Mathematical Programming (2023) 198:1–25
https://doi.org/10.1007/s10107-021-01752-w

FULL LENGTH PAPER

Series A

Arboricity games: the core and the nucleolus

Han Xiao[1] · **Qizhi Fang**[1]

Received: 29 October 2020 / Accepted: 30 November 2021 / Published online: 29 January 2022

Abstract
The arboricity of a graph is the minimum number of forests required to cover all its edges. In this paper, we examine arboricity from a game-theoretic perspective and investigate cost-sharing in the minimum forest cover problem. We introduce the arboricity game as a cooperative cost game defined on a graph. The players are edges, and the cost of each coalition is the arboricity of the subgraph induced by the coalition. We study properties of the core and propose an efficient algorithm for computing the nucleolus when the core is not empty. In order to compute the nucleolus in the core, we introduce the prime partition which is built on the densest subgraph lattice. The prime partition decomposes the edge set of a graph into a partially ordered set defined from minimal densest minors and their invariant precedence relation. Moreover, edges from the same partition always have the same value in a core allocation. Consequently, when the core is not empty, the prime partition significantly reduces the number of variables and constraints required in the linear programs of Maschler's scheme and allows us to compute the nucleolus in polynomial time. Besides, the prime partition provides a graph decomposition analogous to the celebrated core decomposition and the density-friendly decomposition, which may be of independent interest.

Keywords Core · Nucleolus · Arboricity · Density · Graph decomposition

Mathematics Subject Classification 05C57 · 91A12 · 91A43 · 91A46

Supported in part by the National Natural Science Foundation of China (Nos. 12001507, 11971447, 11871442) and the Natural Science Foundation of Shandong (No. ZR2020QA024).

✉ Han Xiao
hxiao@ouc.edu.cn

Qizhi Fang
qfang@ouc.edu.cn

[1] School of Mathematical Sciences, Ocean University of China, Qingdao, China

1 Introduction

The arboricity of a graph is the minimum number of forests required to cover all edges of the graph. Hence arboricity concerns forest cover, a special case of matroid covering. Besides, arboricity is a measure of graph density. A graph with large arboricity always contains a dense subgraph. By employing the nontrivial interplay between forest cover and graph density under the polyhedral framework, we examine arboricity from a game-theoretic perspective and introduce the so-called arboricity game. Briefly, the arboricity game is a cooperative cost game defined on a graph, where the players are edges and the cost of each coalition is the arboricity of the subgraph induced by the coalition.

A central question in cooperative game theory is to distribute the total cost to its participants. Many solution concepts have been proposed for cost-sharing. One solution concept is the core, which requires that no coalition benefits by breaking away from the grand coalition. Another solution concept is the nucleolus, which is the unique solution that lexicographically maximizes the vector of non-decreasingly ordered excess. Following the definition, Kopelowitz [22] and Maschler et al. [24] proposed a standard procedure to compute the nucleolus by solving a sequence of linear programs. However, the size of these linear programs may be exponentially large due to the number of constraints corresponding to all possible coalitions. Hence it is in general unclear how to apply this procedure. The first polynomial algorithm for computing the nucleolus was proposed by Megiddo [25] for cooperative cost games defined on directed trees. Later on, a number of polynomial algorithms were developed for, e.g., bankruptcy games [1], matching games [5,9,19,20,32], standard tree games [18], airport profit games [6], flow games [10], voting games [14], spanning connectivity games [2], shortest path games [3], and network strength games [4]. On the negative side, NP-hardness results for computing the nucleolus were shown for, e.g., minimum spanning tree games [15], threshold games [13], b-matching games [21], flow games and linear production games [10,16].

The main contribution of this paper is twofold. One contribution is concerned with the arboricity game, where cost-sharing in the minimum forest cover problem is considered. We study properties of the core and propose an efficient algorithm for computing the nucleolus when the core is nonempty. Our results are in the same spirit as [4,20], but justifications are different. The other contribution goes to the prime partition, which is a graph decomposition analogous to the celebrated core decomposition [31] and the density-friendly decomposition [33,34]. The prime partition is inspired by the principle partition of matroids [7] and by the graph decompositions developed in [2,4]. For arboricity games, the prime partition dramatically reduces the size of linear programs involved in Maschler's scheme and enables us to compute the nucleolus in polynomial time.

The rest of this paper is organized as follows. Section 2 introduces relevant concepts. Section 3 reviews some polyhedral results on arboricity. Section 4 studies properties of the core. Section 5 is devoted to the prime partition, a graph decomposition of independent interest. Section 6 develops an efficient algorithm for computing the nucleolus. Section 7 concludes this paper.

2 Preliminaries

A *cooperative game* $\Gamma = (N, \gamma)$ consists of a player set N and a characteristic function $\gamma : 2^N \to \mathbb{R}$ with convention $\gamma(\emptyset) = 0$. The player set N is called the *grand coalition*. Any subset S of N is called a *coalition*. Given a vector $\boldsymbol{x} \in \mathbb{R}^N$, we use $x(S)$ to denote $\sum_{i \in S} x_i$ for any $S \subseteq N$. A vector $\boldsymbol{x} \in \mathbb{R}^N_{\geq 0}$ is called an *allocation* of Γ if $x(N) = \gamma(N)$. The *excess* of a coalition S at an allocation $\boldsymbol{x}$ is defined as $e(S, \boldsymbol{x}) = \gamma(S) - x(S)$. The *core* of Γ, denoted by $\mathcal{C}(\Gamma)$, is the set of allocations where all excesses are nonnegative, i.e.,

$$\mathcal{C}(\Gamma) = \left\{\boldsymbol{x} \in \mathbb{R}^N_{\geq 0} : x(N) = \gamma(N);\ x(S) \leq \gamma(S),\ \forall S \subseteq N\right\}.$$

The *excess vector* $\theta(\boldsymbol{x})$ of an allocation $\boldsymbol{x}$ is the $2^{|N|} - 2$ dimensional vector whose components are the non-trivial excesses $e(S, \boldsymbol{x})$ for $S \in 2^N \backslash \{\emptyset, N\}$ arranged in a non-decreasing order. The *nucleolus* [28] is the unique allocation $\boldsymbol{x}$ that lexicographically maximizes the excess vector $\theta(\boldsymbol{x})$. When the core is nonempty, the nucleolus always exists and lies in the core. Moreover, the nucleolus can always be computed with a standard procedure of Maschler et al. [22,24] by recursively solving a sequence of linear programs.

$$\begin{aligned}
& & \max\ & \epsilon & & (1)\\
& (LP_1) & \text{s.t.}\ & x(N) = \gamma(N), & & (2)\\
& & & x(S) + \epsilon \leq \gamma(S), & \forall\, S \in 2^N \backslash \{\emptyset, N\}, & (3)\\
& & & x_i \geq 0, & \forall\, i \in N. & (4)
\end{aligned}$$

To compute the nucleolus with Maschler's scheme, first solve linear program LP_1 to maximize the minimum excess among all non-trivial coalitions. For any constant ϵ, let $P_1(\epsilon)$ denote the set of vectors $\boldsymbol{x} \in \mathbb{R}^N$ such that $(\boldsymbol{x}, \epsilon)$ satisfies (2)–(4), i.e., $P_1(\epsilon)$ is the set of allocations whose minimum excess is no less than ϵ. It follows that $\mathcal{C}(\Gamma) = P_1(0)$. Let ϵ_1 be the optimal value of LP_1. Then $P_1(\epsilon_1)$ is the set of optimal solutions of LP_1, which is also called the *least core* of Γ. Thus $\mathcal{C}(\Gamma) \neq \emptyset$ if and only if $\epsilon_1 \geq 0$. For any polyhedron $P \subseteq \mathbb{R}^N$, let $\text{Fix}(P)$ denote the set of coalitions *fixed* by P, i.e.,

$$\text{Fix}(P) = \left\{S \subseteq N : x(S) = y(S),\ \forall\, \boldsymbol{x}, \boldsymbol{y} \in P\right\}.$$

After solving linear program LP_r, let ϵ_r be the optimal value and $P_r(\epsilon_r)$ be the set of optimal solutions. Then solve linear program LP_{r+1} to maximize the minimum excess on coalitions that are not fixed by $P_r(\epsilon_r)$.

$$\begin{aligned}
& & \max\ & \epsilon & & (5)\\
& (LP_{r+1}) & \text{s.t.}\ & x(S) + \epsilon \leq \gamma(S), & \forall\, S \notin \text{Fix}\big(P_r(\epsilon_r)\big), & (6)\\
& & & \boldsymbol{x} \in P_r(\epsilon_r). & & (7)
\end{aligned}$$

Clearly, $\epsilon_{r+1} \geq \epsilon_r$ and $P_{r+1}(\epsilon_{r+1}) \subseteq P_r(\epsilon_r)$. Moreover, the dimension of $P_{r+1}(\epsilon_{r+1})$ decreases before it collapses to zero. Hence it takes up to $|N|$ rounds before $P_{r+1}(\epsilon_{r+1})$ becomes a singleton which is exactly the nucleolus. However, the linear programs involved in Maschler's scheme are usually of exponential size. Even if linear programs $LP_1, \ldots, LP_r$ have been successfully solved, it may be intractable in polynomial time to determine all coalitions not fixed by $P_r(\epsilon_r)$. Hence it is in general unclear how to apply Maschler's scheme. For arboricity games, we show that the number of variables and constraints required in the successive linear programs of Maschler's scheme can be dramatically reduced, and the nucleolus can always be determined efficiently on the second round of Maschler's scheme.

We assume that the readers have a moderate familiarity with graph theory. But assumptions, notions and notations used in this paper should be clarified before proceeding. Throughout this paper, we assume that all graphs are loopless but parallel edges are allowed. We also assume that loops are always removed during edge contraction. An *image* is a vertex obtained from edge contraction. A *minor* is a graph obtained from repeated vertex deletion, edge deletion and edge contraction. Let $G = (V, E)$ be a graph. We use $c(G)$ to denote the number of components of G. We use $n(G)$ and $m(G)$ to denote the number of vertices and edges in G respectively. We write n for $n(G)$ and m for $m(G)$ when no ambiguity occurs. Let $U \subseteq V$ be a set of vertices. We write $G - U$ for the graph obtained from G by deleting all vertices in U. Let $F \subseteq E$ be a set of edges. We write G/F for the graph obtained from G by contracting all edges in F. Let H be a subgraph of G. We write $G - H$ for $G - V(H)$ and write G/H for $G/E(H)$. If X and Y are two sets of vertices, we use (X, Y) and $m(X, Y)$ to denote the set and the number of crossing edges between X and Y respectively. If X and Y are two subgraphs, we write (X, Y) for $\big(V(X), V(Y)\big)$.

3 Polyhedral results on arboricity

This section reviews some polyhedral results on arboricity. For more details about polyhedral combinatorics, we refer to [29,30].

Let $G = (V, E)$ be a graph. A *forest cover* of G is a set of forests that covers all edges of G. The *arboricity* of G, denoted by $a(G)$, is the minimum size of forest covers of G. The arboricity measures how dense a graph is. The *density* of G, denoted by $g(G)$, is the value of $\frac{m(G)}{n(G)-c(G)}$. Hence $g(G) = \frac{m(G)}{n(G)-1}$ if G is connected. By convention, the density of a single vertex is zero. Nash-Williams [26] showed that the arboricity of a graph is lower bounded by the maximum density of subgraphs.

Theorem 1 (Nash-Williams [26]) *The edges of a graph G can be covered by k forests if and only if* $\max_{H \subseteq G} g(H) \leq k$.

The value of $\max_{H \subseteq G} g(H)$ is called the *fractional arboricity* of G and denoted by $a_f(G)$. Theorem 1 implies that $a(G) = \lceil a_f(G) \rceil$. Notice that the fractional arboricity is necessarily achieved at connected subgraphs. It follows that the fractional arboricity of G can be computed by

$$\max_{H \subseteq G} \frac{m(H)}{n(H)-1}. \tag{8}$$

Let $\mathcal{F}$ denote the set of forests in G. Clearly, $\mathcal{F}$ makes a graphic matroid with ground set E, and every forest cover of G is essentially an independent set cover of the graphic matroid. Additionally, the definition for density and (fractional) arboricity in the forest cover problem respects the conventional definition in the matroid covering problem [12]. Hence the forest cover problem is a special case of the matroid covering problem. Notice that the fractional arboricity of a matroid is always equal to the fractional cover number of independent sets [27,30]. It follows that the value of (8) is equal to the optimal value of linear program (9)–(11)

$$\min \quad \sum_{F \in \mathcal{F}} z_F \tag{9}$$

$$\text{s.t.} \quad \sum_{F: e \in F} z_F \geq 1, \qquad \forall\, e \in E, \tag{10}$$

$$z_F \geq 0, \qquad \forall\, F \in \mathcal{F}. \tag{11}$$

and the optimal value of its dual (12)–(14).

$$\max \quad \sum_{e \in E} x_e \tag{12}$$

$$\text{s.t.} \quad \sum_{e: e \in F} x_e \leq 1, \qquad \forall\, F \in \mathcal{F}, \tag{13}$$

$$x_e \geq 0, \qquad \forall\, e \in E. \tag{14}$$

Notice that (8) can be reformulated as

$$\max_{H \subseteq G} \mathbf{1} \cdot \frac{\boldsymbol{\lambda}^H}{n(H)-1}, \tag{15}$$

where $\mathbf{1} \in \mathbb{Z}^E$ is an all-one vector and $\boldsymbol{\lambda}^H \in \mathbb{Z}^E$ is the incidence vector of $E(H)$. Moreover, $\frac{\boldsymbol{\lambda}^H}{n(H)-1}$ satisfies (13) and (14) for any $H \subseteq G$. Consequently, optimal solutions of (12)–(14) are among the vectors $\frac{\boldsymbol{\lambda}^H}{n(H)-1}$ in (15), which leads to the following corollary that will be used in Section 4. In the remainder of this paper, a *densest subgraph* always refers to a *connected* subgraph with the maximum density, and a *densest minor* always refers to a *connected* minor the density of which is equal to the fractional arboricity of the graph.

Lemma 2 *The set of optimal solutions of* (12)–(14) *is the convex hull of the vectors* $\frac{\boldsymbol{\lambda}^H}{n(H)-1}$ *for every densest subgraph* H *of* G.

Lemma 2 suggests that the set of optimal solutions of (12)–(14) is a convex polytope where every extreme point corresponds to a densest subgraph. By polyhedral theory [29], all faces of a convex polytope form a partially ordered set under inclusion. It

turns out that all densest subgraphs also form a partially ordered set under inclusion, which suggests that faces of the optimal solution polytope of (12)–(14) may be related to densest subgraphs. It is this observation that leads to the graph decomposition in Sect. 5.

4 The core and its properties

Throughout this paper, we always assume that the underlying graph of arboricity games is connected. Let $\Gamma_G = (N, \gamma)$ denote the *arboricity game* defined on a graph $G = (V, E)$, where $N = E$ and $\gamma(S) = a(G[S])$ for $S \subseteq N$. We start with an alternative characterization for the core.

Lemma 3 *Let $\Gamma_G = (N, \gamma)$ be an arboricity game and $\mathcal{T}$ be the set of spanning trees in G. Then*

$$\mathcal{C}(\Gamma_G) = \left\{\boldsymbol{x} \in \mathbb{R}^E_{\geq 0} : x(E) = \gamma(E);\ x(T) \leq 1,\ \forall\, T \in \mathcal{T}\right\}. \tag{16}$$

Proof Denote by $\mathcal{C}'(\Gamma_G)$ the right hand of (16). We first show that $\mathcal{C}(\Gamma_G) \subseteq \mathcal{C}'(\Gamma_G)$. Let $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$. For any $T \in \mathcal{T}$, we have $x(T) \leq \gamma(T) = 1$. It follows that $\boldsymbol{x} \in \mathcal{C}'(\Gamma_G)$. Now we show that $\mathcal{C}'(\Gamma_G) \subseteq \mathcal{C}(\Gamma_G)$. Let $\boldsymbol{x} \in \mathcal{C}'(\Gamma_G)$ and $S \in 2^N \backslash \{\emptyset\}$. Assume that $\gamma(S) = k$ and $G[S]$ can be covered by k forests $F_1, \ldots, F_k$. Let T_i be a spanning tree containing F_i. It follows that

$$x(S) = \sum_{i=1}^{k} x(F_i) \leq \sum_{i=1}^{k} x(T_i) \leq k = \gamma(S),$$

which implies that $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$. □

A necessary and sufficient condition for the core nonemptiness follows immediately.

Theorem 4 *Let $\Gamma_G = (N, \gamma)$ be an arboricity game. Then $\mathcal{C}(\Gamma_G) \neq \emptyset$ if and only if $a_f(G) = a(G)$.*

Proof Let $\boldsymbol{x}$ be an optimal solution of (12)–(14). It follows that $x(E) = a_f(G) \leq a(G) = \gamma(E)$. Since $\mathcal{T} \subseteq \mathcal{F}$, Lemma 3 implies that $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$ if $a_f(G) = a(G)$.

Let $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$. Since every forest is a subgraph of a spanning tree, Lemma 3 implies that $\boldsymbol{x}$ is a feasible solution of (12)–(14). It follows that $x(E) \leq a_f(G) \leq a(G) = \gamma(E)$. Hence $x(E) = \gamma(E)$ implies $a_f(G) = a(G)$. □

Since the arboricity game is a special case of the covering game, Theorem 4 respects the universal characterization for the core nonemptiness of covering games [11]. The corollary below also follows from the results for covering games.

Corollary 5 *Let $\Gamma_G = (N, \gamma)$ be an arboricity game. Then the nonemptiness of $\mathcal{C}(\Gamma_G)$ can be determined in polynomial time. Moreover, we can decide in polynomial time if a vector belongs to $\mathcal{C}(\Gamma_G)$, and if not, find a separating hyperplane.*

Theorem 4 implies that, when the core is nonempty, a vector belongs to the core if and only if it is an optimal solution of (12)–(14). Lemma 2 suggests that the nonempty core can be characterized by the incidence vector of densest subgraphs. Thus we have the following corollary.

Corollary 6 *Let $\Gamma_G = (N, \gamma)$ be an arboricity game with a nonempty core. Then $\mathcal{C}(\Gamma_G)$ is the convex hull of the vectors $\frac{\lambda^H}{n(H)-1}$ for every densest subgraph H of G.*

Corollary 6 implies that every core allocation is a convex combination of vectors, each of which is associated with a densest subgraph. For edges not in any densest subgraph, we have the following corollary.

Corollary 7 *Let $\Gamma_G = (N, \gamma)$ be an arboricity game with a nonempty core. For any $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$, $x_e = 0$ if edge e does not belong to any densest subgraph of G.*

It is well known that the nucleolus lies in the core when the core is nonempty. To compute the nucleolus in the core, we need a better understanding of the core polytope. Corollary 6 states that every extreme point of the core polytope is associated with a densest subgraph, which suggests that faces of the core polytope may also be associated with densest subgraphs. Inspired by the face lattice of convex polytopes [29], we introduce a graph decomposition built on densest subgraphs, which is crucial for computing the nucleolus in the core of arboricity games.

5 The prime partition

This section is self-contained and devoted to the prime partition, a graph decomposition analogous to the core decomposition [31] and the density-friendly decomposition [33,34]. The prime partition is inspired by the face lattice of convex polytopes and built on the densest subgraph lattice where the edge set intersection of any two densest subgraphs is either empty or inducing a densest subgraph again. By utilizing the uncrossing technique [23] to a chain of subgraphs with the maximum density, we introduce the prime partition. The *prime partition* decomposes the edge set of a graph into a *non-prime set* and a number of *prime sets*. The *non-prime set* is the set of edges that are not in any densest subgraph. The *prime sets* are the incremental edge sets of a chain of subgraphs with the maximum density. In general, there is more than one chain of subgraphs with the maximum density that defines the prime sets. A partial order can be defined on the prime sets according to the invariant inclusion relation in any chain of subgraphs defining the prime sets. There are other graph decompositions [2,4] inspired by the face lattice of convex polytopes. But they are defined on different discrete structures. The remainder of this section is organized as follows. In Sect. 5.1, we investigate properties of minimal densest subgraphs which are basic ingredients of the prime partition. In Sect. 5.2, we define prime sets by levels and introduce the non-prime set as a byproduct. In Sect. 5.3, we show that every densest subgraph admits a unique decomposition with prime sets. In Sect. 5.4, we introduce the ancestor relation of prime sets and define a partial order from the ancestor relation. Throughout this section, we always assume that the graph $G = (V, E)$ is connected.

5.1 Minimal densest subgraphs

The following properties of minimal densest subgraphs are useful in defining the prime partition.

Lemma 8 (Cut-vertex-free property) *Let H be a minimal densest subgraph of G. Then H has no cut vertex.*

Proof Assume to the contrary that v is a cut vertex in H. Let H_1 and H_2 be two subgraphs of H such that $H_1 \cup H_2 = H$ and $H_1 \cap H_2 = \{v\}$. Since H is a minimal densest subgraph, we have

$$g(H_i) = \frac{m(H_i)}{n(H_i) - 1} < g(H), \tag{17}$$

for $i = 1, 2$. It follows that

$$g(H) = \frac{m(H)}{n(H) - 1} = \frac{m(H_1) + m(H_2)}{[n(H_1) - 1] + [n(H_2) - 1]} < g(H), \tag{18}$$

which is a contradiction. Hence H has no cut vertex. □

Lemma 9 (Noncrossing property) *Let H be a minimal densest subgraph of G. For any densest subgraph K of G, either $E(H) \subseteq E(K)$ or $E(H) \cap E(K) = \emptyset$.*

Proof When $E(H) \cap E(K) \neq \emptyset$, assume to the contrary that $E(H) \nsubseteq E(K)$. Let $X = H \cap K$. Then X is a proper subgraph of H with $E(X) \neq \emptyset$. On one hand, we have

$$\frac{m(H - X) + m(H - X, X)}{n(H - X)} > g(H). \tag{19}$$

Indeed, since otherwise

$$\begin{aligned} g(X) = \frac{m(X)}{n(X) - 1} &= \frac{m(H) - m(H - X) - m(H - X, X)}{n(H) - n(H - X) - 1} \\ &\geq \frac{m(H) - n(H - K) \cdot g(H)}{n(H) - n(H - K) - 1} = \frac{m(H) - n(H - K) \cdot \frac{m(H)}{n(H)-1}}{n(H) - n(H - K) - 1} = g(H), \end{aligned} \tag{20}$$

which contradicts the minimality of H. On the other hand, we have

$$\frac{m(K - X) + m(K - X, X) + m(X)}{n(K - X) + n(X) - 1} = g(K). \tag{21}$$

Since $g(H) = g(K)$, (19) and (21) imply that

$$\begin{aligned} g(H \cup K) &\geq \frac{[m(H - X) + m(H - X, X)] + [m(K - X) + m(K - X, X) + m(X)]}{n(H - X) + [n(K - X) + n(X) - 1]} \\ &> g(K), \end{aligned} \tag{22}$$

which contradicts the maximum density of K. Hence either $E(H) \subseteq E(K)$ or $E(H) \cap E(K) = \emptyset$. □

Lemma 9 implies that any two minimal densest subgraphs share no common edge, which is the key property for defining prime sets. We also notice that any two minimal densest subgraphs share at most one common vertex. This observation can be generalized to a "cycle"-free property for minimal densest subgraphs.

Lemma 10 ("Cycle"-free property) *Let $H_1, \ldots, H_r$ be minimal densest subgraphs of G. Then $|\{v : v \in V(H_i) \cap V(H_j),\ i \neq j\}| < r$.*

Proof Assume to the contrary that $|\{v : v \in V(H_i) \cap V(H_j),\ i \neq j\}| \geq r$. Let $H = \cup_{i=1}^{r} H_i$. Lemma 9 implies that

$$\begin{aligned} g(H) &\geq \frac{\sum_{i=1}^{r} m(H_i)}{\sum_{i=1}^{r} n(H_i) - |\{v : v \in V(H_i) \cap V(H_j),\ i \neq j\}| - 1} \\ &> \frac{\sum_{i=1}^{r} m(H_i)}{\sum_{i=1}^{r} [n(H_i) - 1]} = g(H_1), \end{aligned} \tag{23}$$

which contradicts the maximum density of H_1. □

To illustrate the "cycle"-free property, we introduce an auxiliary graph $\mathcal{H}(G)$. Every vertex v_H in $\mathcal{H}(G)$ is associated with a minimal densest subgraph H of G. Every edge in $\mathcal{H}(G)$ joins two vertices v_{H_1} and v_{H_2} in $\mathcal{H}(G)$ if H_1 and H_2 share a common vertex. Lemma 10 implies that if any three minimal densest subgraphs of G share no common vertex, then $\mathcal{H}(G)$ is acyclic. The "cycle"-free property will be used repeatedly in our arguments.

To define the prime partition, we have to determine all minimal densest subgraphs. Gabow [17] provided an $O(nm \log \frac{n^2}{m})$ algorithm for computing the fractional arboricity of a graph with n vertices and m edges. By employing the algorithm of Gabow, the enumeration of all minimal densest subgraphs can be done in polynomial time.

Lemma 11 *All minimal densest subgraphs of G can be enumerated in $O(n^3 m \log \frac{n^2}{m})$.*

Proof We first show that a minimal densest subgraph of G can be found in $O(n^2 m \log \frac{n^2}{m})$. Initially, compute the fractional arboricity of G and let $H = G$. Compute the fractional arboricity of $H - v$ where $v \in V(H)$. If $a_f(H - v) = a_f(G)$, then a densest subgraph of G can be found in $H - v$. Update H with $H - v$ and repeat the process for H until $a_f(H - v) < a_f(G)$ for any $v \in V(H)$. Then H is a minimal densest subgraph of G. It takes $O(n)$ iterations before achieving a minimal densest subgraph of G. Each iteration, which involves computing the fractional arboricity of a subgraph of G, can be done in $O(nm \log \frac{n^2}{m})$. Hence a minimal densest subgraph of G can be found in $O(n^2 m \log \frac{n^2}{m})$.

Now we show that all minimal densest subgraphs of G can be enumerated in $O(n^3 m \log \frac{n^2}{m})$. Let $H_1, \ldots, H_k$ be minimal densest subgraphs that have been found in G. Let $G_k = G - \cup_{i=1}^{k} E(H_i)$. Compute the fractional arboricity of G_k. If $a_f(G_k) =$

$a_f(G)$, then a minimal densest subgraph H_{k+1} of G can be found in G_k and let $G_{k+1} = G - \cup_{i=1}^{k+1} E(H_i)$. Repeat this process for G_{k+1} until $a_f(G_{k+1}) < a_f(G)$. Then no minimal densest subgraph of G remains in G_{k+1}. Lemma 10 implies that any two minimal densest subgraphs share at most one common vertex and there is a "cycle"-free property among minimal densest subgraphs. It follows that $\cup_{i=1}^{k+1} H_i$ has at least one more vertex than $\cup_{i=1}^{k} H_i$. Therefore, there are $O(n)$ minimal densest subgraphs of G. A minimal densest subgraph of G can be found in $O(n^2 m \log \frac{n^2}{m})$. Therefore, all minimal densest subgraphs of G can be enumerated in $O(n^3 m \log \frac{n^2}{m})$. □

5.2 Defining prime sets by levels

Now we define prime sets in the prime partition. In short, every prime set is the edge set of a minimal densest minor. Since edge contractions are involved, we introduce prime sets by levels. A *prime set of level zero* in G is the edge set of a minimal densest subgraph. By Lemma 9, prime sets of level zero are well defined. Moreover, Lemma 11 implies that all prime sets of level zero can be enumerated efficiently. To define prime sets of higher levels, we study properties of densest subgraphs under edge contraction.

Lemma 12 (Density preserving contraction) *Let H be a proper densest subgraph of G. Then $g(G/H) \le g(G)$ and $a_f(G/H) \le a_f(G)$. Moreover, both equalities hold if $g(G) = a_f(G)$.*

Proof Let $\hat{G} = G/H$ and v_H be the image of H in $\hat{G}$. We first prove that $g(\hat{G}) \le g(G)$ and the equality holds if $g(G) = f(G)$. Notice that $m(\hat{G} - v_H) = m(G - H)$, $m(\hat{G} - v_H, v_H) = m(G - H, H)$ and $n(G - H) = n(\hat{G} - v_H)$. Hence $g(G) \le g(H)$ implies that

$$\underbrace{\frac{m(\hat{G} - v_H) + m(\hat{G} - v_H, v_H)}{n(\hat{G} - v_H)}}_{=g(\hat{G})} \le \underbrace{\frac{[m(G - H) + m(G - H, H)] + m(H)}{n(G - H) + [n(H) - 1]}}_{=g(G)} \le \underbrace{\frac{m(H)}{n(H) - 1}}_{=g(H)}. \tag{24}$$

Notice that $g(\hat{G}) = \frac{m(G)-m(H)}{n(G)-n(H)}$. Therefore, $g(\hat{G}) = g(G)$ if $g(G) = g(H)$.

Now we prove that $a_f(\hat{G}) \le a_f(G)$. It suffices to show that $g(\hat{G}') \le a_f(G)$ for any induced subgraph $\hat{G}'$ of $\hat{G}$. When $v_H \notin V(\hat{G}')$, it is trivial that $g(\hat{G}') \le a_f(G)$. Now assume that $v_H \in V(\hat{G}')$. Then there is an induced subgraph G' of G such that $H \subseteq G'$ and $\hat{G}' = G'/H$. Hence (24) implies that $g(\hat{G}') \le g(G') \le a_f(G)$. It follows that $a_f(\hat{G}) \le a_f(G)$. We have seen that $g(G) = g(H)$ implies $g(\hat{G}) = g(G)$. Therefore, $a_f(\hat{G}) = a_f(G)$ if $g(G) = g(H)$. □

Lemma 12 implies that contracting a densest subgraph does not change the fractional arboricity if this subgraph is a proper subgraph of another densest subgraph. By Lemma 9, minimal densest subgraphs possess an uncrossing property. Hence minimal

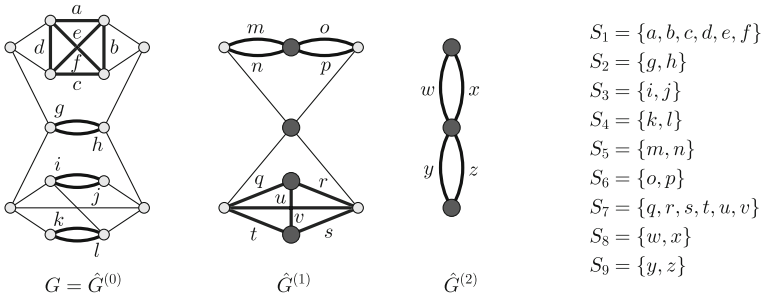

Fig. 1 An example for the prime partition

densest subgraphs can be contracted simultaneously. After contracting all minimal densest subgraphs, if the fractional arboricity remains unchanged, densest subgraphs of the resulting graph are densest minors of the original graph; moreover, minimal densest subgraphs of the resulting graph are used to define prime sets of level one. The procedure can be repeated to define prime sets of higher levels until the fractional arboricity of the resulting graph decreases. In the following, we formally define prime sets of higher levels.

Let $\hat{G}^{(0)} = G$ and $\hat{G}^{(k+1)}$ be the graph obtain from $\hat{G}^{(k)}$ by contracting all edges in minimal densest subgraphs of $\hat{G}^{(k)}$, where $k \geq 0$. If $a_f(\hat{G}^{(k+1)}) = a_f(G)$, then a *prime set of level* $k+1$ is the edge set of a minimal densest subgraph in $\hat{G}^{(k+1)}$. Otherwise, there is no more prime set and the edge set of $\hat{G}^{(k+1)}$ is the *non-prime set*. Therefore, every prime set is essentially the edge set of a minimal densest minor, and the non-prime set is the set of edges that are not in any minimal densest minor. For simplicity, we introduce some notations for the prime partition of G. For a prime set P of level k, we use $n(P)$ to denote the number of vertices in its defining minimal densest subgraph $\hat{G}^{(k)}[P]$. We use $\mathcal{P}_k$ to denote the collection of all prime sets of level k, use $\mathcal{P} = \cup_k \mathcal{P}_k$ to denote the collection of all prime sets, use E_0 to denote the non-prime set, and use $\mathcal{E} = \mathcal{P} \cup \{E_0\}$ to denote the prime partition. Figure 1 provides an example of the prime partition. The fractional arboricity of G is 2. There are 4 prime sets of level zero, 3 prime sets of level one, and 2 prime sets of level two. The non-prime set is empty.

Since the enumeration of minimal densest subgraphs can be done in polynomial time, the prime partition can be computed efficiently.

Theorem 13 *The prime partition of G has $O(n)$ prime sets and can be computed in $O(n^4 m \log \frac{n^2}{m})$.*

Proof Lemmas 9 and 10 imply that contractions on different minimal densest subgraphs can be performed simultaneously. Notice that $n(\hat{G}^{(k+1)}) \leq n(\hat{G}^{(k)}) - |\mathcal{P}_k|$. Consequently, there are $O(n)$ prime sets in $\mathcal{P}$. The definition of prime sets naturally yields an efficient algorithm for computing the prime partition of G. Since there are $O(n)$ prime sets, it takes $O(n)$ iterations to compute the prime partition of G, and each iteration computes all prime sets of the same level. Computing all prime sets of the same level is equivalent to enumerating all minimal densest subgraphs, which

can be done in $O(n^3 m \log \frac{n^2}{m})$. Hence the prime partition of G can be computed in $O(n^4 m \log \frac{n^2}{m})$. □

5.3 Decomposing densest subgraphs with prime sets

To show that any densest subgraph admits a decomposition of prime sets, we first generalize the uncrossing property of minimal densest subgraphs to prime sets.

Lemma 14 (Generalized noncrossing property) *For any prime set P and any densest subgraph H of G, either $P \subseteq E(H)$ or $P \cap E(H) = \emptyset$.*

Proof We apply induction on the level of prime set P. When $P \in \mathcal{P}_0$, $G[P]$ is a minimal densest subgraph of G. Lemma 9 implies that either $P \subseteq E(H)$ or $P \cap E(H) = \emptyset$.

Now assume that $P \in \mathcal{P}_l$ where $l \geq 1$ and assume that for any prime set Q of level less than l either $Q \subseteq E(H)$ or $Q \cap E(H) = \emptyset$. Let $\hat{H}^{(0)} = H$ and $\hat{H}^{(k+1)}$ be the graph obtained from $\hat{H}^{(k)}$ by contracting all edges in prime sets of level k, where $k \geq 0$. Assume that $E(\hat{H}^{(l)}) \neq \emptyset$, since otherwise we have $P \cap E(H) = \emptyset$. By the induction hypothesis and Lemma 12, we have $g(\hat{H}^{(l)}) = g(H) = a_f(G) = a_f(\hat{G}^{(l)})$. Hence $\hat{H}^{(l)}$ is a densest subgraph of $\hat{G}^{(l)}$. Since $\hat{G}^{(l)}[P]$ is a minimal densest subgraph of $\hat{G}^{(l)}$, Lemma 9 implies that either $P \subseteq E(\hat{H}^{(l)})$ or $P \cap E(\hat{H}^{(l)}) = \emptyset$. Therefore, either $P \subseteq E(H)$ or $P \cap E(H) = \emptyset$. □

It follows from Lemma 14 that every densest subgraph admits a decomposition of prime sets.

Lemma 15 (Prime set decomposition) *For any densest subgraph H of G, there are prime sets $P_1, \ldots, P_r$ such that $E(H) = \cup_{i=1}^{r} P_i$, where $r \geq 1$. Moreover, $n(H) = \sum_{i=1}^{r} [n(P_i) - 1] + 1$.*

Proof Lemmas 12 and 14 imply that there are prime sets $P_1, \ldots, P_r$ such that $E(H) = \cup_{i=1}^{r} P_i$. Assume that $P_1, \ldots, P_r$ are arranged in a non-decreasing order of levels. Let $\hat{H}_0 = H$ and $\hat{H}_k = \hat{H}_{k-1}/P_k$ for $k = 1, \ldots, r$. By Lemmas 9 and 10, we have $n(\hat{H}_k) = n(\hat{H}_{k-1}) - n(P_k) + 1$. Besides, $n(\hat{H}_r) = 1$. Therefore, $n(H) = \sum_{i=1}^{r} [n(\hat{H}_{i-1}) - n(\hat{H}_i)] + n(\hat{H}_r) = \sum_{i=1}^{r} [n(P_i) - 1] + 1$. □

Since the non-prime set consists of edges that are not in any densest subgraph, we have the following corollary.

Lemma 16 *Let E_0 be the non-prime set of G. Then every component of $G - E_0$ is a densest subgraph of G.*

The left Venn diagram in Figure 2 illustrates the relation of all 11 densest subgraphs of G in Figure 1. It shows that the intersection of any two densest subgraphs is either empty or a densest subgraph again. Hence, all densest subgraphs of G, together with $\emptyset$, form a lattice under inclusion. It also shows that every densest subgraph can be decomposed into prime sets.

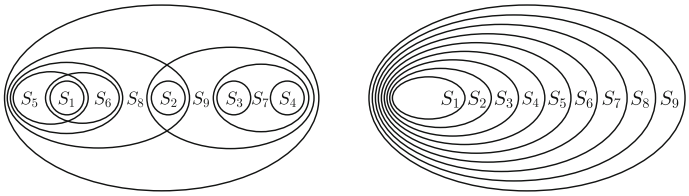

Fig. 2 Illustrations for the prime partition of G in Figure 1

5.4 The ancestor relation

Lemmas 14 and 15 suggest that there exists a laminar family of subgraphs with the maximum density that defines the prime sets. We may determine a chain of subgraphs that defines the prime sets as follows. Start with $G_0 = G - E_0$. For $k \geq 1$, let S_k be the edge set of a minimal densest subgraph in G_{k-1}, G_k be the graph obtained from G_{k-1} by contracting edges in S_k, and $H_k = G[\cup_{i=1}^{k} S_i]$. Then $H_l = G - E_0$ for some integer l and $H_k = G[\cup_{i=1}^{k} S_i]$ is a subgraph with the maximum density for $k = 1, \ldots, l$. Consequently, $\{H_1, \ldots, H_l\}$ with $H_1 \subsetneq \ldots \subsetneq H_l$ is a chain of subgraphs with the maximum density that defines the prime sets. Furthermore, every prime set is precisely the incremental edge set of consecutive subgraphs in the chain. The right Venn diagram in Fig. 2 provides a chain of subgraphs with the maximum density that defines the prime sets of G in Fig. 1. It shows that all prime sets of G are precisely incremental edges sets of subgraphs in the chain.

Generally, there is more than one chain of subgraphs with the maximum density that defines the prime sets. However, some prime sets are always preceded by other prime sets in any chain of subgraphs defining the prime sets, as some minimal densest minors occur only after the contraction of other minimal densest minors. Therefore, we introduce the notion of ancestor to represent the invariant precedence relation in the prime sets. A prime set Q is called an *ancestor* of a prime set P if the minimal densest minor defining P occurs only after the contraction of Q. Alternatively, Q is an ancestor of P if Q always precedes P in any chain of subgraphs with the maximum density that defines the prime sets. Clearly, the ancestor relation is *transitive*. If Q is an ancestor of P but not an ancestor of any other ancestors of P, then Q is called a *parent* of P.

To prove the ancestor relation is well defined, it suffices to show that the parent relation is well defined since the ancestor relation is transitive. Let P be a prime set of level $k \geq 1$. Let G' denote the graph obtained from G by contracting all non-parent ancestors of P by levels, i.e., first contract all non-parent ancestors of level zero and then repeatedly contract all non-parent ancestors of higher levels. Let G'' denote the graph obtained from G' by contracting all parents of P. Let e_1, e_2 be edges from P that become incident at some vertex in G''. Let $Q_1, \ldots, Q_r$ be a minimal collection of parents of P whose contraction concatenates e_1 and e_2. Clearly, $G'[\cup_{i=1}^{r} Q_i]$ is connected. Denote by v_Q the image of $\cup_{i=1}^{r} Q_i$ in G''. We show that $Q_1, \ldots, Q_r$ make the *unique* collection of parents of P that concatenates e_1 and e_2 at v_Q in G''. Assume to the contrary that there exists another collection of parents of P, say

$Q'_1, \ldots, Q'_s$, such that $G'[\cup_{i=1}^{s} Q'_i]$ is connected and v_Q is the image of $\cup_{i=1}^{s} Q'_i$ in G''. Then G' has a "cycle" consisting of minimal densest subgraphs induced by prime sets from $Q_1, \ldots, Q_r, Q'_1, \ldots, Q'_s$, which contradicts Lemma 10. Hence the parent relation is well defined.

Notice that all ancestors of a prime set constitute the minimal collection of prime sets that have to be contracted before arriving at its corresponding minimal densest minor. Thus if a densest subgraph contains a prime set, it also contains all ancestors of the prime set.

Lemma 17 *Let H be a densest subgraph of G. Let P be a prime set of G and Q be an ancestor of P. Then $P \subseteq E(H)$ implies $Q \subseteq E(H)$.*

Moreover, the ancestor relation can be determined efficiently.

Lemma 18 *Given the prime partition of G, the ancestor relation can be determined in $O(n^2m)$.*

Proof Let P be a prime set of level $k+1$, where $k \geq 0$. We show that all ancestors of P can be determined in $O(nm)$. To determine all ancestors of P, it suffices to check every prime set Q of level less than $k+1$. Let $\hat{G}_{-Q}^{(l+1)}$ denote the graph obtained from $\hat{G}_{-Q}^{(l)}$ by contracting all edges in prime sets of level l, where $\hat{G}_{-Q}^{(0)} = G - Q$. Then Q is an ancestor of P if and only if $n(\hat{G}_{-Q}^{(k+1)}[P]) \neq n(\hat{G}^{(k+1)}[P])$. Since there are $O(n)$ prime sets, all ancestors of P can be determined in $O(nm)$. Therefore, all ancestors for $O(n)$ prime sets can be determined in $O(n^2m)$. □

We conclude this section with a partially ordered set defined from the prime sets and the ancestor relation. Indeed, if we view every prime set as an ancestor of itself, then the ancestor relation naturally yields a partial order on the prime sets. Write $P \prec Q$ for any two prime sets P and Q if Q is an ancestor of P. Consequently, a partial order $\prec$ is defined on the prime sets from the ancestor relation.

6 Computing the nucleolus

In this section, we develop an efficient algorithm for computing the nucleolus of arboricity games when the core is not empty. In Sect. 6.1, we employ the prime partition of the underlying graph to reformulate linear programs involved in Maschler's scheme. In Sect. 6.2, we prove the correctness of our formulation for Maschler's scheme. In Sect. 6.3, we show that Maschler's scheme always terminates on the second round and the nucleolus can be computed in polynomial time. Throughout this section, in addition to assuming that graph $G = (V, E)$ is connected, we further assume that arboricity game $\Gamma_G = (N, \gamma)$ has a nonempty core.

6.1 Reformulating Maschler's scheme

To compute the nucleolus of Γ_G, the first round of Maschler's scheme is to solve linear program LP_1 (1)–(4) defined from the standard characterization for the core.

By referring to the alternative characterization for the core in Lemma 3, we introduce linear program LP_1' (25)–(28). For any constant ϵ, let $P_1'(\epsilon)$ denote the set of vectors $\boldsymbol{x} \in \mathbb{R}^E$ such that $(\boldsymbol{x}, \epsilon)$ satisfies (26)–(28). We show that LP_1 and LP_1' are equivalent.

$$
\begin{aligned}
(LP_1') \qquad & \max \quad \epsilon && && (25)\\
& \text{s.t.} \quad x(E) = \gamma(E), && && (26)\\
& \qquad\quad x(T) + \epsilon \le 1, && \forall\, T \in \mathcal{T}, && (27)\\
& \qquad\quad x_e \ge 0, && \forall\, e \in E. && (28)
\end{aligned}
$$

Lemma 19 *Let ϵ_1 and ϵ_1' be the optimal value of LP_1 and LP_1' respectively. Then $\epsilon_1 = \epsilon_1'$ and $P_1(\epsilon_1) = P_1'(\epsilon_1')$.*

Proof We first show that $\epsilon_1 = \epsilon_1'$. It is easy to see that $\epsilon_1 \le \epsilon_1'$, since LP_1' is a relaxation of LP_1. Let $S \subseteq E$ and $\mathcal{C}_S$ be a minimum forest cover in $G[S]$. For any $\boldsymbol{x} \in P_1'(\epsilon_1')$, Lemma 3 implies that

$$\gamma(S) - x(S) = \sum_{T_S \in \mathcal{C}_S} 1 - \sum_{e \in S} x_e = \sum_{T_S \in \mathcal{C}_S} (1 - \sum_{e \in T_S} x_e) \ge \sum_{T_S \in \mathcal{C}_S} \epsilon_1' \ge \epsilon_1'. \quad (29)$$

We remark that the last inequality follows from the assumption that $\mathcal{C}(\Gamma_G) \neq \emptyset$ which implies $\epsilon_1 \ge 0$. By the optimality of ϵ_1, we have $\epsilon_1 \ge \epsilon_1'$. Thus $\epsilon_1 = \epsilon_1'$ follows.

Next we show that $P_1(\epsilon_1) = P_1'(\epsilon_1')$. Clearly, $P_1(\epsilon_1) \subseteq P_1'(\epsilon_1')$, since $\epsilon_1 = \epsilon_1'$ and LP_1' is a relaxation of LP_1. Since for any $\boldsymbol{x} \in P_1'(\epsilon_1')$, $\boldsymbol{x}$ also satisfies the constraints of LP_1 and gives the optimum. Then $\boldsymbol{x} \in P_1(\epsilon_1)$, which implies that $P_1'(\epsilon_1') \subseteq P_1(\epsilon_1)$. Thus, $P_1(\epsilon_1) = P_1'(\epsilon_1')$. □

Before proceeding to the second round of Maschler's scheme, we have to determine the optimal value ϵ_1' of LP_1'. Clearly, $\epsilon_1' \ge 0$ as $\mathcal{C}(\Gamma_G) \neq \emptyset$. Assume that $\gamma(E) = k$ and that G can be covered by k disjoint forests $F_1, \ldots, F_k$. Let T_i be a spanning tree containing F_i and $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$. Clearly, $x(F_i) \le x(T_i) \le \gamma(T_i) = 1$. It follows that $x(E) = \sum_{i=1}^{k} x(F_i) \le \sum_{i=1}^{k} x(T_i) \le k = \gamma(E)$. Then $x(E) = \gamma(E)$ implies $x(F_i) = x(T_i) = 1$. Hence $\epsilon_1' = 0$, implying that the core $P_1'(0)$ and the least core $P_1'(\epsilon_1')$ coincide. Consequently, there are spanning trees $T \in \mathcal{T}$ such that $x(T) = 1$ for any $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$, i.e., T is fixed by $P_1'(\epsilon_1')$. Denote by $\mathcal{T}_0$ the set of spanning trees that are fixed by $P_1'(\epsilon_1')$. Let E_0 denote the set of edges that are not in any densest subgraph of G. Corollary 7 implies that $x_e = \epsilon_1' = 0$ for any $e \in E_0$. By Lemma 19, the second round of Maschler's scheme can be formulated as LP_2' from LP_1'.

$$
\begin{aligned}
(LP_2') \qquad & \max \quad \epsilon && && (30)\\
& \text{s.t.} \quad x(T) + \epsilon \le 1, && \forall\, T \in \mathcal{T} \backslash \mathcal{T}_0, && (31)\\
& \qquad\quad x(T) = 1, && \forall\, T \in \mathcal{T}_0, && (32)\\
& \qquad\quad x_e \ge \epsilon, && \forall\, e \in E \backslash E_0, && (33)\\
& \qquad\quad x_e = 0, && \forall\, e \in E_0. && (34)
\end{aligned}
$$

However, LP_2' still has an exponential number of constraints. We derive an equivalent formulation of LP_2' that has only polynomial size by resorting to the prime partition of G. Notice that E_0 in (34) is precisely the non-prime set of G. Let $\mathcal{P} = \cup_k \mathcal{P}_k$ denote the collection of all prime sets of G, where $\mathcal{P}_k$ is the collection of all prime sets of level k. Let $\mathcal{E} = \mathcal{P} \cup \{E_0\}$ denote the the prime partition of G. Corollary 7 states that all edges in the non-prime set have the same value in a core allocation. It turns out that this property also holds for edges from the same prime set.

Lemma 20 *Let $\boldsymbol{x}$ be a core allocation of Γ_G and P be a prime set of G. Then $x_e = x_f$ for any $e, f \in P$.*

Proof Let $\boldsymbol{x}^H$ be the vector associated with a densest subgraph H of G. By Lemma 14, either $P \subseteq E(H)$ or $P \cap E(H) = \emptyset$. Thus for any $e, f \in P$, $x_e^H = x_f^H = \frac{1}{n(H)-1}$ if $P \subseteq E(H)$ and $x_e^H = x_f^H = 0$ otherwise. Since any vector $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$ is a convex combination of vectors associated with a densest subgraph of G, we have $x_e = x_f$ for any $e, f \in P$. □

Corollary 7 and Lemma 20 state that all edges in the same set of the prime partition have the same value in a core allocation. Hence every core allocation $\boldsymbol{x} \in \mathbb{R}^E$ of Γ_G defines a vector $\boldsymbol{y} \in \mathbb{R}^{\mathcal{E}}$ associated with the prime partition of G. Moreover, LP_2' can be reformulated with $\boldsymbol{y}$. Let $(\mathcal{P}, \prec)$ denote the partially ordered set defined on $\mathcal{P}$ from the ancestor relation. Let $\mathcal{P}_{\min}$ denote the set of minimal prime sets in $(\mathcal{P}, \prec)$. Denote by LP_2'' the linear program (35)–(39) defined on $\boldsymbol{y}$. In Sect. 6.2, we show that LP_2' and LP_2'' are equivalent, i.e., $\boldsymbol{x}$ is feasible to LP_2' if and only if $\boldsymbol{y}$ is feasible to LP_2''. In Sect. 6.3, we propose a combinatorial algorithm for LP_2'' and show that LP_2'' has a unique optimal solution which yields the nucleolus of Γ_G.

$$
(LP_2'') \quad
\begin{aligned}
&\max \quad \epsilon && && (35)\\
&\text{s.t.} \quad y_P + \epsilon \le y_Q, && \forall\ P \prec Q, && (36)\\
&\qquad \textstyle\sum_{P \in \mathcal{P}} [n(P) - 1] y_P = 1, && && (37)\\
&\qquad y_P \ge \epsilon, && \forall\ P \in \mathcal{P}_{\min}, && (38)\\
&\qquad y_{E_0} = 0. && && (39)
\end{aligned}
$$

6.2 Equivalence of LP_2' and LP_2''

Let $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$ and $\boldsymbol{y} \in \mathbb{R}^{\mathcal{E}}$ be a pair of associated vectors. We show that $\boldsymbol{x}$ is feasible to LP_2' if and only if $\boldsymbol{y}$ is feasible to LP_2''. By Corollary 7, it is trivial that (34) and (39) are equivalent. The equivalence of (33) and (38) follows from Lemma 20 and the observation below.

Lemma 21 *Let $\boldsymbol{x}$ be a core allocation of Γ_G and P, Q be two prime sets of G such that Q is an ancestor of P. Then $x_e \le x_f$ for any $e \in P$ and any $f \in Q$.*

Proof Let $\boldsymbol{x}^H$ be the vector associated with a densest subgraph H of G. Lemma 17 implies that $Q \subseteq E(H)$ if $P \subseteq E(H)$. If follows that for any $e \in P$ and any $f \in Q$, $x_e^H = x_f^H = \frac{1}{n(H)-1}$ if $P \subseteq E(H)$ and $x_e^H = 0 \le x_f^H$ otherwise. Since any vector

$\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$ is a convex combination of vectors associated with densest subgraph of G, we have $x_e \leq x_f$ for any $e \in P$ and any $f \in Q$. □

Next, we show the equivalence of (32) and (37). Notice that (32) provides a characterization for trees in $\mathcal{T}_0$ with $\boldsymbol{x} \in \mathbb{R}^E$. Thus (37) serves the same purpose. To associate trees in $\mathcal{T}_0$ with $\boldsymbol{y} \in \mathbb{R}^{\mathcal{E}}$, we introduce the following lemma.

Lemma 22 *A spanning tree T belongs to $\mathcal{T}_0$ if and only if for any prime set P we have*

$$|T \cap P| = n(P) - 1. \tag{40}$$

Proof ($\Leftarrow$) Assume that T is a spanning tree satisfying (40) for any prime set P. Let H be a densest subgraph of G. The vector $\boldsymbol{x}^H$ associated with H is defined by $x_e^H = \frac{1}{n(H)-1}$ if $e \in E(H)$ and $x_e^H = 0$ otherwise. By Lemma 15, we have $E(H) = \cup_{i=1}^r P_i$ and $n(H) = \sum_{i=1}^r [n(P_i) - 1] + 1$, where $P_i \in \mathcal{P}$ for $i = 1, \ldots, r$. Thus

$$x^H(T) = \frac{\sum_{i=1}^r |T \cap P_i|}{n(H) - 1} = \frac{\sum_{i=1}^r [n(P_i) - 1]}{n(H) - 1} = \frac{n(H) - 1}{n(H) - 1} = 1.$$

Since every vector in $\mathcal{C}(\Gamma_G)$ is a convex combination of vectors associated with densest subgraph of G, we have $x(T) = 1$ for any $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$, implying that $T \in \mathcal{T}_0$.

($\Rightarrow$) Assume that $T \in \mathcal{T}_0$, i.e., T is a spanning tree such that $x(T) = 1$ for any $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$. Among all densest subgraphs of G containing P, let H be a minimal one. We apply induction on the level of prime set $P \in \mathcal{P}$.

First assume that $P \in \mathcal{P}_0$. Hence $P = E(H)$, implying that H is a densest subgraph of G. Then the vector $\boldsymbol{x}^H \in \mathcal{C}(\Gamma_G)$ associated with H is defined by $x_e^H = \frac{1}{n(H)-1}$ for $e \in P$ and $x_e^H = 0$ otherwise. Since $x^H(T) = \frac{1}{n(H)-1}|T \cap P| = 1$, it follows that $|T \cap P| = n(H) - 1 = n(P) - 1$. Hence (40) holds for P.

Now assume that $P \in \mathcal{P}_k$, where $k \geq 1$. Lemma 15 implies that there exist prime sets $Q_1, \ldots, Q_r$ such that $E(H) = P \cup (\cup_{i=1}^r Q_i)$. The minimality of H implies that $P \prec Q_i$ for $i = 1, \ldots, r$. By induction hypothesis, we have $|T \cap (\cup_{i=1}^r Q_i)| = \sum_{i=1}^r [n(Q_i) - 1]$. Then the vector $\boldsymbol{x}^H \in \mathcal{C}(\Gamma)$ associated with H is defined by $x_e^H = \frac{1}{n(H)-1}$ if $e \in P \cup (\cup_{i=1}^r Q_i)$ and $x_e^H = 0$ otherwise. Since

$$x^H(T) = \frac{|T \cap [P \cup (\cup_{i=1}^r Q_i)]|}{n(H) - 1} = \frac{|T \cap P| + \sum_{i=1}^r [n(Q_i) - 1]}{n(H) - 1} = 1,$$

Lemma 15 implies that $|T \cap P| = n(H) - 1 - \sum_{i=1}^r [n(Q_i) - 1] = n(P) - 1$. Hence (40) holds for $P \in \mathcal{P}_k$ where $k \geq 1$. □

Now we are ready to prove the equivalence of (32) and (37).

Lemma 23 *Let $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$ be a vector and $\boldsymbol{y} \in \mathbb{R}^{\mathcal{E}}$ be the vector defined from $\boldsymbol{x}$. Then $\boldsymbol{x}$ satisfies (32) if and only if $\boldsymbol{y}$ satisfies (37).*

Proof Notice that every spanning tree $T \in \mathcal{T}$ admits a decomposition from the prime partition. Lemma 22 implies that

$$x(T) = \sum_{e \in T} x_e = |T \cap E_0| y_{E_0} + \sum_{P \in \mathcal{P}} |T \cap P| y_P = \sum_{P \in \mathcal{P}} [n(P) - 1] y_P = 1. \quad (41)$$

Thus $\boldsymbol{x}$ satisfies (32) if and only if $\boldsymbol{y}$ satisfies (37). □

Finally, we come to the equivalence of LP_2' and LP_2''. We first show that any vector $\boldsymbol{y} \in \mathbb{R}^{\mathcal{E}}$ defined from a feasible solution $\boldsymbol{x} \in \mathbb{R}^E$ of LP_2' satisfies (36) and hence is a feasible solution of LP_2''. Our proof is based on the idea that for two any prime sets P and Q with $P \prec Q$, a specific spanning tree outside $\mathcal{T}_0$ can be constructed from any spanning tree in $\mathcal{T}_0$ by repeatedly performing edge exchanges along a pathway consisting of ancestors of P and ending with Q. To this end, we need the following lemma.

Lemma 24 *Let $P, Q \in \mathcal{P}$ be a pair of prime sets with $P \prec Q$ in $(\mathcal{P}, \prec)$. For any spanning tree $T \in \mathcal{T}_0$, there exists a spanning tree $T' \in \mathcal{T} \backslash \mathcal{T}_0$ such that*

$$|T' \cap P| = |T \cap P| + 1, \quad (42)$$
$$|T' \cap Q| = |T \cap Q| - 1, \quad (43)$$
$$|T' \cap R| = |T \cap R|, \qquad \forall\, R \in \mathcal{P} \backslash \{P, Q\}. \quad (44)$$

Proof Assume that $P \in \mathcal{P}_l$ and $Q \in \mathcal{P}_{l-r}$, where $l \geq r \geq 1$. We claim that there exist

- a sequence $S_0, \ldots, S_r$ of ancestors of P such that $S_0 = P$, $S_r = Q$ and $S_k \in \mathcal{P}_{l-k}$ for $k = 0, \ldots, r$;
- a sequence $T_0, \ldots, T_r$ of spanning trees obtained from T such that $T_0 = T$ and

$$T_{k+1} = T_k + e_k' - e_{k+1}, \quad (45)$$

  where $e_k, e_k' \in S_k$ for $k = 0, \ldots, r$.

It follows that

$$T_{k+1} = T_0 + e_0' - \sum_{i=1}^{k} (e_i - e_i') - e_{k+1}. \quad (46)$$

Notice that $|T_{k+1} \cap S_0| = |T_0 \cap S_0| + 1$, $|T_{k+1} \cap S_{k+1}| = |T_0 \cap S_{k+1}| - 1$, and $|T_{k+1} \cap S| = |T_0 \cap S|$ for any $S \in \mathcal{P} \backslash \{S_0, S_{k+1}\}$. Lemma 22 implies that T_r is a spanning tree satisfying (42)–(44) in $\mathcal{T} \backslash \mathcal{T}_0$.

The sequence $S_0, \ldots, S_r$ sets a pathway for edge exchange operations in (45), which can be identified as follows. Start with $S_r = Q$ and work backwards. Suppose $S_{k+1} \in \mathcal{P}_{l-k-1}$ has been identified. If S_{k+1} is a parent of an ancestor $R \in \mathcal{P}_{l-k}$ of P, then let $S_k = R$. Otherwise, there exist two ancestors $R_1, R_2 \in \mathcal{P}_{l-k}$ of P such that $\hat{G}^{(l-k-1)}[R_1]$ and $\hat{G}^{(l-k-1)}[R_2]$ share no common vertex but $\hat{G}^{(l-k)}[R_1]$ and $\hat{G}^{(l-k)}[R_2]$ share a common vertex s_{k+1} which is the image of S_{k+1} in $\hat{G}^{(l-k)}$.

Then let S_k be any one of R_1 and R_2, say $S_k = R_1$. Repeat this process until $S_0 = P$. Denote by $\mathcal{S}$ the set of $S_0, \ldots, S_r$.

It remains to show how to perform edge exchange operations in (45). Let $\hat{G}_k$ be the graph obtained from G by contracting all edges in prime sets of level less than S_k and all edges in other prime sets of the same level with S_k. Let s_{k+1} denote the image of S_{k+1} in $\hat{G}_k$. Clearly, s_{k+1} is a vertex in $\hat{G}_k[S_k]$. Let $T_k \in \mathcal{T}$ be a spanning tree constructed from T_{k-1} by (45). It follows that $T_k \cap S_i = T \cap S_i$ for $i = k+1, \ldots, r$ and $T_k \cap S = T \cap S$ for $S \in \mathcal{P} \backslash \mathcal{S}$. Let $\hat{G}_{k+1}[T_k]$ denote the edge-induced subgraph of $\hat{G}_{k+1}$ on the common edges of $\hat{G}_{k+1}$ and T_k. Lemma 22 implies that $\hat{G}_{k+1}[T_k]$ is a spanning tree of $\hat{G}_{k+1}$. In particular, $\hat{G}_{k+1}[T_k \cap S_{k+1}]$ is a spanning tree of $\hat{G}_{k+1}[S_{k+1}]$. To construct T_{k+1} from T_k, we distinguish two cases based on whether S_{k+1} is a parent of S_k.

First assume that S_{k+1} is a parent of S_k. Then there exist edges from S_k incident to two distinct vertices $u_1, u_2 \in V(\hat{G}_{k+1}[S_{k+1}])$ in $\hat{G}_{k+1}$. To construct T_{k+1} from T_k by (45), we concentrate on $\hat{G}_{k+1}$ and further distinguish two cases on edges in $T_k \cap S_k$.

- Edges from $T_k \cap S_k$ are only incident to one vertex, say u_1, of $\hat{G}_{k+1}[S_{k+1}]$ (cf., left graph in Fig. 3). Then there exists an edge $e'_k \in S_k$ incident to u_2. Since $\hat{G}_{k+1}[T_k]$ is a spanning tree of $\hat{G}_{k+1}$, adding e'_k to T_k creates a cycle involving edges in $\hat{G}_{k+1}[T_k \cap S_{k+1}]$. Remove an edge e_{k+1} from $\hat{G}_{k+1}[T_k \cap S_{k+1}]$ to break the cycle and denote the new tree by T_{k+1}. Thus we have $T_{k+1} = T_k + e'_k - e_{k+1}$.
- Edges from $T_k \cap S_k$ are incident to more than one vertices in $\hat{G}_{k+1}[S_{k+1}]$ (cf., middle graph in Fig. 3). For $i = 1, 2$, let $f_i \in T_k \cap S_k$ be an edge incident to $u_i \in V(\hat{G}_{k+1}[S_{k+1}])$, and v_i be the other endpoint of f_i. Since $\hat{G}_{k+1}[T_k \cap S_{k+1}]$ is a tree, v_1 and v_2 are distinct vertices in $\hat{G}_{k+1}[T_k]$. For $i = 1, 2$, let U_i be the set of vertices in $\hat{G}_{k+1}[S_k]$ that are connected to u_i with edges in T_k. Clearly, $v_1 \in U_1$ and $v_2 \in U_2$. Now consider $\hat{G}_k$. Since $\hat{G}_{k+1}[T_k]$ is a spanning tree of $\hat{G}_{k+1}$, $\hat{G}_k[T_k \cap S_k]$ is a spanning tree of $\hat{G}_k[S_k]$. Notice that $\hat{G}_k[T_k]$ is a spanning tree of $\hat{G}_k[\cup_{i=1}^k S_i]$. It follows that $U_1 \cup U_2 \cup \{s_{k+1}\} = V(\hat{G}_k[S_k])$. Notice that $\hat{G}_k[S_k]$ is a minimal densest subgraph of $\hat{G}_k$. By Lemma 8, there is a crossing edge $e'_k \in S_k$ between U_1 and U_2 in $\hat{G}_k[S_k]$. Since $\hat{G}_{k+1}[T_k]$ is a spanning tree of $\hat{G}_{k+1}$, adding e'_k to T_k creates a cycle involving edges in $\hat{G}_{k+1}[T_k \cap S_{k+1}]$. Remove an edge e_{k+1} from $\hat{G}_{k+1}[T_k \cap S_{k+1}]$ to break the cycle and denote the new tree by T_{k+1}. Thus we have $T_{k+1} = T_k + e'_k - e_{k+1}$.

Now assume that S_{k+1} is not a parent of S_k where $k \geq 1$ (cf., right graph in Fig. 3). Then there exists another ancestor $S'_k \in \mathcal{P}_{l-k}$ of S_0 such that $\hat{G}^{(l-k-1)}[S_k]$ and $\hat{G}^{(l-k-1)}[S'_k]$ share no common vertex but $\hat{G}^{(l-k)}[S_k]$ and $\hat{G}^{(l-k)}[S'_k]$ share a common vertex s_{k+1} which is the image of S_{k+1}. Now consider $\hat{G}_{k+1}$. Notice that $\hat{G}_{k+1}[S_{k+1}]$, $\hat{G}_{k+1}[S_k]$ and $\hat{G}_{k+1}[S'_k]$ are all minimal densest subgraphs in $\hat{G}_{k+1}$. Moreover, $\hat{G}_{k+1}[S_{k+1}]$ shares a common vertex u with $\hat{G}_{k+1}[S_k]$ and shares a common vertex u' with $\hat{G}_{k+1}[S'_k]$ respectively. Clearly, $u \neq u'$. Since $|T_k \cap S_k| = |T_0 \cap S_k| - 1 = n(S_k) - 2$, $\hat{G}_{k+1}[T_k \cap S_k]$ is not connected. Notice that $\hat{G}_{k+1}[T_k]$ is a spanning tree of $\hat{G}_{k+1}$. Let U and U' be the set of vertices in $\hat{G}_{k+1}[S_k]$ that are connected to u and u' respectively in $\hat{G}_{k+1}[T_k]$. Hence U and U' form a nontrivial bipartition of $V(\hat{G}_{k+1}[S_k])$.

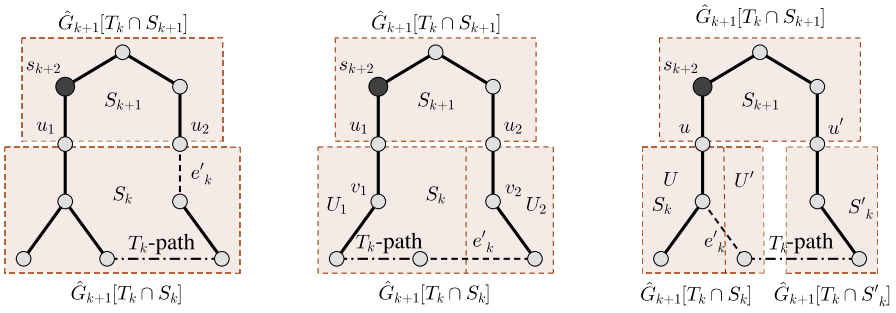

Fig. 3 The dashed line denotes the edge e'_k added to T_k. The dash-dotted line denotes the path in T_k avoiding edges in S_k

Then there is a crossing edge $e'_k \in S_k$ between U and U' in $V(\hat{G}_{k+1}[S_k])$. Since $\hat{G}_{k+1}[T_k]$ is a spanning tree of $\hat{G}_{k+1}$, adding e'_k to T_k creates a cycle involving edges in $\hat{G}_{k+1}[T_k \cap S_{k+1}]$. Remove an edge e_{k+1} from $\hat{G}_{k+1}[T_k \cap S_{k+1}]$ to break the cycle and denote the new tree by T_{k+1}. Thus we have $T_{k+1} = T_k + e'_k - e_{k+1}$. □

Lemma 25 *Let $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$ be a vector satisfying (32)–(34) and $\boldsymbol{y} \in \mathbb{R}^{\mathcal{E}}$ be the vector defined from $\boldsymbol{x}$. If $\boldsymbol{x}$ satisfies (31), then $\boldsymbol{y}$ satisfies (36).*

Proof Let $T \in \mathcal{T}_0$ be a spanning tree. Lemma 24 implies that there exists a spanning tree $T' \in \mathcal{T}\backslash\mathcal{T}_0$ such that $|T' \cap P| = |T \cap P| + 1$, $|T' \cap Q| = |T \cap Q| - 1$, and $|T' \cap R| = |T \cap R|$ for any $R \in \mathcal{P}\backslash\{P, Q\}$. It follows that

$$\begin{aligned} x(T') &= |T' \cap P| \cdot y_P + |T' \cap Q| \cdot y_Q + \sum_{R \in \mathcal{P}\backslash\{P,Q\}} |T' \cap R| \cdot y_R \\ &= (|T \cap P| + 1) \cdot y_P + (|T \cap Q| - 1) \cdot y_Q + \sum_{R \in \mathcal{P}\backslash\{P,Q\}} |T \cap R| \cdot y_R \\ &= x(T) + y_P - y_Q. \end{aligned}$$

Since $T \in \mathcal{T}_0$ and $T' \in \mathcal{T}\backslash\mathcal{T}_0$, we have $x(T) = 1$ and $x(T') + \epsilon \leq 1$. Hence $y_P + \epsilon \leq y_Q$ follows. □

Now we show that if $\boldsymbol{y} \in \mathbb{R}^{\mathcal{E}}$ is a feasible solution of LP''_2, then its associated vector $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$ satisfies (31) and hence is a feasible solution of LP'_2. Our proof is based on the idea that a spanning tree in $\mathcal{T}_0$ can be constructed from any spanning tree outside $\mathcal{T}_0$ by repeatedly performing edge exchanges between a prime set and the non-prime set or between a prime set and another prime set of higher level. To this end, we need the following lemma.

Lemma 26 *Let T be a spanning tree in $\mathcal{T}\backslash\mathcal{T}_0$. Among prime sets that violate (40) with T, let P be a prime set of minimum level. Then we have $|T \cap P| < n(P) - 1$.*

Proof Let T be a spanning tree in $\mathcal{T}\backslash\mathcal{T}_0$ and k be the minimum level of prime sets that violate (40) with T. For any prime set $P \in \mathcal{P}_k$ that violates (40) with T, we show

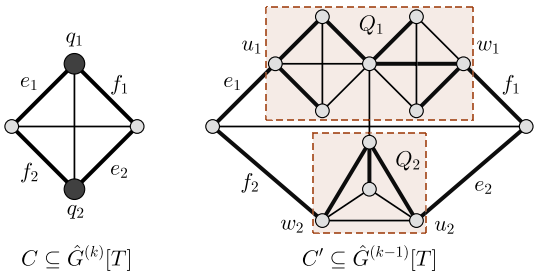

Fig. 4 A cycle C' in $\hat{G}^{(k-1)}[T]$ is constructed from a cycle C in $\hat{G}^{(k)}[T]$

that $|T \cap P| > n(P) - 1$ is absurd. Assume to the contrary that $|T \cap P| > n(P) - 1$. It follows that $\hat{G}^{(k)}[T \cap P]$ contains a cycle consisting of edges from T. We apply induction on k to show that a cycle in $\hat{G}^{(k)}[T]$ implies a cycle in T, which is absurd.

It is trivial for $k = 0$, since $\hat{G}^{(0)}[T] = T$. Now assume that $k \geq 1$ and that all prime sets of level less than k satisfies (40) with T. Hence $\hat{G}^{(k-1)}[T]$ is a tree. Let C be a cycle in $\hat{G}^{(k)}[T]$ (cf. left graph in Fig. 4). It follows that C contains images of prime sets of level $k-1$. Let $q_1, \ldots, q_s$ be the images of prime sets of level $k-1$ in C which appear in a clockwise order along C. Denote by e_i and f_i the two edges incident to q_i in C and denote by Q_i the union of prime sets of level $k-1$ with image q_i, for $i = 1, \ldots, s$. Hence e_i and f_i are incident to two distinct vertices u_i and w_i in $\hat{G}^{(k-1)}[Q_i]$ respectively. By assumption, $|T \cap Q| = n(Q) - 1$ for any $Q \in \mathcal{P}_{k-1}$. It follows that $\hat{G}^{(k-1)}[T \cap Q_i]$ is a tree. Let p_i denote the unique u_i-w_i path in $\hat{G}^{(k-1)}[T \cap Q_i]$. Now inserting the u_i-w_i path p_i between e_i and f_i in C for $i = 1, \ldots, s$ creates a cycle C' in $\hat{G}^{(k-1)}[T]$ (cf. right graph in Fig. 4). However, this contradicts the acyclicity of $\hat{G}^{(k-1)}[T]$. □

Lemma 27 *Let $\boldsymbol{x} \in \mathcal{C}(\Gamma_G)$ be a vector satisfying (32)–(34) and $\boldsymbol{y} \in \mathbb{R}^{\mathcal{E}}$ be the vector defined from $\boldsymbol{x}$. If $\boldsymbol{y}$ satisfies (36), then $\boldsymbol{x}$ satisfies (31).*

Proof Let $T \in \mathcal{T} \backslash \mathcal{T}_0$. Among all prime sets that violates (40) with T, let $P \in \mathcal{P}_k$ be a prime set of minimum level, where $k \geq 0$. Since every prime set of level less than k satisfies (40) with T, $\hat{G}^{(k)}[T]$ is a spanning tree of $\hat{G}^{(k)}$.

Lemma 26 implies that $|T \cap P| < n(P) - 1$. It follows that $\hat{G}^{(k)}[T \cap P]$ is not connected and there exists an edge e' in $P \backslash T$ that joins two components of $\hat{G}^{(k)}[T \cap P]$. Moreover, e' joins two non-adjacent vertices of $\hat{G}^{(k)}[T]$. Hence adding e' to $\hat{G}^{(k)}[T]$ creates a cycle C. As we shall see, C involves edges either from the non-prime set E_0 or from a prime set of level greater than k. Now we show that a new spanning tree $T' \in \mathcal{T}$ can be constructed from T such that $x(T) \leq x(T') - \epsilon$ with an edge exchange operation. We distinguish two cases.

- $C \cap E_0 \neq \emptyset$. Remove an edge e from $C \cap E_0$ to break the cycle C and denote the new tree by T'. Hence $T' = T - e + e'$ where $e \in C \cap E_0$ and $e' \in P \backslash T$. Since $x_e = 0$ and $x_{e'} \geq \epsilon$, it follows that $x(T) = x(T') + x_e - x_{e'} \leq x(T') - \epsilon$.
- $C \cap E_0 = \emptyset$. It follows that C is a cycle in a component of $\hat{G}^{(k)} - E_0$. Lemmas 12 and 16 imply that every component of $\hat{G}^{(k)} - E_0$ is a densest subgraph of $\hat{G}^{(k)}$. By Lemma 10, C involves edges from prime sets of level higher than k,

since otherwise there are minimal densest subgraphs in $\hat{G}^{(k)}$ which are pairwise connected along the cycle C and the number of common vertices violates Lemma 10. Let $Q \in \mathcal{P}_l$ where $l > k$ be a prime set of the largest level that intersects C. We claim that $\hat{G}^{(l)}[Q \cap C]$ is a cycle. To see this, consider $\hat{G}^{(l)}[C]$ which is the edge-induced subgraph of $\hat{G}^{(l)}$ on the common edges of $\hat{G}^{(l)}$ and $C \subseteq \hat{G}^{(k)}$. If there exists a cycle C' in $\hat{G}^{(l)}[C]$ involving more than one prime sets of level l, then their defining minimal densest subgraphs are pairwise connected along the cycle C' and the number of common vertices violates Lemma 10. Hence the claim follows. It follows that $Q \prec P$. To see this, we apply induction on $l-k$. If $l = k+1$, then two edges in $C \cap Q$ become incident (or share one more common vertex) at the image v_P of P in $\hat{G}^{(l)}$. Thus P is a parent of Q and $Q \prec P$ follows. Now assume that $l > k+1$. Let C' be the cycle in $\hat{G}^{(k+1)}$ consisting of edges from C and involving edges in Q. For any prime set $R \in \mathcal{P}_{k+1}$ that intersects C', $R \prec P$ implies $Q \prec P$ inductively. Hence assume that P is not a parent of any prime set of level $k+1$ that intersects C'. Let v_P be the image of P in $\hat{G}^{(k+1)}$. Then v_P is a vertex in C' which concatenates two minimal densest subgraphs of $\hat{G}^{(k+1)}$ involving edges of C'. There exists a prime set $R \in \mathcal{P}_r$ where $k+1 \prec r \leq l$ such that two edges in $R \cap C'$ become incident (or share one more common vertex) in $\hat{G}^{(r)}$, and a cycle C'' in $\hat{G}^{(r)}$ consisting of edges from C' and involving edges from Q and R. Further assume that the prime set R introduced above is of minimum level. Hence P is a parent of R. If $Q = R$, then $Q \prec P$ follows directly. Otherwise, $Q \prec R$ follows inductively. Thus in either case, we have $Q \prec P$. Remove an edge e in $C \cap Q$ to break the cycle C and denote the new tree by T'. Then $T' = T - e + e'$ where $e \in C \cap Q$ and $e' \in P \backslash T$. Since $Q \prec P$, we have $y_Q + \epsilon \leq y_P$ which implies $x_e + \epsilon \leq x_{e'}$. It follows that $x(T) = x(T') + x_e - x_{e'} \leq x(T') - \epsilon$.

Hence a new spanning tree T' can be constructed from T such that $x(T) \leq x(T') - \epsilon$ with an edge exchange operation. Now we consider T'. If $T' \notin \mathcal{T}_0$, then among all prime sets that violates (40) associated with T', let P' be one of minimum level. By Lemma 26, $|T' \cap P'| < n(P') - 1$ follows again. Denote T by T_0 and T' by T_1. Then repeating the process that constructs T_1 from T_0 yields a sequence $T_1, \ldots, T_k \in \mathcal{T}$ of spanning trees until the last tree T_k appears in $\mathcal{T}_0$. And we have $x(T_i) \leq x(T_{i+1}) - \epsilon$ for $i = 1, \ldots, k-1$. This sequence ends properly because each time an edge exchange operation is performed between a prime set and the non-prime set or between a prime set and another prime set of higher levels. This sequence ends with a spanning tree in $\mathcal{T}_0$ because each time an edge is added to the prime set of minimum level that violates (40). Finally, $T_k \in \mathcal{T}_0$ implies that

$$x(T) = x(T_0) \leq x(T_k) - k\epsilon = 1 - k\epsilon \leq 1 - \epsilon, \tag{47}$$

where the last inequality follows from the fact that $\epsilon > 0$ □

6.3 A combinatorial algorithm for LP_2''

The following lemma reveals how to solve LP_2''.

Lemma 28 *Let $(\boldsymbol{y}^*, \epsilon^*)$ be an optimal solution of LP_2''. Then for each prime set $P \in \mathcal{P}$, either (36) or (38) is tight for $(\boldsymbol{y}^*, \epsilon^*)$.*

Proof Assume to the contrary that neither (36) nor (38) is tight for $P_0 \in \mathcal{P}$. For a constant $\delta > 0$ small enough, define $\boldsymbol{y}^\star$ by $y^\star_{P_0} = y^*_{P_0} - \delta$ and $y^\star_P = y^*_P$ for any $P \in \mathcal{P}\backslash\{P_0\}$. Then $(\boldsymbol{y}^\star, \epsilon^*)$ satisfies (36), (38) and (39), but $\sum_{P\in\mathcal{P}}[n(P)-1]y^\star_P < 1$. Hence $(\boldsymbol{y}^\star, \epsilon^*)$ can be scaled up with a constant $\theta > 1$ such that $(\theta\boldsymbol{y}^\star, \theta\epsilon^*)$ satisfies (36)–(39). However, this contradicts the optimality of $(\boldsymbol{y}^*, \epsilon^*)$. □

Based on the lemma above, we derive a combinatorial algorithm for solving LP_2''.

Algorithm A combinatorial algorithm for LP_2''

$k = 0$
while $\mathcal{P} \neq \emptyset$ **do**
  $k \leftarrow k + 1$
  $\mathcal{P}_{\min} \leftarrow$ the set of minimal prime sets in $(\mathcal{P}, \prec)$
  $y_P \leftarrow k\epsilon$ for any $P \in \mathcal{P}_{\min}$
  $\mathcal{P} \leftarrow \mathcal{P}\backslash\mathcal{P}_{\min}$
end while
Since $y_P = k_P\epsilon$ where k_P is an integer for any $P \in \mathcal{P}$, solving ϵ in (37) gives the unique optimal solution of LP_2''.

The algorithm above implies that LP_2'' has a unique optimal solution, which yields the nucleolus of Γ_G. Now we are ready to present our main result.

Theorem 29 *Let $\Gamma_G = (N, \gamma)$ be an arboricity game with a nonempty core. The nucleolus of Γ_G can be computed in $O(n^4 m \log \frac{n^2}{m})$.*

Proof The prime partition can be computed in $O(n^4 m \log \frac{n^2}{m})$. The ancestor relation of prime sets can be determined in $O(n^2 m)$. The algorithm above takes $O(n)$ iterations and each iteration requires $O(n^2)$ time to determine the minimal prime sets in the remaining partially ordered set. Hence the algorithm above ends in $O(n^3)$. Notice that the prime partition computation dominates the computing time of all other parts. Thus the nucleolus can be computed in $O(n^4 m \log \frac{n^2}{m})$. □

7 Concluding remarks

This paper provides an efficient algorithm for computing the nucleolus of arboricity games when the core is not empty. Notice that a variant of the arboricity game arises when the cost of each coalition is defined by fractional arboricity instead of arboricity. Despite a new cost function, our algorithm for computing the nucleolus remains valid since the variant always has a nonempty core.

This paper also offers a graph decomposition built on the densest subgraph lattice. The prime partition decomposes the edge set of a graph into a non-prime set and a number of prime sets, where prime sets correspond to minimal densest minors. Notice that the non-prime set can be further decomposed following the same procedure for defining prime sets. Therefore, the prime partition indeed provides a hierarchical graph decomposition analogous to the celebrated core decomposition.

Acknowledgements We are grateful to all reviewers for their comments and suggests which greatly improved the presentation of this work.

References

1. Auman, R., Maschler, M.: Game theoretic analysis of a bankruptcy problem from the Talmud. J. Econ. Theory **36**(2), 195–213 (1985)
2. Aziz, H., Lachish, O., Paterson, M., Savani, R.: Wiretapping a hidden network. In: Proceedings of 5th International Workshop on Internet and Network Economics—WINE'09, pp. 438–446 (2009)
3. Baïou, M., Barahona, F.: An algorithm to compute the nucleolus of shortest path games. Algorithmica **81**, 3099–3113 (2019)
4. Baïou, M., Barahona, F.: Network strength games: the core and the nucleolus. Math. Program. **180**, 117–136 (2020)
5. Biró, P., Kern, W., Paulusma, D.: Computing solutions for matching games. Int. J. Game Theory **41**(1), 75–90 (2012)
6. Brânzei, R., Iñarra, E., Tijs, S., Zarzuelo, J.M.: A simple algorithm for the nucleolus of airport profit games. Int. J. Game Theory **34**(2), 259–272 (2006)
7. Catlin, P.A., Grossman, J.W., Hobbs, A.M., Lai, H.-J.: Fractional arboricity, strength, and principal partitions in graphs and matroids. Disc. Appl. Math. **40**(3), 285–302 (1992)
8. Chen, G., Jing, G., Zang, W.: Proof of the Goldberg-Seymour conjecture on edge-colorings of multigraphs. arXiv: 1901.10316 (2019)
9. Chen, N., Lu, P., Zhang, H.: Computing the nucleolus of matching, cover and clique games. In: Proceedings of 26th AAAI Conference on Artificial Intelligence—AAAI'12, pp. 1319–1325 (2012)
10. Deng, X., Fang, Q., Sun, X.: Finding nucleolus of flow game. In: Proceedings of 17th ACM-SIAM Symposium on Discrete Algorithms—SODA'06, pp. 124–131 (2006)
11. Deng, X., Ibaraki, T., Nagamochi, H.: Algorithmic aspects of the core of combinatorial optimization games. Math. Oper. Res. **24**(3), 751–766 (1999)
12. Edmonds, J.: Minimum partition of a matroid into independent subsets. J. Res. Nat. Bur. Stan. **69**, 67–72 (1965)
13. Elkind, E., Goldberg, L.A., Goldberg, P., Wooldridge, M.: Computational complexity of weighted threshold games. In: Proceedings of 22nd AAAI Conference on Artificial Intelligence—AAAI'07, pp. 718–723 (2007)
14. Elkind ,E., Pasechnik, D.: Computing the nucleolus of weighted voting games. In: Proceedings of 20th ACM-SIAM Symposium on Discrete Algorithms—SODA'09, pp. 327–335 (2009)
15. Faigle, U., Kern, W., Kuipers, J.: Computing the nucleolus of min-cost spanning tree games is NP-hard. Int. J. Game Theory **27**(3), 443–450 (1998)
16. Fang, Q., Zhu, S., Cai, M., Deng, X.: On computational complexity of membership test in flow games and linear production games. Int. J. Game Theory **31**(1), 39–45 (2002)

17. Gabow, H.N.: Algorithms for graphic polymatroids and parametric $\bar{s}$-sets. J. Algo. **26**(1), 48–86 (1998)
18. Granot, D., Maschler, M., Owen, G., Zhu, W.R.: The kernel/nucleolus of a standard tree game. Int. J. Game Theory **25**(2), 219–244 (1996)
19. Kern, W., Paulusma, D.: Matching games: the least core and the nucleolus. Math. Oper. Res. **28**(2), 294–308 (2003)
20. Könemann, J., Pashkovich, K., Toth, J.: Computing the nucleolus of weighted cooperative matching games in polynomial time. Math. Program. **183**, 555–581 (2020)
21. Könemann, J., Toth, J., Zhou, F.: On the complexity of nucleolus computation for bipartite b-matching games. In: Proceedings of 14th International Symposium on Algorithmic Game Theory—SAGT'21, pp. 171–185 (2021)
22. Kopelowitz, A.: Computation of the kernels of simple games and the nucleolus of n-person games. Technical Report RM-31, Hebrew University of Jerusalem (1967)
23. Lau, L.C., Ravi, R., Singh, M.: Iterative Methods in Combinatorial Optimization. Cambridge University Press, Cambridge (2011)
24. Maschler, M., Peleg, B., Shapley, L.S.: Geometric properties of the kernel, nucleolus, and related solution concepts. Math. Oper. Res. **4**(4), 303–338 (1979)
25. Megiddo, N.: Computational complexity of the game theory approach to cost allocation for a tree. Math. Oper. Res. **3**(3), 189–196 (1978)
26. Nash-Williams, C.S.J.A.: Decomposition of finite graphs into forests. J. London Math. Soc. **s1-39**(1), 12 (1964)
27. Scheinerman, E.R., Ullman, D.H.: Fractional Graph Theory: A Rational Approach to the Theory of Graphs. Wiley (1997)
28. Schmeidler, D.: The nucleolus of a characteristic function game. SIAM J. Appl. Math. **17**(6), 1163–1170 (1969)
29. Schrijver, A.: Theory of Linear and Integer Programming. Wiley (1986)
30. Schrijver, A.: Combinatorial Optimization—Polyhedra and Efficiency. Springer (2003)
31. Seidman, S.B.: Network structure and minimum degree. Social Netw. **5**(3), 269–287 (1983)
32. Solymosi, T., Raghavan, T.E.: An algorithm for finding the nucleolus of assignment games. Int. J. Game Theory **23**(2), 119–143 (1994)
33. Tatti, N.: Density-friendly graph decomposition. ACM Trans. Knowl. Discov. Data **13**(5), 1–29 (2019)
34. Tatti, N., Gionis, A.: Density-friendly graph decomposition. In: Proceedings of 24th International Conference on World Wide Web—WWW'15, pp. 1089–1099 (2015)
35. Toko Worou, B.M., Galtier, J.: Fast approximation for computing the fractional arboricity and extraction of communities of a graph. Disc. Appl. Math. 179–195 (2016)