Convexity of b-matching Games

Soh Kumabe[1,2] and **Takanori Maehara**[2]
[1]The University of Tokyo
[2]RIKEN AIP
soh_kumabe@mist.i.u-tokyo.ac.jp, takanori.maehara@riken.jp

Abstract

The b-matching game is a cooperative game defined on a graph, which generalizes the matching game to allow each individual to have more than one partner. The game has several applications, such as the roommate assignment, the multi-item version of the seller-buyer assignment, and the international kidney exchange. In contrast to the matching game, the b-matching game is computationally hard, i.e., the core non-emptiness problem and the core membership problem are co-NP-hard. Therefore, we focus on the convexity of the game, which is a sufficient condition of the core non-emptiness, often more tractable than the core non-emptiness, and has several additional benefits. In this study, we give a necessary and sufficient condition of the convexity of the b-matching game, which yields an $O(n \log n + m\alpha(n))$ time algorithm to determine whether a given game is convex or not, where n and m are the number of vertices and edges, respectively, and α is the inverse-Ackermann function. Using our characterization, we also give a polynomial-time algorithm to compute the Shapley value of a convex b-matching game.

1 Introduction

1.1 Background and Motivation

Finding a "reasonable" matching is a fundamental research topic in algorithmic cooperative game theory [Deng and Fang, 2008]. After the seminal work of Shapley and Shubik 1971, the matching game and its variants have been extensively studied.

In this study, we consider the *b-matching game* [Deng *et al.*, 1999], which is also known as *multi-partner assignment game* [Sotomayor, 1992; Biró *et al.*, 2018]. This game generalizes the matching game [Deng *et al.*, 1999; Biró *et al.*, 2012; Kern and Paulusma, 2003] to allow each individual to have more than one partner. Such a multi-partner setting makes the situation interesting and complex; thus, the game attracts the attention of theorists. Moreover, the game has several real-world applications such as the assignment of roommates [Granot, 1984; Tamir and Mitchell, 1998], the multiple-item version of the seller-buyer assignment [Shapley and Shubik, 1971; Biró *et al.*, 2012], and the international kidney exchange [Biró *et al.*, 2019]. Therefore, establishing efficient algorithms for the b-matching game is regarded as an important topic in the area.

Formally, the b-matching game is defined as follows. Let $G = (V, E)$ be an undirected graph, where V is the set of vertices, and E is the set of edges. Through the paper, we denote by $n = |V|$ and $m = |E|$. The vertices have *budgets* $b\colon V \to \mathbb{Z}_{\geq 1}$, and the edges have *weights* $c\colon E \to \mathbb{R}_{>0}$. Let $X \subseteq V$. A vector $x \in \mathbb{Z}^E$ is a *b-matching of X* if it satisfies the following conditions:

$$\sum_{e \in \delta(v)} x(e) \leq b(v), \qquad (v \in X), \tag{1.1}$$

$$x(e) = 0, \qquad (e \not\subseteq X), \tag{1.2}$$

where $\delta(v) = \{(u, v) \in E\}$ be the set of edges incident to v. The first condition is the budget constraint, and the second constraint requires that the support of x is contained in X. The *b-matching game* is a cooperative game (V, ν) whose characteristic function $\nu\colon 2^V \to \mathbb{R}$ is defined by

$$\nu(X) = \max_{x : b\text{-matching of X}} \sum_{e \in E} c(e)x(e). \tag{1.3}$$

The main research focus in algorithmic cooperative game theory is the complexity of finding solution concepts such as the core, the kernel, and the Shapley value. In this viewpoint, the b-matching game is much harder than the matching game. The core non-emptiness problem and the core-membership problem are solvable in polynomial time if $b = 1$ [Deng *et al.*, 1999; Biró *et al.*, 2012] (i.e., the matching game). However, these are co-NP-hard [Biró *et al.*, 2018; Biró *et al.*, 2019] if $b \geq 3$; note that they also showed that these are polynomial if $b \leq 2$. Therefore, it will be hopeless to obtain an element in the core in a realistic computational time.

Under these hardness results, we here focus on the convexity, which is often a computationally more tractable concept than the core non-emptiness. A cooperative game (V, ν) is *convex* if its characteristic function ν is *supermodular*, i.e., for all $X, Y \subseteq V$, the following inequality holds:

$$\nu(X) + \nu(Y) \leq \nu(X \cup Y) + \nu(X \cap Y). \tag{1.4}$$

Any convex game has a non-empty core, and an element of the core is obtained in polynomial time using an evaluation

oracle of ν. This implies that we can use the convexity as an alternative (a sufficient condition) of the core non-emptiness. Moreover, a convex game has its own benefits. For example, the Shapley value is in the core, and the kernel contains a unique point that coincides with the nucleolus, which can also be computed in polynomial time.

The b-matching game is not necessarily convex even if $b = 1$ [Kumabe and Maehara, 2020]. Therefore, we are interested in a condition when the game is convex. Also, we seek an algorithm that can check the convexity of a given b-matching game efficiently.

1.2 Our Contribution

In this study, we give a necessary and sufficient condition of the convexity of a b-matching game (Lemma 3.6). Based on this characterization, we propose a polynomial-time algorithm to check the convexity of a given b-matching game. Formally, the following is our main theorem.

Theorem 1.1. *There is an $O(n \log n + m\alpha(n))$ time algorithm to check whether a given b-matching game is convex, where α is the inverse Ackermann function*[1]

As an application of our characterization, we derive a polynomial-time algorithm to compute the Shapley value of a convex b-matching game. Computing the Shapley value is #P-hard even if $b = 1$ and $c(e) = 1$ for all $e \in E$ [Aziz and de Keijzer, 2014]; therefore, this result indicates that the convexity yields computational tractability on this game.

1.3 Related Work

Matching Games. Matching games are one of the well-studied games in algorithmic cooperative game theory. Shapley and Shubik 1971 considered the assignment game, which is a 1-matching game on a bipartite graph. Several authors [Deng *et al.*, 1999; Biró *et al.*, 2012; Kern and Paulusma, 2003] studied the 1-matching game. The b-matching game is also studied by several authors within several different settings [Deng *et al.*, 1999; Biró *et al.*, 2019; Biró *et al.*, 2018; Granot, 1984; Sotomayor, 1992; Tamir and Mitchell, 1998].

The b-matching game is a special case of the integral linear production game. The *linear production game* [Owen, 1975] is defined as follows. Let V be a set of players. For integers n and m, let $A \in \mathbb{R}^{n \times m}$, $b_v \in \mathbb{R}^n$ for each $v \in V$, and $c \in \mathbb{R}^m$. Then, the characteristic function ν of the linear production game is defined by

$$\nu(X) = \left(\begin{array}{ll} \text{maximize} & c^\top x \\ \text{subject to} & Ax \leq \sum_{v \in X} b_v, \\ & x \in \mathbb{R}^m_{\geq 0}. \end{array} \right. \tag{1.5}$$

The *integral linear production game* [Deng *et al.*, 2000] assumes the integrality of x to the above problem. The b-matching game is the integral linear production game in which A is the incidence matrix of the graph, and $b_v = b(v)e_v$, where e_v is the v-th unit vector.

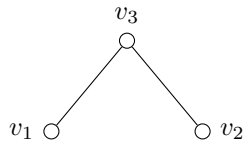

Figure 1: Configuration of Lemma 2.1.

Deng et al. 2000 studied the condition of the core non-emptiness of an integral linear production game when A's entries are zero and one, and $b(v) = 1$ for all $v \in V$. This case is also called the *packing game* [Deng *et al.*, 1999], *synergy coalition game* [Conitzer and Sandholm, 2006], and the *hypergraph matching game* [Kumabe and Maehara, 2020]. Introducing the budget $b \geq 2$ makes the situation complicated. Biro et al. [Biró *et al.*, 2018; Biró *et al.*, 2019] showed that the core non-emptiness problem and the core membership problem are solvable in polynomial time if $b(v) \leq 2$ for all $v \in V$; however, it is co-NP-hard if $b(v) \leq 3$ for all $v \in V$. This study aims at studying the b-matching from the convexity perspective.

Convexity of Games. The convexity is an important concept in algorithmic cooperative game theory [Deng *et al.*, 1999; Shapley, 1971]. Most combinatorial games are not convex; hence, they need additional conditions to be convex [Deng and Fang, 2008]. Researchers have been devoted their effort for revealing the condition of the convexity for several games, including the minimum base game [Nagamochi *et al.*, 1997], the maximum spanning tree game [Koh and Sanità, 2019; Okamoto, 2003], the minimum coloring game [Bietenhader and Okamoto, 2004], and the hypergraph matching game [Conitzer and Sandholm, 2006; Kumabe and Maehara, 2020]. This study aims at adding the b-matching game in this line of studies.

Note that the budget b affects a significant impact on the condition of the convexity. In the matching game case (i.e., $b = 1$), the game is convex if and only if the underlying graph is a matching, i.e., no two edges intersect [Kumabe and Maehara, 2020]. On the other hand, we can see much more complicated graph structures appear in a convex b-matching game with $b \geq 2$.

2 Warm-Up: Unweighted Case

Our goal is to give a necessary and sufficient condition of the convexity of the b-matching game and derive an efficient algorithm for checking the convexity. However, the general case, which will be given in the next section, is very complicated; therefore, as a warm-up, we start from the case when the graph has no edge weights, i.e., $c(e) = 1$ for all $e \in E$.

The proof strategy for this case and the general case are the same. We first derive a necessary condition of the convexity by considering a small graph. Then, we prove that the derived condition is also sufficient for the convexity.

[1] The inverse Ackermann function is an extremely slow-growing function, which is less than five for any practical purpose. See [Cormen *et al.*, 2001] for the definition.

2.1 Deriving Necessary Conditions

By looking at the interaction of three vertices, we obtain the following lemma.

Lemma 2.1. *Suppose that ν is supermodular. If $v_1, v_2, v_3 \in V$ satisfy $(v_1, v_3), (v_2, v_3) \in E$ but $(v_1, v_2) \notin E$ as in Figure 1, then $b(v_1) + b(v_2) \leq b(v_3)$ holds.*

Proof. We first observe that the following inequality holds:

$$\begin{aligned} &\min\{b(v_1), b(v_3)\} + \min\{b(v_2), b(v_3)\} && (2.1)\\ =&\nu(\{v_1, v_3\}) + \nu(\{v_2, v_3\}) && (2.2)\\ \leq&\nu(\{v_3\}) + \nu(\{v_1, v_2, v_3\}) && (2.3)\\ =&\min\{b(v_1) + b(v_2), b(v_3)\}, && (2.4)\end{aligned}$$

where the first line is the definition of ν, the second line is the supermodularity of ν, and the third line is from $\nu(\{v_3\}) = 0$ and the definition of ν.

Suppose the contrary that $b(v_1) + b(v_2) > b(v_3)$. Then, by Eq. (2.4), we see that $\min\{b(v_1), b(v_3)\} + \min\{b(v_2), b(v_3)\} \leq b(v_3)$. This indicates that $b(v_1) < b(v_3)$ and $b(v_2) < b(v_3)$. Therefore, $b(v_1) + b(v_2) = \min\{b(v_1), b(v_3)\} + \min\{b(v_2), b(v_3)\} \leq b(v_3)$, which leads to a contradiction. □

Using this lemma, we can determine the topology of the graph. Let $V = \{v_1, \ldots, v_n\}$ be the vertices of the graph. Without loss of generality, we assume that $b(v_1) \leq \cdots \leq b(v_n)$. For $i = 0, \ldots, n$, let $G_i = G[\{v_1, \ldots, v_i\}]$ be the subgraph induced by the vertices $\{v_1, \ldots, v_i\}$.[2] We denote by $N(v) = \{u \in V : (u, v) \in E\}$ the set of neighbors of v.

Lemma 2.2. *Suppose that ν is supermodular. If C is a connected component of G_{i-1}, then $N(v_i) \cap C$ is either $\emptyset$ or C.*

Proof. Suppose the contrary. Then, both $C^\circ = C \cap N(v_i)$ and $C^\bullet = C \setminus N(v_i)$ are non-empty. Because C is connected, there exists $u_1 \in C^\circ$ and $u_2 \in C^\bullet$ such that $(u_1, u_2) \in E$. By Lemma 2.1, $b(u_2) + b(v_i) \leq b(u_1)$ holds. However, this contradicts to the ordering of the vertices. □

Lemma 2.2 gives a "canonical representation" for the graph of a convex b-matching game. For $X \subseteq V$, we refer the *root* $r(X)$ of X as the vertex in X with the largest index.

Let C_i be the connected component of G_i that contains v_i. Note that, by Lemma 2.2, there are edges between v_i and all other vertices in C_i. Let $X_{i,1}, \ldots, X_{i,k_i}$ be the connected components of $C_i \setminus \{v_i\}$. Let $F_i = \{(v_i, r(X_{i,1})), \ldots, (v_i, r(X_{i,k_i}))\}$ be the set of edges from v_i to the connected components directed from v_i. Then, we construct a directed graph $T = (V, \bigcup_{i=1}^n F_i)$. This directed graph is a branching[3]. We refer to this branching as an *auxiliary forest*. By the construction and Lemma 2.2, the auxiliary forest can reconstruct the original graph by taking its comparability graph. Here, the *comparability graph* [Brandstadt *et al.*, 1999] of a directed acyclic graph is an undirected graph obtained by taking the transitive closure[4] and forgetting the edge orientations. Figure 2 shows an example of a graph and the corresponding auxiliary forest.

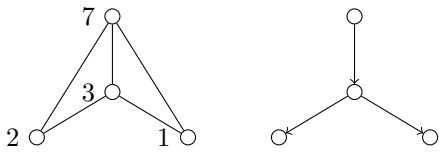

Figure 2: Original graph (left) and the auxiliary forest (right). The integers represents the budgets of the corresponding vertices.

Now we derive a condition about the budgets using the auxiliary forest T. We introduce a few notations. Let A_v be the set of proper ancestors[5] of v in T. Let T_v be the set of descendants v in T. For a vertex subset $X \subseteq V$, we denote by X^+ the set of vertices $v \in X$ that has no proper ancestor in X, and let $X^- := X \setminus X^+$. Note that $T_v^- = T_v \setminus \{v\}$.

Using these notations, we obtain the following lemma, which generalizes Lemma 2.1 to the more than three vertices. For $X \subseteq V$, we denote by $b(X) = \sum_{v \in X} b(v)$.

Lemma 2.3. *Suppose that ν is supermodular. Then, for all $v \in V$, we have $b(T_v^-) \leq b(v)$.*

Proof. We see

$$\begin{aligned} b(T_v^-) =& \sum_{u \in T_v^-} \nu(\{v, u\}) && (2.5)\\ \leq&\nu(T_v) + (|T_v^-| - 1)\nu(\{v\}) && (2.6)\\ \leq&\frac{b(T_v)}{2}, && (2.7)\end{aligned}$$

where the first line is from $b(u) \leq b(v)$ for all $u \in T_v^-$ and there is an edge $(u, v) \in E$ because the graph is obtained by the transitive closure of T, the second line is from supermodularity, and the last line follows from $\nu(\{v\}) = 0$ and the fact that each edge in the optimal b-matching consumes two unit of the budget. By arranging these terms we obtain $b(T_v^-) \leq b(v)$. □

Now we obtain the following two necessary conditions of the convexity of ν.

Condition U1 G is a comparability graph of a branching.

Condition U2 For every $v \in V$, $b(T_v^-) \leq b(v)$.

2.2 Proving the Sufficiency of Conditions

Now we prove that the above two necessary conditions, Conditions U1 and U2, are also sufficient for the convexity of ν. We first give an explicit formula of ν.

Lemma 2.4. *Suppose that Conditions U1 and U2 hold. Then, $\nu(X) = b(X^-)$ for all $X \subseteq V$.*

[2] For a graph $G = (V, E)$ and a vertex subset X, the subgraph *induced by* X is a graph $G[X] := (X, E[X])$ where $E[X] = \{(u, v) \in E : u, v \in X\}$.

[3] A directed graph is a *branching* if the in-degree of a vertex is at most one.

[4] The *transitive closure* of a directed graph $D = (V, F)$ is a directed acyclic graph $\overline{D} = (V, \overline{F})$ such that $(u, v) \in \overline{F}$ if there exists a path from u to v in D.

[5] Let $T = (V, F)$ be a branching. A vertex $u \in V$ is an *ancestor* of $v \in V$ if there exists a path from u to v. Here, v is a *descendant* of u. We say that u is a *proper ancestor (resp. proper descendant)* of v if u is an ancestor (resp. descendant) of u and $u \neq v$.

Proof. We prove this lemma for a connected X; the general case follows by adding the contributions of the connected components. In this case, X^+ consists of a single vertex. Thus, we put $X^+ = \{v\}$.

We first prove $\nu(X) \geq b(X^-)$. By Condition U1, we see that $(v, u) \in E$ for all $u \in X^-$. We observe that each (v, u) can be used $b(u)$ time because Condition U1 implies $b(X^-) \leq b(T_v^-)$, and Condition U2 implies $b(T_v^-) \leq b(v)$. Because the size of this b-matching is $b(X^-)$, we have $\nu(X) \geq b(X^-)$.

We then prove $\nu(X) \leq b(X^-)$. Each unit of an edge in a b-matching of X consumes at least one amount of budgets of vertices in X^-. Therefore, $\nu(X) \leq b(X^-)$. □

Using this representation, we prove that ν is supermodular.

Lemma 2.5. *Suppose that Conditions U1 and U2 hold. Then, ν is supermodular.*

Proof. We see that $b(X^+) = b(X) - b(X^-) = b(X) - \nu(X)$, where the last equality is Lemma 2.4. Thus, we prove that $b(X^+)$ is submodular in X (i.e., $-b(X^+)$ is supermodular in X).

We prove the submodularity for the case when $X \cup Y$ is connected; the general case follows by adding the contributions of the connected components. In this case, $(X \cup Y)^+$ consists of a single vertex. Thus, we put $(X \cup Y)^+ = \{v\}$. Without loss of generality, we assume that $v \in X^+$.

We prove $b((X \cap Y)^+) \leq b(Y^+)$ as follows. First,

$$b((X \cap Y)^+) = \sum_{w \in Y^+} \sum_{u \in (X \cap Y)^+ \cap T_w} b(u). \tag{2.8}$$

Thus, it is sufficient to show

$$\sum_{u \in (X \cap Y)^+ \cap T_w} b(u) \leq b(w) \tag{2.9}$$

for each $w \in Y^+$. If $w \in (X \cap Y)^+$ then $(X \cap Y)^+ \cap T_w = \{w\}$; therefore, the above inequality holds in equality. Otherwise,

$$\sum_{u \in (X \cap Y)^+ \cap T_w} b(u) \leq b(T_w^-) \leq b(w). \tag{2.10}$$

Here, the first inequality follows from $w \notin (X \cap Y)^+$, and the second inequality is from Condition U2. Using this inequality, we obtain

$$b(X^+) + b(Y^+) \geq b(X^+) + b((X \cap Y)^+) \tag{2.11}$$

$$= b((X \cup Y)^+) + b((X \cap Y)^+). \tag{2.12}$$

Here, the last line follows from $(X \cup Y)^+ = \{v\} = X^+$ because $v \in X^+$. □

2.3 Algorithm

Now we propose a polynomial-time algorithm to check the convexity of the b-matching game. By Lemma 2.5, it is sufficient to construct an algorithm that checks Conditions U1 and U2 efficiently.

Our algorithm, shown in Algorithm 1, maintains the family $\mathcal{S}$ of the connected components of $G[\{v_1, \ldots, v_i\}]$ and checks Conditions U1 and U2 in Lines 8 and 4, respectively. This algorithm gives the following theorem.

Algorithm 1 Checking the supermodularity of ν

Input: A graph $G = (V, E)$, budgets $b(v)$ for all $v \in V$

1: Sort the vertices in the non-decreasing order of the budgets. Let $v_1, \ldots, v_n$ be the order.
2: $\mathcal{S} \leftarrow \emptyset$
3: **for** $i = 1, \ldots, n$ **do**
4: **if** $b(N(v_i) \cap \{v_1, \ldots, v_{i-1}\}) > b(v_i)$ **then**
5: **return** `false`
6: **end if**
7: Let $\mathcal{C}_i = \{C \mid C \in \mathcal{S}, C \cap N(v_i) \neq \emptyset\}$
8: **if** $|\bigcup_{C \in \mathcal{C}_i} C| \neq |N(v_i) \cap \{v_1, \ldots, v_{i-1}\}|$ **then**
9: **return** `false`
10: **end if**
11: $\mathcal{S} \leftarrow (\mathcal{S} \setminus \mathcal{C}_i) \cup \{\bigcup_{C \in \mathcal{C}_i} C \cup \{v_i\}\}$
12: **end for**
13: **return** `true`

Theorem 2.6. *There is a $O(n \log n + m\alpha(n))$ time algorithm to check whether ν is supermodular, where $\alpha(n)$ is inverse-Ackerman function.*

Proof. The correctness of the algorithm is clear. Now we analyze the time complexity. Sorting the vertices can be done in $O(n \log n)$ time. Line 4 is done in $O(|N(v_i)|)$ time in each iteration. Thus, it takes $O(m)$ time in total. By maintaining $\mathcal{S}$ by the disjoint-set data structure [Galler and Fisher, 1964], Line 8 is evaluated by $O(|N(v_i)|)$ queries to the disjoint-set data structure in each step. Therefore, it takes $O(m\alpha(n))$ time in total. Therefore, the total time complexity is $O(n \log n + m\alpha(n))$. □

3 Main Result: Weighted Case

Now we describe our result in the weighted case. The proof strategy is the same as the unweighted case: We first derive necessary conditions of the convexity. Then we prove that these conditions are also sufficient. However, the details are much more complicated than the unweighted case. Therefore, due to the space limitation, we only give an outline of the proof. The complete proof will appear in the full paper.

3.1 Deriving Necessary Conditions

We start from the following lemma. This is a weighted version of Lemma 2.1 and is proved by a similar strategy to Lemma 2.1 with a careful case analysis.

Lemma 3.1. *Suppose that ν is supermodular. If v_1, v_2, v_3 satisfy $(v_1, v_3), (v_2, v_3) \in E$ but $(v_1, v_2) \notin E$ (see Figure 1), then $b(v_1) + b(v_2) \leq b(v_3)$.*

Lemma 3.1 is syntactically the same as Lemma 2.1. Therefore, it allows us to define the auxiliary forest in the same way as the unweighted case. This means that G should be a comparability graph of a branching. We use the same notations as in the unweighted case. The next lemma is a weighted version of Lemma 2.3.

Lemma 3.2. *Suppose that ν is supermodular. Then, for all $v \in V$, we have $b(T_v^-) \leq b(v)$.*

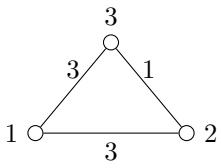

Figure 3: An example that shows Conditions W1, W2, and W3 are not sufficient for the convexity: Let X be the left and the top vertices and Y be the left and the right vertices. Then $\nu(X) = 3$, $\nu(Y) = 3$ but $\nu(X \cup Y) = 5$.

The proof is not as easy as Lemma 2.3. We use the following lemma. We write $c((u, v))$ as $c(u, v)$ for simplicity.

Lemma 3.3. *Suppose that ν is supermodular. Consider three vertices v_1, v_2, v_3 such that $b(v_1) \leq b(v_2) \leq b(v_3)$ and $(v_1, v_2), (v_2, v_3), (v_3, v_1) \in E$. Then, $c(v_1, v_3) \geq c(v_2, v_3)$.*

The proof of Lemma 3.3 is unexpectedly involved. We used a linear programming relaxation and a careful case analysis to prove this lemma.

Thus far, we obtain the following three necessary conditions of supermodularity of ν.

Condition W1 G is a comparability graph of a branching.

Condition W2 For every $v \in V$, $b(T_v^-) \leq b(v)$.

Condition W3 For every $v_1, v_2, v_3 \in V$ with $b(v_1) \leq b(v_2) \leq b(v_3)$ with edges (v_1, v_2), (v_1, v_3), we have $c(v_1, v_2) \leq c(v_1, v_3)$.

You should notice that Condition W3 does not assume the existence of (v_2, v_3) even though Lemma 3.3 assumes. This is because the existence is guaranteed from Condition W1, and by omitting this condition from Condition W3, we can reduce the complexity of checking Condition W3.

Unfortunately, these three conditions are not sufficient to make ν supermodular; Figure 3 shows an example. Therefore, we derive one more condition. For a vertex subset X and a vertex v with $A_v \cap X \neq \emptyset$, let $a_X(v)$ be the unique vertex in $X^+ \cap A_v$. For a vertex $s \in V^-$, let $p(s)$ be the parent (i.e., the immediate ancestor) of s in the auxiliary forest.

Lemma 3.4. *Assume that ν is supermodular. Then, the following condition holds:*

Condition W4 *For all vertex $s \in V^-$,*

$$\sum_{v \in T_s^-} c(s, v)b(v) \leq c(p(s), s)b(s) + \sum_{v \in (T_s^-)^-} c(a_{T_s^-}(v), v)b(v). \quad (3.1)$$

3.2 Proving the Sufficiency of Conditions

Now we obtain the four necessary conditions. We show that these are also sufficient for the convexity of ν.

We first give an explicit formula of ν, which is a weighted version of Lemma 2.4. Then, we obtain the following. The proof is similar to the proof of Lemma 2.4.

Lemma 3.5. *Suppose that Conditions W1, W2, and W3 hold. Then, for all $X \subseteq V$, $\nu(X) = \sum_{v \in X^-} c(a_X(v), v)b(v)$.*

We prove the submodularity of ν using this representation as follows. Its proof requires a long chain of inequalities.

Lemma 3.6. *Suppose that Conditions W1, W2, W3, and W4 hold. Then, ν is supermodular.*

3.3 Algorithm

Now we propose a polynomial-time algorithm to check the convexity of the b-matching game.

We first see that Conditions W1 and W2 are the same as the unweighted case. Therefore, these can be checked by the algorithm for the unweighted case (Algorithm 1). Condition W3 requires that for each vertex $v_1 \in V$ the ordering of $N(v_1) \cap \{u \in V : b(u) \geq b(v_1)\}$ by $b(\cdot)$ and the ordering by $c(v_1, \cdot)$ coincide. We can check this in $O(m)$ by comparing (v_1, v_2) and $(v_1, p(v_2))$ for each $(v_1, v_2) \in E$ with $b(v_2) \geq b(v_1)$. To check Condition W4, we first compute $\nu(T_v)$ for all $v \in V$. This is done in $O(m)$ time. Then, we check Condition W4 in $O(m)$ time.

In summary, we obtain the following theorem.

Theorem 3.7. *There is a $O(n \log n + m\alpha(n))$ time algorithm to check whether ν is supermodular.*

4 The Shapley Value

Our necessary and sufficient condition restricts the graph of a convex b-matching game as a comparability graph of a branching. By exploiting this structure, we can obtain an efficient algorithm to calculate the Shapley value of a convex b-matching game.

The Shapley value is defined as follows [Shapley, 1953]. Imagine that all the players join the coalition one-by-one in random order. Then, the marginal contribution of the player v to the coalition is given by $\nu(X \cup \{v\}) - \nu(X)$, where X is the set of players that appear before v. The Shapley value is the expectation of this quantity. Formally, it is given by

$$s_v := \sum_{X \subseteq V} \frac{|X|!(n - |X| - 1)!}{n!} \left(\nu(X \cup \{v\}) - \nu(X)\right). \quad (4.1)$$

For an efficient computation of the Shapley value, we try to represent the marginal contribution $\nu(X \cup \{v\}) - \nu(X)$ by a simple form.

As a preprocessing, we compute the auxiliary forest T of the graph, which is obtained during the algorithm in Theorem 1.1. For a vertex $v \in V$, we denote by $d(v)$ the *depth* of v, which is the maximum length of the path in T that ends at v. For example, if v has no parent in T, then $d(v) = 0$. For $i = 0, \ldots, d(v) - 1$, let a_i^v be the unique ancestor of v with $d(a_i^v) = i$.

We first evaluate the contribution of X with $X \cap A_v \neq \emptyset$. This is a direct consequence of Lemma 3.5.

Lemma 4.1. *Suppose that $X \cap A_v \neq \emptyset$. Then, $\nu(X \cup \{v\}) - \nu(X) = c(a_X(v), v)b(v)$.*

The event $a_X(v) = a_i^v$ occurs if and only if a_i^v and v are the first two elements added to X in $\{a_0^v, \ldots, a_i^v, v\}$ in this order. Therefore, such a probability is $1/(i + 1)(i + 2)$. This shows the next lemma.

Lemma 4.2. *The case $X \cap A_v \neq \emptyset$ contributes to the Shapley value by*

$$\sum_{i=0}^{d(v)-1} \frac{1}{(i+1)(i+2)} c(a_i^v, v) b(v). \tag{4.2}$$

Next, we evaluate the contribution from X with $X \cap A_v = \emptyset$. This is also a direct consequence of Lemma 3.5.

Lemma 4.3. *Suppose that $X \cap A_v = \emptyset$. Then,*

$$\nu(X \cup \{v\}) - \nu(X) \tag{4.3}$$

$$= \sum_{u \in X} c(v,u) b(u) - \sum_{u \in X^-} c(a_X(u), u) b(u). \tag{4.4}$$

For $u \in T_v^-$, the event $u \in X$ occurs if and only if u and v are the first two elements added to X in $A_v \cup \{u\}$ in this order. Therefore, such a probability is $1/(d(v)+1)(d(v)+2)$. Furthermore, for $w \in T_v^-$ and $u \in T_w^-$, the event $a_X(u) = w$ occurs if and only if w and v are the first two elements added to X in A_w in this order, and u is added before v. Therefore, such a probability is $2/(d(w)(d(w)+1)(d(w)+2))$. From the linearity of the expectation, we obtain the following.

Lemma 4.4. *The case $X \cap A_v = \emptyset$ contributes to the Shapley value by*

$$\frac{1}{(d(v)+1)(d(v)+2)} \sum_{u \in T_v^-} c(v,u) b(u) \tag{4.5}$$

$$- \sum_{w \in T_v^-} \frac{2}{d(w)(d(w)+1)(d(w)+2)} \sum_{u \in T_w^-} c(w,u) b(u). \tag{4.6}$$

From Lemmas 4.2 and 4.4, the Shapley value of a convex b-matching game is given as follows.

Theorem 4.5. *The Shapley value of the player v is given by*

$$\sum_{i=0}^{d(v)-1} \frac{1}{(i+1)(i+2)} c(a_i^v, v) b(v)$$
$$+ \frac{1}{(d(v)+1)(d(v)+2)} \sum_{u \in T_v^-} c(v,u) b(u)$$
$$- \sum_{w \in T_v^-} \frac{2}{d(w)(d(w)+1)(d(w)+2)} \sum_{u \in T_w^-} c(w,u) b(u). \tag{4.7}$$

This theorem immediately gives an efficient algorithm to compute the Shapley value as follows.

Theorem 4.6. *The Shapley values of convex b-matching game is computed in $O(n \log n + m\alpha(n))$ time.*

Proof. The algorithm is presented in Algorithm 2. The algorithm computes term of Eq. (4.7) independently: Lines 2–5 compute the second term in Eq. (4.7) in $O(n+m)$ time. Lines 6–10 compute the first term in Eq. (4.7). Because there are only $O(m)$ summands in the first term, this computation takes $O(m)$ time. Lines 11–15 compute the third term. Because $\sum_{v \in V} |T_v^-| = m$, this computation takes $O(m)$ time in total. Therefore, we obtain the correctness and the time complexity of the algorithm. □

Algorithm 2 Calculating the Shapley Value

Input: A graph $G = (V, E)$, budgets $b(v)$ for all $v \in V$, weights $c(e)$ for all $e \in E$, such that the b-matching game defined by G, b, c is convex

1: Calculate an auxiliary forest T of the game and the depth $d(v)$ for each $v \in V$
2: **for** $v \in V$ **do**
3: $\quad t_v \leftarrow \sum_{u \in T_v^-} c(v,u) b(u)$
4: **end for**
5: $s_v \leftarrow \frac{t_v}{(d(v)+1)(d(v)+2)}$ for all $v \in V$
6: **for** $v \in V$ **do**
7: $\quad$ **for** $i = 0, \ldots, d(v)-1$ **do**
8: $\quad\quad s_v \leftarrow s_v + \frac{c(a_v^i, v) b(u)}{(i+1)(i+2)}$
9: $\quad$ **end for**
10: **end for**
11: **for** $v \in V$ **do**
12: $\quad$ **for** $i = 0, \ldots, d(v)-1$ **do**
13: $\quad\quad s_v \leftarrow s_v - \sum_{w \in T_v^-} \frac{2t_w}{d(w)(d(w)+1)(d(w)+2)}$
14: $\quad$ **end for**
15: **end for**
16: **return** s_v for all $v \in V$

5 Conclusion

In this study, we give a necessary and sufficient condition of the convexity of the b-matching game. Then, we propose an efficient algorithm to check whether a given game is convex or not. Our characterization also gives an efficient algorithm to compute the Shapley value of a convex b-matching game.

There are several interesting future works. The most important future work is to extend the conditions of the convexity to more general games that include the matching games. A necessary and sufficient condition of the convexity of the hypergraph matching game is given in [Kumabe and Maehara, 2020]. Therefore, it will be natural to expect a characterization on the hypergraph b-matching game, which is a common generalization of the b-matching game and the hypergraph matching game.

Another interesting future work is about integrality. We can prove that Conditions W1 to W4 are still necessary and sufficient conditions for the *fractional b-matching game*, which relaxes the integrality condition of the problem. This implies the following property: "the fractional b-matching game is convex if and only if the integral b-matching game is convex." The same result is obtained in the hypergraph matching game [Kumabe and Maehara, 2020]. Therefore, we conjecture that the same result holds on the linear production game and the integral linear production game.

References

[Aziz and de Keijzer, 2014] Haris Aziz and Bart de Keijzer. Shapley meets Shapley. In *Proceedings of the 31st Inter-*

national Symposium on Theoretical Aspects of Computer Science (STACS'14), pages 99–111, 2014.

[Bietenhader and Okamoto, 2004] Thomas Bietenhader and Yoshio Okamoto. Core stability of minimum coloring games. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 389–401. Springer, 2004.

[Biró *et al.*, 2012] Péter Biró, Walter Kern, and Daniël Paulusma. Computing solutions for matching games. *International journal of game theory*, 41(1):75–90, 2012.

[Biró *et al.*, 2018] Péter Biró, Walter Kern, Daniël Paulusma, and Péter Wojuteczky. The stable fixture problem with payments. *Games and Economic Behaviour*, 108:245–268, 2018.

[Biró *et al.*, 2019] Péter Biró, Walter Kern, Dömötör Pálvölgyi, and Daniel Paulusma. Generalized matching games for international kidney exchange. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 413–421. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[Brandstadt *et al.*, 1999] Andreas Brandstadt, Jeremy P Spinrad, et al. *Graph classes: a survey*, volume 3. Siam, 1999.

[Conitzer and Sandholm, 2006] 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.

[Cormen *et al.*, 2001] Thomas H Cormen, Charles Eric Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*, volume 6. MIT press Cambridge, MA, 2001.

[Deng and Fang, 2008] Xiaotie Deng and Qizhi Fang. Algorithmic cooperative game theory. In *Pareto Optimality, Game Theory And Equilibria*, pages 159–185. Springer, 2008.

[Deng *et al.*, 1999] 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.

[Deng *et al.*, 2000] Xiaotie Deng, Toshihide Ibaraki, Hiroshi Nagamochi, and Wenan Zang. Totally balanced combinatorial optimization games. *Mathematical Programming*, 87(3):441–452, 2000.

[Galler and Fisher, 1964] Bernard A Galler and Michael J Fisher. An improved equivalence algorithm. *Communications of the ACM*, 7(5):301–303, 1964.

[Granot, 1984] Daniel Granot. A note on the room-mates problem and a related revenue allocation problem. *Management Science*, 30(5):633–643, 1984.

[Kern and Paulusma, 2003] Walter Kern and Daniël Paulusma. Matching games: the least core and the nucleolus. *Mathematics of operations research*, 28(2):294–308, 2003.

[Koh and Sanità, 2019] Zhuan Khye Koh and Laura Sanità. An efficient characterization of submodular spanning tree games. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 275–287. Springer, 2019.

[Kumabe and Maehara, 2020] Soh Kumabe and Takanori Maehara. Convexity of hypergraph matching game. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'20)*, pages 663–671. International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[Nagamochi *et al.*, 1997] Hiroshi Nagamochi, Dao-Zhi Zeng, Naohiśa Kabutoya, and Toshihide Ibaraki. Complexity of the minimum base game on matroids. *Mathematics of Operations Research*, 22(1):146–164, 1997.

[Okamoto, 2003] Yoshio Okamoto. Submodularity of some classes of the combinatorial optimization games. *Mathematical Methods of Operations Research*, 58(1):131–139, 2003.

[Owen, 1975] Guillermo Owen. On the core of linear production games. *Mathematical programming*, 9(1):358–370, 1975.

[Shapley and Shubik, 1971] Lloyd S Shapley and Martin Shubik. The assignment game i: The core. *International Journal of game theory*, 1(1):111–130, 1971.

[Shapley, 1953] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[Shapley, 1971] Lloyd S Shapley. Cores of convex games. *International Journal of Game Theory*, 1(1):11–26, 1971.

[Sotomayor, 1992] Marilda Sotomayor. The multiple partners game. In *Equilibrium and Dynamics*, pages 322–354. Springer, 1992.

[Tamir and Mitchell, 1998] Arie Tamir and Joseph SB Mitchell. A maximum b-matching problem arising from median location models with applications to the roommates problem. *Mathematical Programming*, 80(2):171–194, 1998.