was to provide useful solution concepts for games with non-empty cores. Therefore, no studies have been conducted on the approximate core of the minimum spanning tree game. In this regard, our setting demonstrates the usefulness of considering an approximate core, even for games with a non-empty core.

1.1.3 Shapley Value

Let \mathfrak{S}_V be the set of all permutations over V . For $\sigma \in \mathfrak{S}_V$ and $v \in V$, let $x_{\sigma,v} = \nu(\{\sigma(1), \dots, \sigma(k)\}) - \nu(\{\sigma(1), \dots, \sigma(k-1)\})$, where k is the integer with $\sigma(k) = v$. The *Shapley value* [27] of game (V, ν) is the vector s defined by:

$$s_v = \frac{1}{|V|!} \sum_{\sigma \in \mathfrak{S}_V} x_{\sigma,v}.$$

The Shapley value does not necessarily belong to the core, even if it exists, and the computation is #P-hard in most optimization games. However, they exhibit various desirable properties and have a wide range of applications [16]. Therefore, investigating the Lipschitz continuity of Shapley values is a natural task.

Considering its various properties, it is natural to expect the Shapley value to always have a small Lipschitz constant for general optimization games. Conversely, given the difficulty in computing it, it is natural to anticipate that it may not have a bounded Lipschitz constant. However, neither was accurate. Specifically, we demonstrate that whether the Shapley value in optimization games has a small Lipschitz constant depends on the game.

► **Theorem 6.** *There is a graph G such that the Shapley value of the matching game on G has a Lipschitz constant $\Omega(\log n)$.*

► **Theorem 7.** *The Shapley value of the minimum spanning tree game has a Lipschitz constant of 2.*

Notably, the computation of the Shapley value for the minimum spanning tree game is #P-hard [1]. The result of Theorem 7 is particularly interesting because it shows a value that is computationally difficult to calculate may still have a small Lipschitz constant.

1.2 Related Work

1.2.1 Optimization Games

The history of optimization games begins with the *assignment game* proposed by Shapley and Shubik [29]. They showed that the core of the assignment game is represented by the optimal solution of dual linear programming and is always non-empty. Deng, Ibaraki, and

Nagamochi [8] defined a class of games in which the characteristic function is represented by integer programming, and discussed the core structures of such games. In particular, for the matching game, they proved that the core is not always non-empty, but that the core non-emptiness problem, core membership problem, and that of generating a vector in the core if it is non-empty are solvable in polynomial time. Aziz and de Keijzer [2] proved that computing the Shapley value of the matching game is #P-hard. Additionally, studies on the generalization of matching games such as the *hypergraph matching game* [7, 8, 20] and *b-matching game* [5, 19, 30, 34], are being conducted. For a more detailed survey of the algorithmic aspects of matching games, see [3].

The minimum spanning tree game was proposed by Claus and Kleitman [6]. Bird [4] later proposed an allocation defined as follows: Regarding the minimum spanning tree of a given graph as a rooted tree rooted at r , each agent corresponding to a vertex v is allocated a cost equal to the weight of the edge from v to its parent. Granot and Huberman [14] proved this allocation is in the core. Unfortunately, it is not (Lipschitz) continuous. Faigle et al. [13] proved that the core membership problem for the minimum spanning tree game is coNP-hard. Ando [1] proved that the Shapley value of the minimum spanning tree game is #P-hard.

The concept of an approximate core was introduced by Faigle and Kern [11] as a useful solution for games where the core is empty. Their work discussed the approximate core allocations for several optimization games, including the matching game. In particular, for the matching game, they constructed a $\frac{2}{3}$ -core allocation based on LP relaxation. Subsequently, extensive studies have been conducted to determine the best α for which the α -core is always non-empty in various optimization games in which the core can be empty, such as the traveling salesman game [10, 11] and bin-packing games [11, 12, 17, 18, 26, 33]. Recently, following the rediscovery of Faigle and Kern's results by Vazirani [32], approximate cores have been derived for several optimization games such as the *b-matching game* [34] and the edge cover game [24].

Notably, in the definition by Faigle and Kern [11] and in subsequent papers, the right-hand side of Equations (2) or (3) being $(1 \pm \epsilon)\nu(S)$ is referred to as the ϵ -core. However, following the conventions in the field of discrete optimization and the study by Vazirani [32] and subsequent studies [24, 34], we adopt the definitions of Equations (2) and (3). An ϵ -core in our definition corresponds to a $|1 - \epsilon|$ -core in Faigle and Kern's definition.

1.2.2 Lipschitz Continuity of Discrete Algorithms

Inspired by recent studies on the average sensitivity [15, 21, 25, 31, 35] for *unweighted* discrete optimization problems, Kumabe and Yoshida [23] introduced the Lipschitz continuity of a randomized algorithm \mathcal{A} for *weighted* discrete optimization problems as follows:

$$\sup_{\substack{w, w' \in \mathbb{R}_{\geq 0}^V, \\ w \neq w'}} \frac{\min_{\mathcal{D} \in \Pi(\mathcal{A}(G, w), \mathcal{A}(G, w'))} \mathbf{E}_{(S, S') \sim \mathcal{D}} [d_w((S, w), (S', w'))]}{\|w - w'\|_1}, \quad (4)$$

where $\mathcal{A}(G, w)$ represents the output distribution of algorithm \mathcal{A} for input G with weight vector $w \in \mathbb{R}_{\geq 0}^V$, $\Pi(X, X')$ denotes the set of all joint distributions for random variables X and X' , and

$$d_w((S, w), (S', w')) = \left\| \sum_{v \in S} w_v \mathbf{1}_v - \sum_{v \in S'} w'_v \mathbf{1}_v \right\|_1 = \sum_{v \in S \cap S'} |w_v - w'_v| + \sum_{v \in S \setminus S'} w_v + \sum_{v \in S' \setminus S} w'_v.$$

They also proposed algorithms with small Lipschitz constants for the minimum spanning tree, shortest path, and maximum weight matching problems. In a subsequent study [22], the authors obtained algorithms with small Lipschitz constants for the minimum weight vertex cover, minimum weight set cover, and feedback vertex set problems.

In discrete optimization, the outputs of the algorithms are discrete sets. Thus, deterministic algorithms cannot be Lipschitz continuous; they consider randomized algorithms and adopt the earth mover's distance in the numerator of (4). In contrast, the outputs of our algorithms are allocations of continuous values, allowing deterministic algorithms to be Lipschitz continuous and randomized algorithms to be derandomized by taking the expectation. Therefore, in our setting, we can use a simple definition of the deterministic algorithms expressed in Equation (1).

1.3 Organization

The remainder of this paper is organized as follows: In Section 2, we provide several useful lemmas to analyze Lipschitz continuity of allocations. In Section 3, we prove Theorem 4 by providing a Lipschitz continuous polynomial-time algorithm that returns an approximate core allocation to the matching game. In Section 4, we prove Theorem 5 by providing a Lipschitz continuous polynomial-time algorithm that returns an approximate core allocation for the minimum spanning tree game. The proofs of Theorem 6 and Theorem 7 are given in the full version.

2 Basic Facts

In this section, we provide useful lemmas to analyze the core approximability and Lipschitz continuity of our allocations. The following lemma is useful for obtaining the Lipschitz constant: We omit this proof because it is similar to Lemma 1.7 of [23].

► **Lemma 8.** *Let $(G = (V, E), w)$ be an optimization game and \mathcal{A} be an algorithm that takes a pair (G, w) and outputs an allocation. Suppose that there exist some $c > 0$ and $L > 0$ such that*

$$\|(\mathcal{A}(G, w), \mathcal{A}(G, w + \delta \mathbf{1}_e))\|_1 \leq \delta L$$

holds for any $e \in E$, $w \in \mathbb{R}_{\geq 0}^E$, and $\delta > 0$ with either $\delta \leq c \cdot w_e$ or $w_e = 0$, where $\mathbf{1}_e$ represents the characteristic vector of e . Then, \mathcal{A} has a Lipschitz constant L .

In the analysis of the core approximability and Lipschitz continuity of allocations, we can simplify the discussion by ignoring the constraint $\sum_{v \in V} x_v = \nu(V)$ for a grand coalition. Therefore, we require a method to obtain an allocation with a bounded Lipschitz constant from vectors that satisfy only the constraints for partial coalitions with a bounded Lipschitz constant. To achieve this, we require a mild assumption in the game. An optimization game $(G = (V, E), w)$ is said to be *reasonable* if the inequality $|\nu(V, w) - \nu(V, w')| \leq \|w - w'\|_1$ holds for all $w, w' \in \mathbb{R}_{\geq 0}^E$. Notably, this is a fair assumption. For instance, games defined for optimization problems in the form of

$$\max \text{ or } \min \sum_{e \in X} w_e, \text{ subject to } X \in \mathcal{F}$$

for $\mathcal{F} \subseteq 2^E$ are all reasonable. The next lemma applies to welfare allocation games, such as the matching game. Due to the space limit, the proofs of the next two lemmas are given in the full version.

■ **Algorithm 1** Lipschitz continuous allocation for the matching game.

```

1 Procedure MATCHINGGAME( $G, w, b, \alpha$ )
  Input: A graph  $G = (V, E)$ , where the edge set is indexed with integers in
            $\{1, 2, \dots, |E|\}$ , a weight vector  $w \in \mathbb{R}_{\geq 0}^E$ ,  $b \in [0, 1]$ , and  $\alpha \in (1, 2]$ .
2   For each  $e \in E$  with  $w_e > 0$ , let  $\hat{w}_e \leftarrow \alpha^{i_e+1+b}$ , where  $i_e$  is the unique integer
           such that  $\alpha^{i_e+b} \leq w_e < \alpha^{i_e+1+b}$ ;
3    $z \leftarrow 0^V$ ,  $M \leftarrow \emptyset$ ;
4   for  $e \in E$  in descending order of  $w_e$ , where ties are broken according to their
           indices do
5     if none of the two endpoints of  $e$  are covered by  $M$  then
6       Add  $e$  to  $M$ ;
7        $z_v \leftarrow \hat{w}_e$  for each endpoint  $v$  of  $e$ ;
8   return  $z$ ;

```

► **Lemma 9.** Let $D \geq 1$. Let \mathcal{A} be an algorithm that takes a weight vector $w \in \mathbb{R}_{\geq 0}^E$ and returns an allocation $\mathcal{A}(w) \in \mathbb{R}_{\geq 0}^V$ for a reasonable welfare allocation game. Assume \mathcal{A} satisfies $\|\mathcal{A}(w)\|_1 \leq D\nu(V, w)$, $\sum_{v \in S} \mathcal{A}(w)_v \geq \nu(S, w)$ for all weight vector w and $S \subseteq V$, and $\|\mathcal{A}(w) - \mathcal{A}(w')\|_1 \leq L\|w - w'\|_1$ for all two weight vectors w and w' . Then, there is an algorithm that returns $\frac{1}{D}$ -approximate core allocation with Lipschitz constant $2L + 1$.

The next lemma applies to cost allocation games, such as the minimum spanning tree game.

► **Lemma 10.** Let $D \geq 1$. Let \mathcal{A} be an algorithm that takes a weight vector $w \in \mathbb{R}_{\geq 0}^E$ and returns an allocation $\mathcal{A}(w) \in \mathbb{R}_{\geq 0}^V$ for a reasonable cost allocation game. Assume \mathcal{A} satisfies $\|\mathcal{A}(w)\|_1 \geq \nu(V, w)$, $\sum_{v \in S} \mathcal{A}(w)_v \leq D\nu(S, w)$ for all weight vector w , and $\|\mathcal{A}(w) - \mathcal{A}(w')\|_1 \leq L\|w - w'\|_1$ for all two weight vectors w and w' . Then, there is an algorithm that returns D -approximate core allocation with Lipschitz constant $2L + 1$.

3 Matching Game

In this section, we prove Theorem 4 by giving a Lipschitz continuous algorithm that returns an approximate core allocation of the matching game. We obtain the proof by constructing an algorithm that satisfies the assumptions in Lemma 9. Specifically, we prove that an algorithm that returns a vector represented by

$$\int_0^1 \text{MATCHINGGAME}(G, w, b, \alpha) db \quad (5)$$

satisfies the assumptions of Lemma 9, where the procedure MATCHINGGAME is provided in Algorithm 1. We give a deterministic algorithm to compute this integral in Section 3.3.

MATCHINGGAME(G, w, b, α) first rounds each edge weight w_e to a value \hat{w}_e that is proportional to a power of α , where the proportionality constant is determined by b . Thereafter, it sorts the edges in descending order of \hat{w}_e and greedily selects them to form maximal matching M . Finally, for each edge $e \in M$, the algorithm allocates \hat{w}_e to both endpoints of the edges in M .

3.1 Core Approximability

The proofs of the following two lemmas for the core approximability analysis of Algorithm 1 are relatively straightforward.

► **Lemma 11.** *We have $\|z\|_1 \leq 2\alpha \text{OPT}(V, w)$.*

Proof. Because M is a matching of G , we have $\sum_{e \in M} w_e \leq \text{OPT}(V, w)$. Because the modified weight \hat{w}_e of each edge e in M contributes twice to $\|z\|_1$, we obtain

$$\|z\|_1 = 2 \sum_{e \in M} \hat{w}_e \leq 2\alpha \sum_{e \in M} w_e \leq 2\alpha \text{OPT}(V, w). \quad \blacktriangleleft$$

► **Lemma 12.** *Let $S \subseteq V$. Then, we have $\sum_{v \in S} z_v \geq \text{OPT}(S, w)$.*

Proof. Let $e = (u, v) \in E$. When edge e begins to be examined in the loop starting from Line 4, if at least one of u or v (say, u) is already covered by M , then we have $\hat{w}_e \leq z_u$. If neither u nor v are covered, then edge e is added to M , resulting in $\hat{w}_e = z_u = z_v$. Therefore, $\hat{w}_e \leq \max\{z_u, z_v\} \leq z_u + z_v$. Let M' be the maximum matching of $G[S]$. Then, we have

$$\sum_{v \in S} z_v \geq \sum_{(u,v) \in M'} (z_u + z_v) \geq \sum_{e \in M'} \hat{w}_e \geq \sum_{e \in M'} w_e = \text{OPT}(S, w). \quad \blacktriangleleft$$

3.2 Lipschitz Continuity

Let $G = (V, E)$ be a graph, $f \in E$, $\delta > 0$, $b \in [0, 1]$, and $\alpha > 1$. We will bound

$$\frac{1}{\delta} \int_0^1 \|\text{MATCHINGGAME}(G, w, b, \alpha) - \text{MATCHINGGAME}(G, w + \delta \mathbf{1}_f, b, \alpha)\|_1 db, \quad (6)$$

which is an upper bound on

$$\frac{1}{\delta} \left\| \int_0^1 \text{MATCHINGGAME}(G, w, b, \alpha) db - \int_0^1 \text{MATCHINGGAME}(G, w + \delta \mathbf{1}_f, b, \alpha) db \right\|_1.$$

From Lemma 8, bounding (6) for $\delta \leq w_f$ or $w_f = 0$ is sufficient to prove Lipschitz continuity. We denote the value of \hat{w} , M , and z in $\text{MATCHINGGAME}(G, w, b, \alpha)$ (resp., $\text{MATCHINGGAME}(G, w + \delta \mathbf{1}_f, b, \alpha)$) as \hat{w} , M , and z (resp., \hat{w}' , M' , and z').

When $\hat{w}_f = \hat{w}'_f$, $\text{MATCHINGGAME}(G, w, b, \alpha)$ and $\text{MATCHINGGAME}(G, w + \delta \mathbf{1}_f, b, \alpha)$ output the same vector. Assume otherwise. In $\text{MATCHINGGAME}(G, w, b)$, edge e_1 coming *before* edge e_2 refers to e_1 being considered before e_2 in the loop starting from Line 4, and is denoted as $e_1 \prec_w e_2$. In other words, either $\hat{w}_{e_1} > \hat{w}_{e_2}$ or $\hat{w}_{e_1} = \hat{w}_{e_2}$ and the index of e_1 comes earlier than that of e_2 . For $e_1 \neq f \neq e_2$, the relations $e_1 \prec_w e_2$ and $e_1 \prec_{w'} e_2$ are equivalent. Thus, we simply denote this as $e_1 \prec e_2$. The following lemma forms the core of our Lipschitzness analysis:

► **Lemma 13.** *Assume $\hat{w}_f \neq \hat{w}'_f$. Then, we have $\|z - z'\|_1 \leq 2\hat{w}'_f$.*

Proof. For each edge $e \neq f$ such that $e \in M' \setminus M$ (resp., $e \in M \setminus M'$), edge g is a *witness* of e if it is adjacent to e in M (resp., M') and $g \prec_w e$ (resp., $g \prec_{w'} e$). Intuitively, a witness of e is the edge that directly causes e to be excluded from M or M' .

From this definition, the witness of $e \in M' \setminus M$ belongs to $M \setminus M'$ and vice versa. Because \prec is an ordering on $E \setminus \{f\}$, by tracing the witnesses from any edge in $M \Delta M'$, we will consequently arrive at f . This implies that as long as $M \neq M'$, the edges in $M \Delta M'$

102:10 Lipschitz Continuous Allocations for Optimization Games

form a single path or cycle including f . Moreover, in this case, we have $f \in M' \setminus M$, because if we can add f to M in $\text{MATCHINGGAME}(G, w, b, \alpha)$, we could also add f to M' in $\text{MATCHINGGAME}(G, w + \delta \mathbf{1}_f, b, \alpha)$.

Let us now complete the proof. When $M = M'$, we have $\|z - z'\|_1 \leq 2(\widehat{w}'_f - \widehat{w}_f) \leq 2\widehat{w}'_f$. Otherwise, we let $f = (u_0, v_0)$. Then, there exists a unique maximal sequence of vertices (u_0, \dots, u_k) such that $(u_i, u_{i+1}) \in M \Delta M'$ for all i and $f \prec_{\widehat{w}} (u_0, u_1) \prec (u_1, u_2) \prec \dots \prec (u_{k-1}, u_k)$, and a unique maximal sequence of vertices (v_0, \dots, v_l) such that $(v_i, v_{i+1}) \in M \Delta M'$ for all i and $f \prec_{\widehat{w}'} (v_0, v_1) \prec (v_1, v_2) \prec \dots \prec (v_{l-1}, v_l)$ (if $M \Delta M'$ forms a cycle, then $(u_{k-1}, u_k) = (v_l, v_{l-1})$). Now, we have

$$\begin{aligned} \|z - z'\|_1 &\leq \sum_{i=0}^k |z_{u_i} - z'_{u_i}| + \sum_{i=0}^l |z_{v_i} - z'_{v_i}| \\ &= \left(|\widehat{w}'_f - \widehat{w}_{(u_0, u_1)}| + \sum_{i=1}^k |\widehat{w}_{(u_{i-1}, u_i)} - \widehat{w}_{(u_i, u_{i+1})}| \right) + \\ &\quad \left(|\widehat{w}'_f - \widehat{w}_{(v_0, v_1)}| + \sum_{i=1}^l |\widehat{w}_{(v_{i-1}, v_i)} - \widehat{w}_{(v_i, v_{i+1})}| \right) \\ &\leq 2\widehat{w}'_f, \end{aligned}$$

where the last inequality is from the fact that the sequences defined by $(\widehat{w}'_f, \widehat{w}_{(u_0, u_1)}, \dots, \widehat{w}_{(u_{k-1}, u_k)})$ and $(\widehat{w}'_f, \widehat{w}_{(v_0, v_1)}, \dots, \widehat{w}_{(v_{l-1}, v_l)})$ are both decreasing. \blacktriangleleft

The following lemma analyzes the probability that $\widehat{w}_f \neq \widehat{w}'_f$ happens.

► **Lemma 14.** *If b is sampled uniformly from $[0, 1]$, $\widehat{w}_f \neq \widehat{w}'_f$ happens with a probability of at most $\frac{\delta}{w_f \ln \alpha}$.*

Proof. $\widehat{w}_f \neq \widehat{w}'_f$ happens when there exists an integer i with $w_f < \alpha^{i+b} \leq w_f + \delta$, indicating that $\lceil \log_\alpha w_f - b \rceil \neq \lceil \log_\alpha w_f + \delta - b \rceil$. This happens with probability

$$\log_\alpha(w_f + \delta) - \log_\alpha w_f = \log_\alpha \left(1 + \frac{\delta}{w_f} \right) \leq \frac{1}{\ln \alpha} \cdot \frac{\delta}{w_f} = \frac{\delta}{w_f \ln \alpha}. \quad \blacktriangleleft$$

Now, we complete our Lipschitzness analysis.

► **Lemma 15.** *We have*

$$\int_0^1 \|\text{MATCHINGGAME}(G, w, b, \alpha) - \text{MATCHINGGAME}(G, w + \delta \mathbf{1}_f, b, \alpha)\|_1 db \leq \frac{12}{\alpha - 1} \delta.$$

Proof. If $w_f = 0$, we have

$$\begin{aligned} &\int_0^1 \|\text{MATCHINGGAME}(G, w, b, \alpha) - \text{MATCHINGGAME}(G, w + \delta \mathbf{1}_f, b, \alpha)\|_1 db \\ &\leq 2\widehat{w}'_f \leq 2\alpha(w_f + \delta) = 2\alpha\delta \leq \frac{12}{\alpha - 1} \delta, \end{aligned}$$

where the last inequality is from $\alpha \leq 2$. Otherwise, we have

$$\begin{aligned} &\int_0^1 \|\text{MATCHINGGAME}(G, w, b, \alpha) - \text{MATCHINGGAME}(G, w + \delta \mathbf{1}_f, b, \alpha)\|_1 db \\ &\leq \frac{\delta}{w_f \ln \alpha} \cdot 2\widehat{w}'_f \leq \frac{\delta}{w_f \ln \alpha} \cdot 2\alpha(w_f + \delta) \leq \frac{\delta}{w_f \ln \alpha} \cdot 4\alpha w_f = \frac{4\alpha\delta}{\ln \alpha} \leq \frac{12}{\alpha - 1} \delta, \end{aligned}$$

where the third inequality is from $\delta \leq w_f$ and the last inequality is from the fact that $\frac{\alpha}{\ln \alpha} \leq \frac{3}{\alpha - 1}$ holds for $\alpha \in (1, 2]$. \blacktriangleleft

3.3 Proof of Theorem 4

Combining Lemmas 11, 12, and 15 and applying Lemma 9 yields the following:

► **Lemma 16.** *Let $\epsilon \in (0, \frac{1}{2}]$. For the matching game, an algorithm that returns $(\frac{1}{2} - \epsilon)$ -approximate core allocation with Lipschitz constant $O(\epsilon^{-1})$ exists.*

Proof. Let $\alpha = 1 + 2\epsilon$. By combining Lemmas 11, 12, 15, and 9, we obtain an algorithm that returns $\frac{1}{2\alpha}$ -approximate core allocation for the matching game with Lipschitz constant $\frac{24}{\alpha-1} + 1$. As $\frac{1}{2(1+2\epsilon)} \geq \frac{1}{2} - \epsilon$ and $\frac{24}{2\epsilon} + 1 \leq O(\epsilon^{-1})$, this algorithm satisfies the claims of the lemma. ◀

Proof of Theorem 4. It is sufficient to prove that the allocation defined by Lemma 16 can be computed in polynomial time. For each edge $e \in E$, let $b_e = \log_\alpha w_e - \lfloor \log_\alpha w_e \rfloor$, and sort the b_e values in ascending order to obtain a sequence $t_1, \dots, t_{|E|}$. For convenience, we set $t_0 = 0$ and $t_{|E|+1} = 1$. For each $i = 0, \dots, |E|$, the behavior of Algorithm 1 for any $b \in [t_i, t_{i+1})$ is identical, except for the constant multiplier on \hat{w} . Therefore, by running Algorithm 1 for $b = t_i$ and appropriately scaling the result, and thereafter summing these results for each i , we can compute the integral in Equation (5) in polynomial time. ◀

4 Minimum Spanning Tree Game

In this section, we prove Theorem 5 by giving a Lipschitz continuous algorithm that returns an approximate core allocation for the minimum spanning tree game. The proof is obtained by constructing an algorithm that satisfies the assumption of Lemma 10. Specifically, we prove that an algorithm that returns a vector represented by

$$\int_0^1 \text{MSTGAME}(G, w, b) db$$

satisfies the assumptions of Lemma 10, where the procedure MSTGAME is provided in Algorithm 3. We give a deterministic algorithm to compute this integral in Section 4.3.

To derive our allocation, we use an *auxiliary tree* that simulates Kruskal's algorithm, constructed as in Algorithm 2. The auxiliary tree is a rooted tree such that each leaf corresponds to a vertex in $V \cup \{r\}$. We provide an overview of Algorithm 2. Initially, for each vertex $v \in V \cup \{r\}$, the algorithm prepares a vertex $u_{\{v\}}$ and sets its *height* $h_{u_{\{v\}}}$ to 0. The auxiliary tree is constructed by adding the edges of E in ascending order of weight to graph $(V \cup \{r\}, \emptyset)$. Edges of the same weight are added simultaneously. When adding the edges of a certain weight results in merging multiple connected components C_1, \dots, C_k into a single connected component C , the algorithm creates a vertex u_C corresponding to C in the auxiliary tree. The height of u_C is set as the weight of the edges at that time, and the edges are added to the auxiliary tree from u_C to u_{C_1}, \dots, u_{C_k} .

For $x \in \mathbb{R}_{\geq 0}$ and a weight vector \hat{w} , let $\mathcal{C}_{\hat{w}, < x}$ and $\mathcal{C}_{\hat{w}, \leq x}$ be the families of connected components of graphs whose vertex sets are $V \cup \{r\}$ and edge sets consist of edges $e \in E$ with $\hat{w}_e < x$ and $\hat{w}_e \leq x$, respectively. For the auxiliary tree T , we denote the subtree rooted at vertex u by T_u . When the edges of an auxiliary tree are referred to as (u, u') , u is the parent of u' . For an edge $e = (u, u')$, we denote $T_e = T_{u'}$. For simplicity, for $e = (u, u') \in E(T)$, we define $h_e := h_u$.

$\text{MSTGAME}(G, w, b)$ first rounds each edge weight w_e to a value \hat{w}_e that is proportional to a power of 2, where the proportionality constant is determined by b . Let T be the auxiliary tree derived from (G, \hat{w}) . Then, for each edge e in T such that T_e does not have r as a

■ **Algorithm 2** Construction of the auxiliary tree.

```

1 Procedure AUXILIARYTREE( $G, \hat{w}$ )
  Input: A graph  $G = (V \cup \{r\}, E)$  and a weight vector  $\hat{w} \in \mathbb{R}_{\geq 0}^E$ .
2   $U \leftarrow \{u_{\{v\}} \mid v \in V \cup \{r\}\}, F \leftarrow \emptyset;$ 
3   $h_u \leftarrow 0$  for each  $u \in U;$ 
4  for  $x \in \mathbb{R}_{\geq 0}$  such that  $\mathcal{C}_{\hat{w}, < x} \neq \mathcal{C}_{\hat{w}, \leq x}$ , in ascending order do
5    for  $C \in \mathcal{C}_{\hat{w}, \leq x} \setminus \mathcal{C}_{\hat{w}, < x}$  do
6      Add a new vertex  $u_C$  to  $U;$ 
7       $h_{u_C} \leftarrow x;$ 
8      for  $C' \in \mathcal{C}_{\hat{w}, < x}$  such that  $C' \subseteq C$  do
9        Add a new edge  $(u_C, u_{C'})$  to  $F;$ 
10 return  $(U, F, h);$ 

```

■ **Algorithm 3** Lipschitz continuous allocation for the minimum spanning tree game.

```

1 Procedure MSTGAME( $G, w, b$ )
  Input: A graph  $G = (V \cup \{r\}, E)$ , a weight vector  $w \in \mathbb{R}_{\geq 0}^E$ , and  $b \in [0, 1]$ .
2  For each  $e \in E$  with  $w_e > 0$ , let  $\hat{w}_e \leftarrow 2^{i_e+1+b}$ , where  $i_e$  be the unique integer
   such that  $2^{i_e+b} \leq w_e < 2^{i_e+1+b};$ 
3   $T \leftarrow$  AUXILIARYTREE( $G, \hat{w}$ ) and identify the vertices of  $G$  with the
   corresponding leaves of  $T;$ 
4  for  $e \in E(T)$  such that  $r \notin T_e$  do
5    Let  $X_e$  be the set of leaves in  $T_e;$ 
6    Let  $z_e \leftarrow \frac{h_e}{|X_e|} \mathbf{1}_{X_e};$ 
7  return  $\sum_{e \in E} z_e;$ 

```

leaf, the value h_e is evenly distributed among the agents corresponding to the leaves of T_e . At first glance, the total value distributed may seem unrelated to the value of the grand coalition. However, it can be proved in Lemmas 18 and 19 that the total value distributed by this method is at least $\text{OPT}(G, \hat{w})$ and at most $2\text{OPT}(G, \hat{w})$.

4.1 Core Approximability

We begin by analyzing the core approximability. Let T be the auxiliary tree for (G, \hat{w}) . For $X \subseteq V \cup \{r\}$, the *connector* $\text{conn}(X)$ of X is the minimal connected subgraph of T that contains all leaves of T corresponding to X . The following lemma bounds the value of the characteristic function for a subset S of V using values that can be computed from connector $\text{conn}(S \cup \{r\})$. Due to the space limit, proofs in this section are given in the full version.

► **Lemma 17.** *Let $S \subseteq V$. Then, we have*

$$\sum_{\substack{(u, u') \in E(\text{conn}(S \cup \{r\})) \\ r \notin T_{u'}}} (h_u - h_{u'}) \leq \text{OPT}(G[S \cup \{r\}], \hat{w}).$$

When $V = S$, the equality holds.

Now, we have the following.

► **Lemma 18.** *Let $S \subseteq V$. Then, we have*

$$\sum_{v \in S} \sum_{e \in E(T)} z_{e,v} \leq 4\text{OPT}(G[S \cup \{r\}], w).$$

We have the following bound for the grand coalition.

► **Lemma 19.** *We have*

$$\sum_{v \in V} \sum_{e \in E(T)} z_{e,v} \geq \text{OPT}(G, w).$$

4.2 Lipschitz Continuity

Let $f \in E(G)$. We bound

$$\frac{1}{\delta} \int_0^1 \|\text{MSTGAME}(G, w, b) - \text{MSTGAME}(G, w + \delta \mathbf{1}_f, b)\|_1 db,$$

which is an upper bound on

$$\frac{1}{\delta} \left\| \int_0^1 \text{MSTGAME}(G, w, b) db - \int_0^1 \text{MSTGAME}(G, w + \delta \mathbf{1}_f, b) db \right\|_1.$$

Without loss of generality, we can assume $\delta \leq w_f$ or $w_f = 0$. Now we fix $b \in [0, 1]$. We denote the value of T , \widehat{w} , X , and z in $\text{MSTGAME}(G, w, b)$ (resp., $\text{MSTGAME}(G, w + \delta \mathbf{1}_f, b)$) as T , \widehat{w} , X , and z (resp. T' , \widehat{w}' , X' , and z').

When $\widehat{w}_f = \widehat{w}'_f$, T and T' are the same and $\text{MSTGAME}(G, w, b)$ and $\text{MSTGAME}(G, w + \delta \mathbf{1}_f, b)$ output the same vector. Otherwise, let C be the connected component of $\mathcal{C}_{\widehat{w}, \leq \widehat{w}_f}$ that contains both the endpoints of f . If C is still connected even after the removal of f , then we have $\mathcal{C}_{\widehat{w}, \leq x} = \mathcal{C}_{\widehat{w}', \leq x}$ for all $x \geq 0$ and thus $T = T'$. Otherwise, let C_1 and C_2 be two connected components of C after the removal of f . Subsequently, T' is obtained from T by the following operation:

- (1.1) If no vertex u_{C_1} exists in T , create a vertex u_{C_1} , set $h_{u_{C_1}} = \widehat{w}_f$, and for each child u_X of u_C with $X \subseteq C_1$, replace the edge (u_C, u_X) with (u_{C_1}, u_X) . Otherwise, delete the edge (u_C, u_{C_1}) .
- (1.2) Do exactly the same for C_2 .
- (2) If u_C has a parent u_Y with $h_{u_Y} = \widehat{w}'_f$, delete the vertex u_C and the edge (u_Y, u_C) , and add two new edges (u_Y, u_{C_1}) and (u_Y, u_{C_2}) . Otherwise, add two new edges (u_C, u_{C_1}) and (u_C, u_{C_2}) and then change the value of h_{u_C} from \widehat{w}_f to \widehat{w}'_f .

We can observe that all edges e of T except for the edges (u_C, u_{C_1}) deleted in (1.1), (u_C, u_{C_2}) deleted in (1.2), and (u_Y, u_C) deleted in (2) naturally correspond to edges in T' that are not added in (2), and if we identify the edges of T with those of T' using that correspondence, it holds that $h_e = h'_e$ and $X_e = X'_e$, which implies $z_e = z'_e$. Therefore, we have the following.

► **Lemma 20.** *Assume $\widehat{w}_f \neq \widehat{w}'_f$. Then, we have*

$$\|\text{MSTGAME}(G, w, b) - \text{MSTGAME}(G, w + \delta \mathbf{1}_f, b)\|_1 \leq \widehat{w}_f + 2\widehat{w}'_f.$$

Proof. Let

$$z_1 = \begin{cases} \frac{1}{|C_1|} \mathbf{1}_{C_1} & \text{if } (u_C, u_{C_1}) \text{ is deleted in (1.1) and } r \notin C_1 \\ \mathbf{0} & \text{otherwise} \end{cases},$$

102:14 Lipschitz Continuous Allocations for Optimization Games

$$\begin{aligned}
 z_2 &= \begin{cases} \frac{1}{|C_2|} \mathbf{1}_{C_2} & \text{if } (u_C, u_{C_2}) \text{ is deleted in (1.2) and } r \notin C_2 \\ \mathbf{0} & \text{otherwise} \end{cases}, \\
 z_3 &= \begin{cases} \frac{1}{|C|} \mathbf{1}_C & \text{if } (u_Y, u_C) \text{ is deleted in (1.2) and } r \notin C \\ \mathbf{0} & \text{otherwise} \end{cases}, \\
 z_4 &= \begin{cases} \frac{1}{|C_1|} \mathbf{1}_{C_1} & \text{if } r \notin C_1 \\ \mathbf{0} & \text{otherwise} \end{cases}, \\
 z_5 &= \begin{cases} \frac{1}{|C_2|} \mathbf{1}_{C_2} & \text{if } r \notin C_2 \\ \mathbf{0} & \text{otherwise} \end{cases}.
 \end{aligned}$$

Then, we have

$$\begin{aligned}
 & \|\text{MSTGAME}(G, w, b) - \text{MSTGAME}(G, w + \delta \mathbf{1}_f, b)\|_1 \\
 &= \|(z_1 + z_2) \widehat{w}_f + (z_3 - z_4 - z_5) \widehat{w}'_f\|_1 \\
 &= \|(z_1 \widehat{w}_f + (z_3 \circ \mathbf{1}_{C_1} - z_4) \widehat{w}'_f)\|_1 + \|(z_2 \widehat{w}_f + (z_3 \circ \mathbf{1}_{C_2} - z_5) \widehat{w}'_f)\|_1 \\
 &\leq \max\left(\widehat{w}_f + \frac{|C_1|}{|C|} \widehat{w}'_f, \widehat{w}'_f\right) + \max\left(\widehat{w}_f + \frac{|C_2|}{|C|} \widehat{w}'_f, \widehat{w}'_f\right) \leq \widehat{w}_f + 2\widehat{w}'_f. \quad \blacktriangleleft
 \end{aligned}$$

The following lemma analyzes the probability that $\widehat{w}_f \neq \widehat{w}'_f$ happens.

► **Lemma 21.** *Assume $w'_f \leq 2w_f$. If b is sampled uniformly from $[0, 1]$, $\widehat{w}_f \neq \widehat{w}'_f$ happens with a probability of at most $\frac{\delta}{w_f \log 2}$.*

Proof. $\widehat{w}_f \neq \widehat{w}'_f$ happens when there is an integer i with $w_f < 2^{i+b} \leq w_f + \delta$, implying that $\lfloor \log_2 w_f - b \rfloor \neq \lfloor \log_2 w_f + \delta - b \rfloor$. This happens with probability

$$\log_2(w_f + \delta) - \log_2 w_f = \log_2\left(1 + \frac{\delta}{w_f}\right) \leq \frac{\delta}{w_f \log 2}. \quad \blacktriangleleft$$

Now, we have the following:

► **Lemma 22.** *We have*

$$\frac{1}{\delta} \int_0^1 \|\text{MSTGAME}(G, w, b) - \text{MSTGAME}(G, w + \delta \mathbf{1}_f, b)\|_1 db \leq \frac{10\delta}{\log 2}.$$

Proof. If $w_f = 0$, we have

$$\frac{1}{\delta} \int_0^1 \|\text{MSTGAME}(G, w, b) - \text{MSTGAME}(G, w + \delta \mathbf{1}_f, b)\|_1 db \leq 2\widehat{w}'_f \leq 4w'_f = 4\delta.$$

Otherwise, we have

$$\begin{aligned}
 & \frac{1}{\delta} \int_0^1 \|\text{MSTGAME}(G, w, b) - \text{MSTGAME}(G, w + \delta \mathbf{1}_f, b)\|_1 db \\
 &\leq \frac{\delta}{w_f \log 2} (\widehat{w}_f + 2\widehat{w}'_f) = \frac{\delta}{w_f \log 2} \cdot 5\widehat{w}_f \leq \frac{\delta}{w_f \log 2} \cdot 10w_f = \frac{10\delta}{\log 2}. \quad \blacktriangleleft
 \end{aligned}$$

4.3 Proof of Theorem 5

Combining Lemmas 18, 19, and 22 and applying Lemma 10 yields Theorem 5.

Proof of Theorem 5. Combining Lemmas 18, 19, 22, and 10, we obtain an algorithm that returns 4-approximate core allocation for the matching game with Lipschitz constant $\frac{20}{\log 2} + 1 = O(1)$. The fact that the allocation defined by Lemma 16 can be computed in polynomial time is obtained using the same argument as in the proof of Theorem 4. \blacktriangleleft

References

- 1 Kazutoshi Ando. Computation of the shapley value of minimum cost spanning tree games: #P-hardness and polynomial cases. *Japan Journal of Industrial and Applied Mathematics*, 29(3):385–400, 2012.
- 2 Haris Aziz and Bart de Keijzer. Shapley meets shapley. *arXiv preprint*, 2013. [arXiv:1307.0332](https://arxiv.org/abs/1307.0332).
- 3 Márton Benedek, Péter Biró, Matthew Johnson, Daniël Paulusma, and Xin Ye. The complexity of matching games: A survey. *Journal of Artificial Intelligence Research*, 77:459–485, 2023.
- 4 Charles G Bird. On cost allocation for a spanning tree: a game theoretic approach. *Networks*, 6(4):335–350, 1976.
- 5 Péter Biró, Walter Kern, Daniël Paulusma, and Péter Wojuteczky. The stable fixtures problem with payments. *Games and economic behavior*, 108:245–268, 2018.
- 6 Armin Claus and Daniel J Kleitman. Cost allocation for a spanning tree. *Networks*, 3(4):289–304, 1973.
- 7 Vincent Conitzer and Tuomas Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6-7):607–619, 2006.
- 8 Xiaotie Deng, Toshihide Ibaraki, and Hiroshi Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24(3):751–766, 1999.
- 9 Kimmo Eriksson and Johan Karlander. Stable outcomes of the roommate game with transferable utility. *International Journal of Game Theory*, 29:555–569, 2001.
- 10 Ulrich Faigle, Sándor P Fekete, Winfried Hochstättler, and Walter Kern. On approximately fair cost allocation in euclidean tsp games. *Operations-Research-Spektrum*, 20:29–37, 1998.
- 11 Ulrich Faigle and Walter Kern. On some approximately balanced combinatorial cooperative games. *Zeitschrift für Operations Research*, 38:141–152, 1993.
- 12 Ulrich Faigle and Walter Kern. Approximate core allocation for binpacking games. *SIAM Journal on Discrete Mathematics*, 11(3):387–399, 1998.
- 13 Ulrich Faigle, Walter Kern, Sándor P Fekete, and Winfried Hochstättler. On the complexity of testing membership in the core of min-cost spanning tree games. *International Journal of Game Theory*, 26:361–366, 1997.
- 14 Daniel Granot and Gur Huberman. Minimum cost spanning tree games. *Mathematical programming*, 21:1–18, 1981.
- 15 Satoshi Hara and Yuichi Yoshida. Average sensitivity of decision tree learning. In *The 11th International Conference on Learning Representations (ICLR)*, 2023.
- 16 Sergiu Hart. Shapley value. In *Game theory*, pages 210–216. Springer, 1989.
- 17 Walter Kern and Xian Qiu. Integrality gap analysis for bin packing games. *Operations Research Letters*, 40(5):360–363, 2012.
- 18 Jeroen Kuipers. Bin packing games. *Mathematical Methods of Operations Research*, 47:499–510, 1998.
- 19 Soh Kumabe and Takanori Maehara. Convexity of b -matching games. In *International Joint Conference on Artificial Intelligence*, pages 261–267, 2020.
- 20 Soh Kumabe and Takanori Maehara. Convexity of hypergraph matching game. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 663–671, 2020.
- 21 Soh Kumabe and Yuichi Yoshida. Average sensitivity of dynamic programming. In *Proceedings of the 33th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1925–1961, 2022.
- 22 Soh Kumabe and Yuichi Yoshida. Lipschitz continuous algorithms for covering problems. *arXiv preprint*, 2023. [arXiv:2307.08213](https://arxiv.org/abs/2307.08213).
- 23 Soh Kumabe and Yuichi Yoshida. Lipschitz continuous algorithms for graph problems. In *Proceedings of the 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 762–797. IEEE, 2023.
- 24 Tianhang Lu, Han Xiao, and Qizhi Fang. Approximate core allocations for edge cover games. In *International Workshop on Frontiers in Algorithmics*, pages 105–115. Springer, 2023.

- 25 Pan Peng and Yuichi Yoshida. Average sensitivity of spectral clustering. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1132–1140, 2020.
- 26 Xian Qiu and Walter Kern. Approximate core allocations and integrality gap for the bin packing game. *Theoretical Computer Science*, 627:26–35, 2016.
- 27 Lloyd S Shapley. *On Balanced Sets and Cores*. RAND Corporation, 1965.
- 28 Lloyd S Shapley and Martin Shubik. Quasi-cores in a monetary economy with nonconvex preferences. *Econometrica: Journal of the Econometric Society*, pages 805–827, 1966.
- 29 Lloyd S Shapley and Martin Shubik. The assignment game I: The core. *International Journal of Game Theory*, 1:111–130, 1971.
- 30 Marilda Sotomayor. The multiple partners game. In *Equilibrium and Dynamics: Essays in Honour of David Gale*, pages 322–354. Springer, 1992.
- 31 Nithin Varma and Yuichi Yoshida. Average sensitivity of graph algorithms. *SIAM Journal on Computing*, 52(4):1039–1081, 2023.
- 32 Vijay V Vazirani. The general graph matching game: Approximate core. *Games and Economic Behavior*, 132:478–486, 2022.
- 33 Gerhard J Woeginger. On the rate of taxation in a cooperative bin packing game. *Zeitschrift für Operations Research*, 42:313–324, 1995.
- 34 Han Xiao, Tianhang Lu, and Qizhi Fang. Approximate core allocations for multiple partners matching games. *arXiv preprint*, 2021. [arXiv:2107.01442](https://arxiv.org/abs/2107.01442).
- 35 Yuichi Yoshida and Shinji Ito. Average sensitivity of Euclidean k -clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.