Mathematical Programming (2020) 180:117–136
https://doi.org/10.1007/s10107-018-1348-3

FULL LENGTH PAPER
Series A

Network strength games: the core and the nucleolus

Mourad Baïou[1] · Francisco Barahona[2]

Received: 17 October 2017 / Accepted: 29 October 2018 / Published online: 9 November 2018

Abstract
The maximum number of edge-disjoint spanning trees in a network has been used as a measure of the strength of a network. It gives the number of disjoint ways that the network can be fully connected. This suggests a game theoretic analysis that shows the relative importance of the different links to form a strong network. We introduce the *Network strength game* as a cooperative game defined on a graph $G = (V, E)$. The player set is the edge-set E and the value of a coalition $S \subseteq E$ is the maximum number of disjoint spanning trees included in S. We study the core of this game, and we give a polynomial combinatorial algorithm to compute the nucleolus when the core is non-empty.

Keywords Network strength game · Strength of a network · Cooperative games · Nucleolus

Mathematics Subject Classification 91A12 · 91A43 · 91A46

1 Introduction

The maximum number of edge-disjoint spanning trees has been used as a measure of the strength of a communication network, see [1,6,18,19]. This is the number of independent ways that the network can be fully connected. This has motivated a game theoretic approach to determine the relative importance of the different links. Thus the *Network strength game* is a cooperative game defined on a graph $G = (V, E)$. The player set is E, and the value of a coalition $S \subseteq E$ is the maximum number of disjoint spanning trees included in S. This type of analysis gives the relative importance of

✉ Francisco Barahona
barahon@us.ibm.com

Mourad Baïou
baiou@isima.fr

1 CNRS, and Université Clermont II, Campus des cézeaux, BP 125, 63173 Aubière cedex, France

2 IBM T. J. Watson Research Center, Yorktown Heights, NY 10589, USA

the different links to form a strong network. We study the core of this game, and we give a polynomial combinatorial algorithm to compute the nucleolus when the core is non-empty. The components of the nucleolus reflect the relevance of the different links.

Games with some similarities with the one studied here are Spanning Connectivity Games [1,2] and Simple Flow Games [20]. Algorithms for computing the nucleolus of several combinatorial games have been found, see [1,2] for Spanning Connectivity Games and [8,25] for Simple Flow Games. Here we use the notion of *prime sets* introduced in [1,2], however our mathematical justification is completely different.

See also [23] for a Cost Allocation Game on trees. Assignment Games have been treated in [28], Standard Tree Games have been studied in [17], see [21] for Matching Games, see [10] for Graph Games, and [3] for Shortest Path Games. On the negative side it is NP-hard to compute the nucleolus of Minimum Cost Spanning Tree Games, see [12]. For other aspects of combinatorial games see [7] and the references therein.

This paper is organized as follows. In Sect. 2 we give definitions and some notation. In Sect. 3 we review results on packing spanning trees. In Sect. 4 we study the core of this game. In Sect. 5 we give properties of the core elements that are needed to compute the nucleolus. Sect. 6 is devoted to the linear programs needed to compute the nucleolus. In Sect. 7 we give a combinatorial algorithm to find the nucleolus.

2 Preliminaries and definitions

Here we give some basic definitions and notation to be used in the subsequent sections. Given an undirected graph $G = (V, E)$, we assume that each player owns an edge in E, thus we identify players with edges. If $S \subseteq E$, we denote by $v(S)$ the maximum number of (edge-)disjoint spanning trees included in S, this is the *strength* of the subgraph spanned by S, cf. [19], and it is the *value* of the coalition S. Thus the *Network Strength Game* is the cooperative game (E, v).

For a vector $y : E \to \mathbb{R}$, and $S \subseteq E$, we use $y(S)$ to denote $\sum_{e \in S} y(e)$. A vector $x : E \to \mathbb{R}$ is called an *allocation* if $x(E) = v(E)$. Here $x(e)$ represents the amount paid to player e. The *core* is a concept introduced in [27], it is based on the following stability condition: No subgroup of players does better if they break away from the joint decision of all players to form their own coalition. Thus an allocation x is in the core if $x(S) \geq v(S)$ for each coalition $S \subseteq E$. We denote the core by $\mathscr{C}$, this is the polytope defined below:

$$\mathscr{C} = \left\{ x \in \mathbb{R}^E \mid x(E) = v(E),\ x(S) \geq v(S) \text{ for } S \subset E \right\}.$$

For a coalition S and a vector x, their *excess* is $e(x, S) = x(S) - v(S)$. Let $\theta(x)$ be the $2^{|E|} - 2$ dimensional vector whose components are the excesses $e(x, S)$, $\emptyset \neq S \neq E$, arranged in nondecreasing order. The *nucleolus*, denoted by η, is the (unique) allocation that lexicographically maximizes the vector $\theta(x)$. This is a concept introduced in [26], and it is "the best" allocation, in the sense just described.

The nucleolus can be computed with a sequence of linear programs as follows, cf. [22]. First solve

$$\max \epsilon \tag{1}$$
$$x(S) \geq v(S) + \epsilon, \quad \forall S \subset E \tag{2}$$
$$x(E) = v(E). \tag{3}$$

Let ϵ_1 be the optimal value of this, and let $P_1(\epsilon_1)$ be the polytope defined above, with $\epsilon = \epsilon_1$, i.e., $P_1(\epsilon_1)$ is the set of optimal solutions of the linear program above. For a polytope $P \subset \mathbb{R}^E$ let

$$\mathcal{F}(P) = \{S \subseteq E \mid x(S) = y(S), \quad \forall x, y \in P\}$$

denote the set of coalitions fixed for P. In general given ϵ_{r-1} we solve

$$\max \epsilon \tag{4}$$
$$x(S) \geq v(S) + \epsilon, \quad \forall S \notin \mathcal{F}(P_{r-1}(\epsilon_{r-1})) \tag{5}$$
$$x \in P_{r-1}(\epsilon_{r-1}). \tag{6}$$

We denote by ϵ_r the optimal value of this, and $P_r(\epsilon_r)$ the polytope above with $\epsilon = \epsilon_r$. We continue for $r = 2, \ldots, |E|$, or until $P_r(\epsilon_r)$ is a singleton. Notice that each time the dimension of $P_r(\epsilon_r)$ decreases by at least one, so it takes at most $|E|$ steps for $P_r(\epsilon_r)$ to be a singleton. Actually, this shows that the nucleolus is unique. As we shall see later, in our case only two linear programs are needed, and they can be solved in a combinatorial way.

We conclude this section with some extra notation. For a graph $G = (V, E)$, and a family $\{S_1, \ldots, S_p\}$ of disjoint subsets of V, we denote by $\delta(S_1, \ldots, S_p)$ the set of edges with both endpoints in different sets $\{S_i\}$. For $S \subset V$ we use $\delta(S)$ to denote the set of edges with exactly one endnode in S. If we *contract* an edge e, we keep working with the graph obtained by identifying the endnodes of e, we remove loops, and this could create parallel edges that are kept. If $\{S_1, \ldots, S_p\}$ is a partition of V, we assume that $p \geq 2$. We use n to denote $|V|$ and m to denote $|E|$. For $F \subseteq E$ the *incidence vector* of F, $x^F : E \to \mathbb{R}$, is defined by $x^F(e) = 1$ if $e \in F$, and $x^F(e) = 0$ otherwise.

3 Network strength

Here we review some results of Polyhedral Combinatorics that will be used in the following sections. Given a graph $G = (V, E)$ and a capacity function $w : E \to \mathbb{Z}_+$, a solution of the integer program

$$\max \sum_T y_T \tag{7}$$
$$\sum \{y_T \mid e \in T\} \leq w(e) \text{ for each edge } e \in E \tag{8}$$
$$y_T \geq 0, \text{ integer valued, for each spanning tree } T, \tag{9}$$

is called an *integral packing of spanning trees*. A min–max relation for (7)–(9) was given by Tutte [29] and Nash-Williams [24], as follows.

Theorem 1 *A graph $G = (V, E)$ has k disjoint spanning trees if and only if for every partition $\{S_1, \ldots, S_p\}$ of V,*

$$|\delta(S_1, \ldots, S_p)| \geq k(p-1).$$

Notice that if an edge e has an integer nonnegative capacity $w(e)$, we can make $w(e)$ parallel copies of the edge e, then this theorem implies that the value of the maximum in (7)–(9) is

$$\min \left\lfloor \frac{w(\delta(S_1, \ldots, S_p))}{p-1} \right\rfloor, \tag{10}$$

where the minimum is taken among all partitions of V, and $\lfloor z \rfloor$ denotes the largest integer less than or equal to z. Polynomial combinatorial algorithms for finding a maximum integral (and fractional) packing of spanning trees have been given in [4,13].

If we relax the integrality condition in (9), and we only ask for variables y to be nonnegative, we obtain a linear program. It also follows from Theorem 1 that the optimal value is

$$\min \frac{w(\delta(S_1, \ldots, S_p))}{p-1}, \tag{11}$$

where the minimum is taken among all partitions of V. The quotient in (11) is the total weight of the edge-set $\delta(S_1, \ldots, S_p)$, divided by the number of new connected components obtained by removing the edges in $\delta(S_1, \ldots, S_p)$. The value of this minimum was called the *strength of the network* in [6], see also [19]. The dual problem is

$$\min wx \tag{12}$$

$$x(T) \geq 1, \text{ for each spanning tree } T, \tag{13}$$

$$x \geq 0. \tag{14}$$

This implies that the extreme points of the polyhedron defined by (13)–(14) are among the vectors that are obtained as follows: for each partition $\{S_1, \ldots, S_p\}$ of V, take the incidence vector of $\delta(S_1, \ldots, S_p)$ and divide each component by $(p-1)$. However one should notice that not all vectors obtained in this way are extreme points. For instance, assume that the graph consists of just one tree, then the extreme points are obtained only when $p = 2$. All this leads to the theorem below that will be used in Sect. 4.

Theorem 2 *Assume that $w(e) > 0$ for each edge e, and let θ be the value of the minimum in (11). Then the set of optimal solutions of (12)–(14) is the convex hull of all vectors obtained as follows:*

– *Pick a partition $\{S_1, \ldots, S_p\}$ of V, with $w(\delta(S_1, \ldots, S_p)) = \theta(p-1)$.*

– *Let $\bar{y}$ be the incidence vector of $\delta(S_1, \ldots, S_p)$. Set*

$$\bar{x} = \frac{1}{p-1}\bar{y}.$$

Cunningham [6] gave an algorithm for finding the minimum in (11) and (10), it requires $O(nm)$ minimum cut problems. Later an algorithm that requires n applications of the preflow algorithm was given in [5].

Now consider problem (11) when $w(e) = 1$, for all $e \in E$. Set

$$\lambda(V) = \min \frac{|\delta(S_1, \ldots, S_p)|}{p-1}, \tag{15}$$

where the minimum is taken among all partitions of V. Here we call (15) *the strength problem* and we say that the graph *has strength* $\lambda(V)$. The following two lemmas give properties of the solutions of the strength problem that will be used in the following sections. In the remainder of this section we use λ instead of $\lambda(V)$.

Lemma 3 *If $\{S_1, \ldots, S_p\}$ is a solution of (15) and $\{T_1, \ldots, T_q\}$ is a partition of S_j, for some index j. Then*

$$|\delta(T_1, \ldots, T_q)| \geq \lambda(q-1).$$

Proof W.l.o.g. assume that $j = 1$. If $|\delta(T_1, \ldots, T_q)| < \lambda(q-1)$ then

$$|\delta(T_1, \ldots, T_q, S_2, \ldots, S_p)| < \lambda(p+q-2),$$

which contradicts the definition of λ. □

Lemma 4 *If $\{S_1, \ldots, S_p\}$ is a solution of (15) and*

$$|\delta(S_1, \ldots, S_l)| \geq \lambda(l-1),$$

for some index l, $1 < l < p$. Then the partition $\{\cup_{i=1}^{i=l} S_i, S_{l+1}, \ldots, S_p\}$ is also a solution of (15). In other words, we can combine $\{S_1, \ldots, S_l\}$ into one set and obtain a new solution of (15).

Proof Since $|\delta(S_1, \ldots, S_l)| \geq \lambda(l-1)$, we have $|\delta(\cup_{i=1}^{i=l} S_i, S_{l+1}, \ldots, S_p)| \leq \lambda(p-l)$. These two are equalities, otherwise $\{S_1, \ldots, S_p\}$ would not be a solution of (15).

Thus $|\delta(\cup_{i=1}^{i=l} S_i, S_{l+1}, \ldots, S_p)| = \lambda(p-l)$, and $\{\cup_{i=1}^{i=l} S_i, S_{l+1}, \ldots, S_p\}$ is a solution of (15). □

4 Properties of the core

Now we study some properties of the core. The core of other combinatorial optimization games has been studied in [9–12,14–16,23], and others. First we need the following lemma.

Lemma 5 *The core can be described as*

$$x(E) = v(E) \tag{16}$$
$$x(T) \geq 1, \quad \textit{for each spanning tree } T \subseteq E, \tag{17}$$
$$x(e) \geq 0, \quad \textit{for each edge } e \in E. \tag{18}$$

Proof For $S \subseteq E$, consider the inequality $x(S) \geq v(S)$. If S does not contain a spanning tree, then $x(S) \geq 0$ is implied by $x(e) \geq 0$, for each edge $e \in S$.

Now assume that $v(S) = q \geq 1$. Thus S contains q disjoint spanning trees $T_1, \ldots, T_q$. Then $x(S) \geq v(S)$ is implied by $x(T_i) \geq 1$, for $i = 1, \ldots, q$ and $x(e) \geq 0$ for $e \in S \setminus \cup_i T_i$. □

As seen in Sect. 3, we also have that $x(E) \geq f(E)$, where $f(E)$ is the value of a maximum fractional packing of spanning trees. Since $v(E)$ is the value of an integral packing, we have $f(E) \geq v(E)$. Thus the core is nonempty if and only if $f(E) = v(E)$. This and the results of Sect. 3 imply the following.

Theorem 6 *The core is nonempty if and only if*

$$\min \frac{|\delta(S_1, \ldots, S_p)|}{p-1} \tag{19}$$

is an integer number. This minimum is taken among all partitions $\{S_1, \ldots, S_p\}$ *of* V.

The theorem above also follows from Theorem 1 in [9] that was proved in a more general setting.

Corollary 7 *We can test whether the core is nonempty in polynomial time.*

Given a vector $\bar{x}$, testing if $\bar{x}$ satisfies inequalities (17) reduces to finding a minimum weight spanning tree problem. Thus we have the following.

Theorem 8 *Given a vector* $\bar{x}$*, we can decide in polynomial time if* $\bar{x}$ *belongs to the core, and if not, we can find a separating hyperplane.*

It follows from Theorem 2 that we can also describe the core as the convex hull of a set of vectors obtained as follows:

– Pick a partition $\{S_1, \ldots, S_p\}$ of V, with $|\delta(S_1, \ldots, S_p)| = \lambda(p-1)$.
– Let $\bar{y}$ be the incidence vector of $\delta(S_1, \ldots, S_p)$. Set

$$\bar{x} = \frac{1}{p-1}\bar{y}.$$

We say that the vector $\bar{x}$ defined above is *the vector associated with the partition* $\{S_1, \ldots, S_p\}$ of V. Thus any vector in the core is a convex combination of vectors associated with partitions. This property will be used when we study the nucleolus.

5 Prime sets

In the last section we have seen that any vector in the core is a convex combination of vectors associated with partitions, that are solutions of the strength problem. To compute the nucleolus we need to study some additional properties of these partitions. This is the purpose of this section.

A *prime set (of level zero)* $P \subseteq E$ is a minimal solution of the strength problem. Later (in Sect. 5.3) we introduce prime sets of level greater than zero, until then, we just write "prime set". If P is a prime set, then there is no solution Q of the strength problem so that $Q \subset P$, $Q \neq P$. Prime sets have been introduced in [1,2] for the Spanning Connectivity Game. There might be many prime sets, one can find one of them as follows. Given a solution of the strength problem, we pick an edge e in the solution, we contract it, and compute the strength of the new graph. If the value increases, then e should be included in the prime set, otherwise we keep e contracted. We repeat this for every other edge, the edges that do not remain contracted form a prime set. If $P = \delta(S_1, \ldots, S_t)$ is a prime set we define $\pi(P) = t$, this is the number of sets in the partition of V defining P. Later we shall see how to compute all prime sets.

5.1 A non-crossing property

A key property of prime sets is given in the following lemma.

Lemma 9 *If $P = \delta(S_1, \ldots, S_p)$ is a prime set, and $\delta(T_1, \ldots, T_q)$ is another solution of the strength problem, then either*

$$P \subseteq \delta(T_1, \ldots, T_q) \;\; or$$
$$P \cap \delta(T_1, \ldots, T_q) = \emptyset.$$

Proof Let λ be the value given by the strength problem. Suppose that there is an index j such that $T_j \cap S_{i_r} \neq \emptyset$, for $r = 1, \ldots, l$, and $T_j \subseteq \cup_{r=1}^{l} S_{i_r}$, with $1 < l < p$. Here a permutation of the indices of the sets $\{S_i\}$ might be needed. See Fig. 1a, where the the sets $\{S_i\}$ appear with solid lines, and the sets $\{T_i\}$ are depicted with dashed lines. Lemma 3 implies $|\delta(S_{i_1}, \ldots, S_{i_l})| \geq |\delta(S_{i_1} \cap T_j, \ldots, S_{i_l} \cap T_j)| \geq \lambda(l-1)$. Then Lemma 4 shows that we could combine the sets $\{S_{i_1}, \ldots, S_{i_l}\}$ into one set and obtain a new solution of the strength problem. This contradicts the minimality of the prime set. Thus each set T_j either intersects all sets $\{S_i\}$ or it intersects only one of them, i.e., it is contained in one set S_i.

Suppose that there are two indices j_1 and j_2 such that $T_{j_r} \cap S_i \neq \emptyset$, for $i = 1, \ldots, p$, for $r = 1, 2$. See Fig. 1b. Lemma 3 implies $|\delta(T_{j_r} \cap S_1, \ldots, T_{j_r} \cap S_p)| \geq \lambda(p-1)$, for $r = 1, 2$. We have a contradiction with $|\delta(S_1, \ldots, S_p)| = \lambda(p-1)$.

Thus there is at most one index j' such that $T_{j'} \cap S_i \neq \emptyset$, for $i = 1, \ldots, p$, and for every other index j, $T_j \subseteq S_{i(j)}$ for some index $i(j)$, i.e., T_j is included in some set S_i. If the index j' exists, Lemma 3 implies $|\delta(T_{j'} \cap S_1, \ldots, T_{j'} \cap S_p)| \geq \lambda(p-1)$, thus every edge in $\delta(S_1, \ldots, S_p)$ has both endnodes in $T_{j'}$. Therefore $\delta(S_1, \ldots, S_p) \cap$

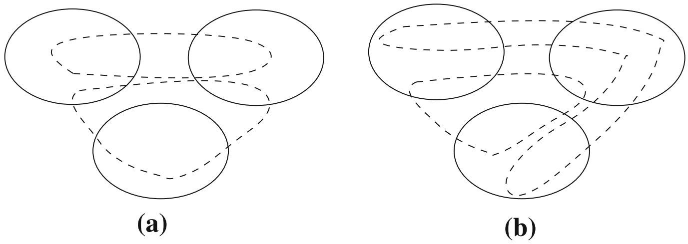

Fig. 1 Impossible configurations

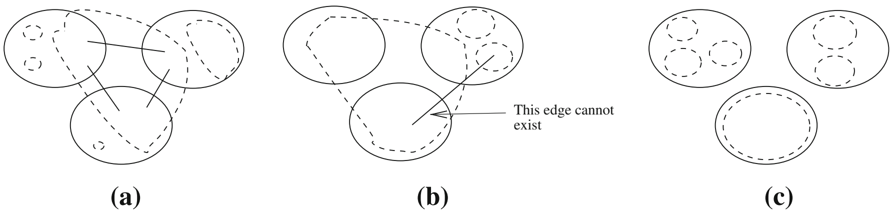

Fig. 2 Possible configurations. The sets $\{S_i\}$ appear with solid lines. The sets $\{T_j\}$ are drawn with dashed lines

$\delta(T_1, \ldots, T_q) = \emptyset$, see Fig. 2a. Otherwise if j' does not exist, $\delta(S_1, \ldots, S_p) \subseteq \delta(T_1, \ldots, T_q)$, see Fig. 2c. □

In Figs. 1 and 2 we depict a partition giving a prime set with solid lines, and a second partition that solves the strength problem with dashed lines. First we show configurations that cannot occur, then we show configurations that are possible.

Lemma 9 leads to an important property of vectors in the core as follows.

Corollary 10 *For any two edges e and f in a prime set P we have $\tilde{x}(e) = \tilde{x}(f)$, for any vector $\tilde{x} \in \mathscr{C}$.*

Proof Let $\bar{x}$ be a vector associated with a partition $\{T_1, \ldots, T_q\}$ that is a solution of the strength problem. Lemma 9 shows that either $P \subseteq \delta(T_1, \ldots, T_q)$ and in this case $\bar{x}(e) = \bar{x}(f) = 1/(q-1)$, or $P \cap \delta(T_1, \ldots, T_q) = \emptyset$ and $\bar{x}(e) = \bar{x}(f) = 0$. Since $\tilde{x} \in \mathscr{C}$ is a convex combination of vectors associated with partitions, we have $\tilde{x}(e) = \tilde{x}(f)$. □

5.2 Finding all prime sets

Let λ be the strength of the graph. Given a prime set P, Lemma 9 shows that we can contract all edges in P, and if the new graph has still strength λ, we can look for a second prime set P'. Then we contract the edges in P and P' and if the resulting graph has still strength λ, we continue. This is a way to generate all prime sets. In Fig. 3a we use dashed lines to draw the partition of V that gives the first prime set, then we used solid lines to show the partition that gives the second prime set. In Fig. 3b we show these two prime sets. In Fig. 4 we have an example. Notice that the graph has parallel edges. Here $\lambda = 3$. The prime set P_1 appears with thin dashed lines. The prime set

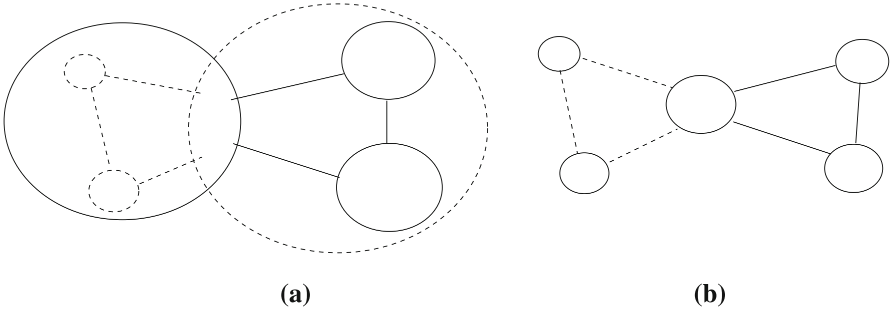

Fig. 3 Two prime sets

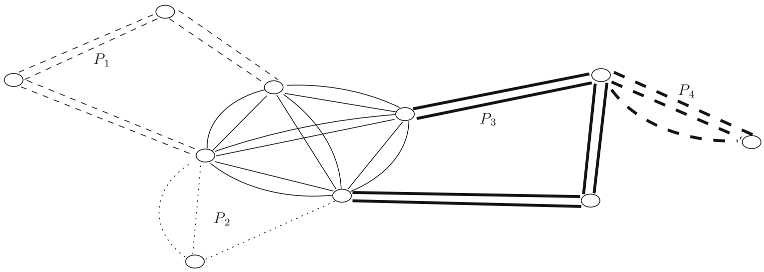

Fig. 4 Four different prime sets. Edges in thin lines do not belong to any prime set

P_2 is depicted with dotted lines. The prime set P_3 is shown with thick lines, and the prime set set P_4 is drawn with thick dashed lines. The edges not in a prime set appear with thin lines.

5.3 A relation among prime sets

Now we extend the definition of prime sets and derive a relation among different prime sets. Starting with a graph that has strength λ, we obtain all prime sets of level zero $\{P_i\}$. Then we remove all edges of these prime sets. Lemma 3 implies that the remaining connected components have strength greater than or equal to λ. Then for each connected component having strength equal to λ, we compute all its prime sets (of level zero with respect to the connected component). We call these *prime sets of level one* (with respect to the original graph). We continue this procedure until each connected component has strength greater than λ, (a single node has strength equal to infinity).

Remark 11 The edges with both endnodes in a component of strength greater than λ, i.e., edges not in any prime set, do not appear in any solution of the strength problem.

Suppose that we are at level j treating a connected component $H = (U, F)$, and let $\{T_1, \ldots, T_p\}$ be a partition of U that gives a prime set P. Then there is at least one prime set of level $j - 1$ that has edges incident to at least two different sets T_i. Otherwise the procedure in Sect. 5.2 would have given a prime set of level $j - 1$. See

Fig. 5a. We could also have prime sets of level less than $j-1$ having edges incident to at least two different sets $\{T_i\}$. Now we have to see that Lemma 9 and Corollary 10 extend to prime sets of level greater than zero.

Lemma 12 *If $P = \delta(S_1, \dots, S_p)$ is a prime set of level $j \geq 1$, and $\delta(T_1, \dots, T_q)$ is a solution of the strength problem in G, then either*

$$P \subseteq \delta(T_1, \dots, T_q) \;\; or$$
$$P \cap \delta(T_1, \dots, T_q) = \emptyset.$$

Proof The prime set P is a solution of the strength problem in a connected component $H = (U, F)$, that is obtained after removing all prime sets of level less than j. Let k be an index so that $U \cap T_i \neq \emptyset$ for $i = 1, \dots, k$, and $U \cap T_i = \emptyset$ for $i = k+1, \dots, q$.

If $k \geq 2$, let $T'_i = U \cap T_i$, for $i = 1, \dots, k$. Since H has strength λ, we have $|\delta(T'_1, \dots, T'_k)| \geq \lambda(k-1)$. Lemma 4 implies $|\delta(T'_1, \dots, T'_k)| = \lambda(k-1)$, because $\delta(T_1, \dots, T_q)$ is a solution of the strength problem in G. Thus $\delta(T'_1, \dots, T'_k)$ is a solution of the strength problem in H, and Lemma 9 implies $P \subseteq \delta(T'_1, \dots, T'_k)$ or $P \cap \delta(T'_1, \dots, T'_k) = \emptyset$.

If $k = 1$, then $P \cap \delta(T_1, \dots, T_q) = \emptyset$. □

This implies the Corollary below whose proof is similar to the one of Corollary 10.

Corollary 13 *For any two edges e and f in a prime set P of level $j \geq 1$, we have $\tilde{x}(e) = \tilde{x}(f)$, for any vector $\tilde{x} \in \mathscr{C}$.*

The lemma below gives a relation among the variables associated to different prime sets.

Lemma 14 *Let $P = \delta(T_1, \dots, T_p)$ be a prime set of level j. Let P' be a prime set of level less than j, so that there are at least two different sets T_{i_1} and T_{i_2}, with $P' \cap \delta(T_{i_k}) \neq \emptyset$, for $k = 1, 2$. Let $\tilde{x} \in \mathscr{C}$, and let α be the value of the variables associated with the edges in P'. If β is the value of the variables associated with the edges in P, then $\alpha \geq \beta$.*

Proof Let $\{W_1, \dots, W_q\}$ be a partition of V that is a solution of the strength problem. If $P \subset \delta(W_1, \dots, W_q)$ then we should also have $P' \subset \delta(W_1, \dots, W_q)$. To see this, recall that $P' = \delta(U_1, \dots, U_r)$ and if $P' \not\subset \delta(W_1, \dots, W_q)$, Lemma 12 implies $\cup_l U_l \subset W_r$, for some index r. This implies $T_{i_k} \subset W_r$, for $k = 1, 2$, and from Lemma 12 we have $P \cap \delta(W_1, \dots, W_q) = \emptyset$, a contradiction.

Thus if $e \in P$, $f \in P'$ and $\bar{x}$ is an extreme point of the core, then $\bar{x}(e) \leq \bar{x}(f)$. Since $\tilde{x}$ is a convex combination of extreme points, the same inequality holds for $\tilde{x}$. □

For the prime sets P and P' in Lemma 14 we write $P' \succ P$. In Fig. 5a we depict a prime set $\delta(S_1, \dots, S_p)$ of level zero, and a prime set $\delta(T_1, \dots, T_q)$ of level one, where $\{T_i\}$ is a partition of S_1. There are edges from $\delta(S_1)$ incident to two different sets T_i, then $\alpha \geq \beta$, where $x(e) = \alpha$, for $e \in \delta(S_1, \dots, S_p)$, and $x(f) = \beta$, for $f \in \delta(T_1, \dots, T_q)$. In Fig. 5b we have two prime sets of level zero. One depicted with

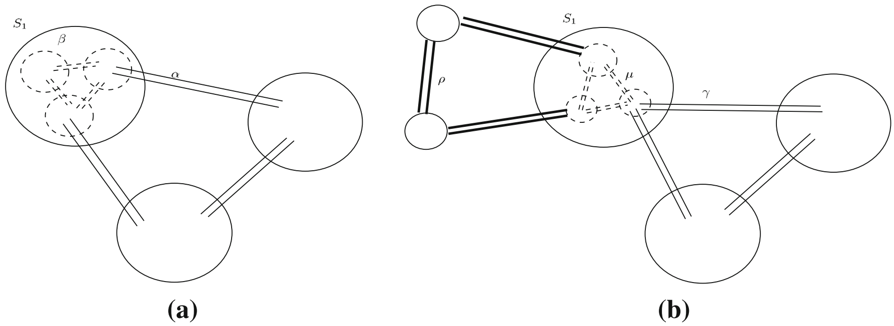

Fig. 5 In Case **a** we have $\alpha \geq \beta$. In Case **b** $\rho \geq \mu$, and there is no relation with γ

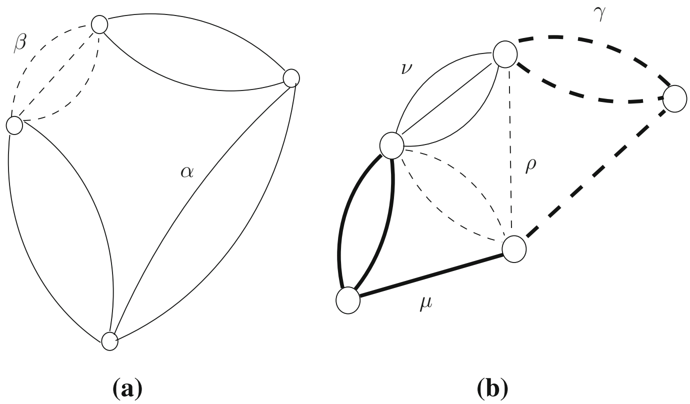

Fig. 6 Examples. In Case **a** we have $\alpha \geq \beta$. In Case **b** there is no relation between γ and μ, but we have $\mu \geq \rho$, $\gamma \geq \rho$ and $\rho \geq \nu$

thick lines whose variables take the value ρ, and one with thin lines whose variables take the value γ. There is a prime set of level one shown with dashed lines, whose variables take the value μ. Here we have $\rho \geq \mu$, and we cannot derive any relation with γ. In Fig. 6 we show some further examples.

Remark 11 and Lemma 12 lead to the following Corollary.

Corollary 15 *If $\delta(S_1, \ldots, S_t)$ is a solution of the strength problem, then $\delta(S_1, \ldots, S_t) = \cup_{i=1}^{r} P_i$, where $\{P_i\}$ are prime sets. The sets $\{P_i\}$ can be obtained by contracting all edges not in $\delta(S_1, \ldots, S_t)$, and then finding all prime sets in the new graph. Moreover $t = \sum_{i=1}^{r} \pi(P_i) - r + 1$.*

We conclude this section with the following theorem.

Theorem 16 *Let $n = |V|$, then $2n - 1$ is an upper bound for the number of prime sets.*

Proof Consider the procedure that generates the prime sets. The first prime set comes from a partition $\{S_1, \ldots, S_p\}$ of V. The second prime set comes from a partition $\{T_1, \ldots, T_q\}$ of some set S_i, even if the second prime set has also level zero. Each new prime set is obtained by taking partitions of some of the node-sets already produced, until all prime sets have been obtained. Thus if put together all the node-sets produced by this procedure, we have a nested (or laminar) family. It is well known that a nested family of sets from a ground set with n elements has at most $2n - 1$ elements. Thus $2n - 1$ is an upper bound for the number of prime sets. □

6 Computing the nucleolus

In Sect. 2 we saw that the computation of the nucleolus reduces to a sequence of linear programs. Here we study the first two linear programs in this sequence. In the next section we shall see that the second linear program has a unique solution, so it is the nucleolus.

The first linear program is

$$\begin{aligned}&\max \epsilon\\&x(E) = v(E)\\&x(S) \geq v(S) + \epsilon \quad \text{for all } S \subset E.\end{aligned}$$

We assume that the core is non-empty, thus $\epsilon \geq 0$. Let $k = v(E)$, so there are k disjoint spanning trees $T_1, \ldots, T_k$. If $x \in \mathscr{C}$, we have $x(T) \geq 1$ for every spanning tree, then to have $x(E) = k$, we have $x(T_i) = 1$, for $i = 1, \ldots, k$. This shows that $\epsilon = 0$ is the optimal value, and the set of optimal solutions is exactly the core.

Using the notation of Sect. 2, the second linear program is

$$\begin{aligned}&\max \epsilon\\&x(S) \geq v(S) + \epsilon, \quad \forall S \notin \mathcal{F}(P_1(0))\\&x \in P_1(0).\end{aligned}$$

Let $\mathscr{T}$ be the set of spanning trees, and $\mathscr{T}_0$ the set of spanning trees T such that $x(T) = 1$ for all $x \in \mathscr{C}$. Also let E_0 be the set of edges e such that $x(e) = 0$ for all $x \in \mathscr{C}$. We claim that (20)–(24) is equivalent to the linear program above. The proof of this similar to the proof of Lemma 5.

$$\max \epsilon \tag{20}$$
$$x(T) = 1, \quad \text{for } T \in \mathscr{T}_0, \tag{21}$$
$$x(e) = 0, \quad \text{for } e \in E_0, \tag{22}$$
$$x(T) \geq 1 + \epsilon, \quad \text{for } T \in \mathscr{T} \backslash \mathscr{T}_0, \tag{23}$$
$$x(e) \geq \epsilon, \quad \text{for } e \in E \backslash E_0. \tag{24}$$

This linear program has an exponential number of inequalities, in the remainder of this section we give properties of these inequalities, so that we can derive an equivalent formulation that has only polynomial size.

6.1 Constraints (21) and (22)

The edges in E_0 are the edges that do not belong to any prime set, so this set is easy to identify. Now we have to identify the set $\mathscr{T}_0$.

Lemma 17 *A tree T belongs to $\mathscr{T}_0$ if and only if*

$$|T \cap P| = \pi(P) - 1, \tag{25}$$

for each prime set P (of any level).

Proof The proof consists of the two parts below.

(i) Assume that T is a tree that satisfies (25) for every prime set. Let $\{S_1, \ldots, S_p\}$ be a partition that gives a solution of the strength problem. Then $\delta(S_1, \ldots, S_p) = \cup_{i=1}^{q} P_i$, where $\{P_i\}$ is a family of prime sets. Moreover $p = \sum_{i=1}^{q} \pi(P_i) - q + 1$. The vector associated with this partition is defined by $\bar{x}(e) = 1/(p-1)$ if $e \in \delta(S_1, \ldots, S_p)$, and $\bar{x}(e) = 0$ otherwise. Thus

$$\bar{x}(T) = \frac{1}{p-1} \sum_{i=1}^{q} \big(\pi(P_i) - 1\big) = \frac{p-1}{p-1} = 1.$$

Since every vector in the core is a convex combination of vectors associated with partitions that are solutions of the strength problem, we have $x(T) = 1$ for every vector $x \in \mathscr{C}$.

(ii) Now let us assume that T is a tree such that $x(T) = 1$ for every vector $x \in \mathscr{C}$. Here we use induction on the level of the prime sets. Consider first a prime set P of level zero. The vector associated with P is defined by $\bar{x}(e) = 1/(\pi(P) - 1)$ if $e \in P$, and $\bar{x}(e) = 0$ otherwise. Since $\bar{x}(T) = 1$, we have that (25) holds for P.

Assume that (25) holds for all prime sets of level less than l, and consider now a prime set Q of level l. Let $\{P_1, \ldots, P_r\}$ be the family of prime sets (of level less than l) with $P_i \succ Q$. We have $|T \cap \cup_{i=1}^{r} P_i| = \sum_{i=1}^{r} \pi(P_i) - r$, because of the induction hypothesis. Let $\tilde{x}$ be the vector associated with $Q \cup (\cup_i P_i)$. Thus $\tilde{x}(e) = 1/(t-1)$ if $e \in Q \cup (\cup_i P_i)$, and $\tilde{x}(e) = 0$ otherwise. Here $t = \sum_{i=1}^{r} \pi(P_i) + \pi(Q) - r$. Since $\tilde{x}(T) = 1$, $|T \cap \big(Q \cup (\cup_i P_i)\big)| = t - 1 = \sum_{i=1}^{r} \pi(P_i) + \pi(Q) - r - 1$. Thus $|T \cap Q| = \pi(Q) - 1$, and the proof is complete. □

6.2 Constraints (23)

To treat inequalities (23) we need the following lemmas.

Remark 18 Let $\hat{x}$ be an optimal solution of (20)–(24), and let $\hat{\epsilon}$ be its optimal value. If T is a spanning tree such that there is a vector $x_T \in \mathscr{C}$ with $x_T(T) > 1$, then $\hat{x}(T) \geq 1 + \hat{\epsilon}$. Also if e is an edge such that there is a vector $x_e \in \mathscr{C}$ with $x_e(e) > 0$, then $\hat{x}(e) \geq \hat{\epsilon}$.

Consider a prime set $P = \delta(S_1, \ldots, S_p)$ of level l. Assume that $Q = \delta(U_1, \ldots, U_q)$ is another prime set with $P \succ Q$. Thus Q has level greater than l, and there are at least

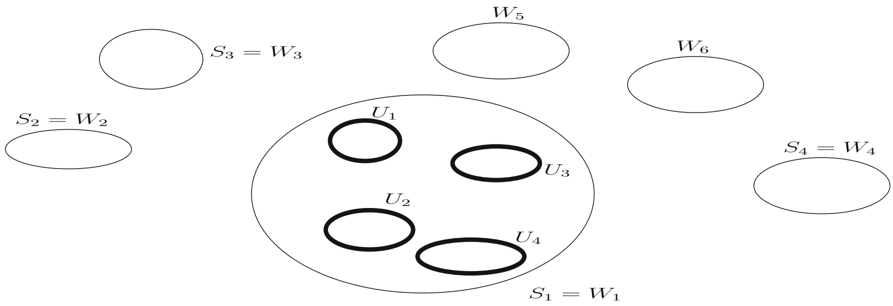

Fig. 7 The sets $\{S_j\}$ and $\{W_k\}$ appear with thin lines, and the sets $\{U_i\}$ appear with thick lines

two edges from P incident to two different sets U_i. Let $\{W_1, \ldots, W_r\}$ be a partition of V that gives a solution of the strength problem, with $\{S_1, \ldots, S_p\} \subseteq \{W_1, \ldots, W_r\}$, and $\cup_i U_i \subset S_1 = W_1$, say. Thus $P \subseteq \delta(W_1, \ldots, W_r)$, and $Q \cap \delta(W_1, \ldots, W_r) = \emptyset$. We depict this in Fig. 7, sets S_j appear with thin lines, some sets W_l coincide with some sets S_j. Let $\bar{x}$ be the vector of the core associated with $\delta(W_1, \ldots, W_r)$. Let T be a spanning tree in $\mathscr{T}_0$, thus $\bar{x}(T) = 1$. With the two lemmas below we show that there is a new tree T' obtained from T by removing one edge in $T \cap \delta(U_1, \ldots, U_q)$, and adding an edge in $\delta(S_1, \ldots, S_p)$, therefore $\bar{x}(T') > 1$. Remark 18 implies that $\hat{x}(T') \geq 1 + \hat{\epsilon}$, if $\hat{x}$ is an optimal solution of (20)–(24) of value $\hat{\epsilon}$. Thus we have

$$\hat{x}(e) \geq \hat{x}(f) + \hat{\epsilon} \quad \text{if } e \in P \text{ and } f \in Q. \tag{26}$$

Now we have to prove the existence of the tree T' mentioned above. If $P = \delta(S_1, \ldots, S_p)$, where $\{S_1, \ldots, S_p\}$ is a family of node-sets, and F is a set of edges, we denote by $G(P, F)$ a graph with nodes $\{s_1, \ldots, s_p\}$ and edge-set F'. Here s_i represents the set S_i, for $i = 1, \ldots, p$. The edge-set F' is built as follows, if F contains an edge uv with $u \in S_i$ and $v \in S_j$, then we add the edge $s_i s_j$ to F'.

Lemma 19 *Let $P = \delta(S_1, \ldots, S_p)$ be a prime set (of any level), and $T \in \mathscr{T}_0$, then the graph $G(P, T)$ is a tree.*

Proof Let $T \in \mathscr{T}_0$. Recall that for a prime set P we have $|T \cap P| = \pi(P) - 1$. Suppose that $P = \delta(S_1, \ldots, S_p)$ is a prime set of level l. Let $P_1, \ldots, P_r$ be the prime sets of level less than l. Let $R = \delta(W_1, \ldots, W_q) = \cup P_i$. We have $q = \sum \pi(P_i) - r$. We have that $\{S_1, \ldots, S_p\}$ is a partition of some set W_i. Let $\tilde{P} = R \cup P$. The set $\tilde{P}$ is also a solution of the strength problem, this is because $\tilde{P}$ was obtained from $R = \delta(W_1, \ldots, W_q)$ by replacing one set W_i by the partition $\{S_1, \ldots, S_p\}$ of W_i with $|\delta(S_1, \ldots, S_p)| = \lambda(p - 1)$.

Clearly $G(\tilde{P}, T)$ is a connected graph, and since $|T \cap \tilde{P}| = \sum \pi(P_i) - r + p - 1$, we have that $G(\tilde{P}, T)$ is a tree. Since $|T \cap P| = p - 1$, if $G(P, T)$ is not a tree, it has a cycle. This implies that $G(\tilde{P}, T)$ has a cycle, a contradiction. ◻

Lemma 20 *The tree T' described above always exists.*

Proof Assume that edges in P are incident to at least U_1 and U_2. We have two cases.

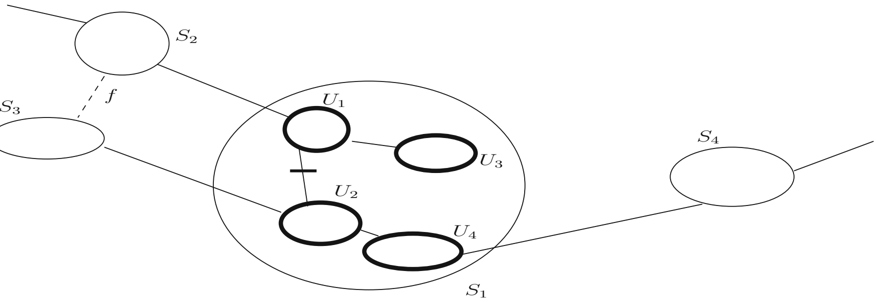

Fig. 8 A new tree obtained by adding an edge with higher weight, and removing an edge with lower weight

- If $T \cap P$ has only edges incident to U_1, say, we add an edge in P that is incident to U_2. This creates a cycle in T that involves edges in $G(Q, T)$. We remove one of these edges.
- Suppose now that $T \cap P$ contains edges incident to $U_1, \ldots, U_r$, with $r > 1$. Since $G(Q, T)$ is a tree, we suppose that there is an edge in T between S_2 and U_1, and an edge in T between S_3 and U_2, also there is no edge in T between S_2 and S_3. Now we treat two cases.
  * If there is an edge f between S_2 and S_3, when we add f to T we obtain a cycle involving edges in $G(Q, T)$, and thus T' is obtained by removing one of these edges, see Fig. 8.
  * If there is no edge between S_2 and S_3, assume that $S' = \{S_2, \ldots, S_{p_1}\}$ are the sets that are connected to U_1 using edges in T, but none of the edges in $G(Q, T)$. Set $S'' = \{S_{p_1+1}, \ldots, S_p\}$. Since P is a prime set, we have that $\{S_1, S_2, \ldots, S_p\}$ is a solution of the strength problem in a subgraph of G. Lemma 4 and the fact that P is a prime set imply

$$|\delta(S_1, S_2, \ldots, S_{p_1})| < \lambda(p_1 - 1) \text{ and } |\delta(S_1, S_{p_1+1}, \ldots, S_p)| < \lambda(p - p_1).$$

    If there is no edge between sets in S' and sets in S'', then

$$|\delta(S_1, S_2, \ldots, S_p)| = |\delta(S_1, S_2, \ldots, S_{p_1})| + |\delta(S_1, S_{p_1+1}, \ldots, S_p)| < \\ < \lambda(p - 1),$$

    a contradiction. Thus there is an edge f between a set in S' and and a set in S''. We add f to T and proceed as in the case above.

□

We have seen that any optimal solution of (20)–(24) satisfies (26). Thus we can restrict the core to vectors satisfying inequalities similar to (26). Now we prove that if $x \in \mathscr{C}$, $x(e) \geq \epsilon$ for all $e \in E \backslash E_0$, and if for every pair P, Q of prime sets with $P \succ Q$, we have

$$x(e) \geq x(f) + \epsilon, \quad \text{if } e \in P,\ f \in Q, \tag{27}$$

then we have

$$x(T) \geq 1 + \epsilon, \tag{28}$$

for every spanning tree $T \in \mathscr{T} \backslash \mathscr{T}_0$. For that we need the following two lemmas.

Lemma 21 *Consider a spanning tree $T \in \mathscr{T} \backslash \mathscr{T}_0$. Among the prime sets that violate (25), choose a prime set P of minimum level l. Then we have $|T \cap P| > \pi(P) - 1$.*

Proof Consider the graph $G' = (V', E')$ obtained by contracting all edges in E_0, all edges in prime sets of level greater than l, and also all edges in prime sets of level l that are different from P. Let $P_1, \ldots, P_r$ be the prime sets that remain. Then $|V'| = \sum_{i=1}^{r} \pi(P_i) - r + 1$. We have $|T \cap P_i| = \pi(P_i) - 1$, if $P_i \neq P$. If $|T \cap P| < \pi(P) - 1$ we have

$$\sum_{i=1}^{r} |T \cap P_i| < \sum_{i=1}^{r} \pi(P_i) - r.$$

Then $|T \cap E'| < |V'| - 1$. We have a contradiction because the graph $H = (V', T \cap E')$ is connected. □

Lemma 22 *Let x be a vector such that*

$$\begin{aligned} x(e) &= 0 \quad \text{for } e \in E_0, \\ x(T) &= 1 \quad \text{for } T \in \mathscr{T}_0, \\ x(e) &\geq \epsilon \quad \text{for } e \in E \backslash E_0. \end{aligned}$$

Then if for every pair P, Q of prime sets with $P \succ Q$, we have

$$x(e) \geq x(f) + \epsilon, \quad \text{if } e \in P,\ f \in Q,$$

this implies

$$x(T) \geq 1 + \epsilon,$$

for $T \in \mathscr{T} \backslash \mathscr{T}_0$.

Proof Let $T \in \mathscr{T} \backslash \mathscr{T}_0$, and let P be a prime set violating (25) of minimum level l. Lemma 21 shows that $|T \cap P| > \pi(P) - 1$.

Let $R_1, \ldots, R_r$ be the prime sets of level less than l. Let $R = \cup R_i$, as in the proof of Lemma 19, we have that $G(R, T)$ is a tree. Also $R = \delta(W_1, \ldots, W_r)$, and P is a prime set in the subgraph induced by a set W_j. Since $G(R, T)$ is a tree, the subgraph of T induced by W_j is also a tree.

Let $P = \delta(L_1, \ldots, L_k)$ where $\{L_1, \ldots, L_k\}$ is a partition of W_j. The graph $G(P, T)$ has a cycle. Thus there is a set L_i, so that the subgraph of T induced by L_i is not connected. We have to remove from T an edge e from $G(P, T)$ and add an edge f with endnodes in L_i, to obtain a new tree; we should also have $x(e) \geq x(f) + \epsilon$. To accomplish this we have to treat several cases:

– If the subgraph induced by L_i has strength greater than λ, we pick an edge f joining two components of the subgraph of T induced by L_i. We remove an edge from $G(P, T)$ to break the cycle just created. We have $f \in E_0$, so $x(f) = 0$, and $x(e) \geq \epsilon$.
– If the subgraph induced by L_i has strength equal to λ, we find a prime set $Q = \delta(U_1, \ldots, U_q)$, where $\{U_1, \ldots, U_q\}$ is a partition of L_i. We have the cases below:
  * If the edges from $G(P, T)$ are incident to only one set U_l, we replace L_i by U_l, and keep working with U_l.
  * Otherwise we have $P \succ Q$. We have two new cases:
    (i) If $G(Q, T)$ is not connected, we add to T an edge $f \in Q$ connecting two components of $G(Q, T)$, we remove an edge from $G(P, T)$ to break the cycle just created. We have $x(e) \geq x(f) + \epsilon$.
    (ii) If $G(Q, T)$ is connected there is a set U_l so that the subgraph of T induced by U_l is not connected. Then we replace L_i by U_l, and keep working with U_l.

Denote by T_1 the new tree obtained, we have $x(T_1) \leq x(T) - \epsilon$.

We keep repeating this procedure to generate a sequence $T_1, \ldots, T_l$ of trees, until the last tree T_l is in $\mathcal{T}_0$. This sequence is finite because each time an edge in a prime set is being exchanged with another edge in a prime set of higher level, or with an edge in E_0. We have $1 = x(T_l) \leq x(T) - l\epsilon$. This implies $x(T) \geq 1 + \epsilon$. □

Thus since there is a polynomial number of prime sets, and for all edges in a prime set its variables take the same value, we have a polynomial number of inequalities (27) implying inequalities (28). This will be used in the next section.

7 A combinatorial algorithm for the nucleolus

Now we can set a linear program of polynomial size that is equivalent to (20)–(24). We use the following four properties.

– For each prime set all variables associated to its edges take the same value.
– Condition (25) characterizes the trees in $\mathcal{T}_0$.
– Condition (27) implies inequalities between variables associated with prime sets.
– We assume that for edges in E_0 their variables are equal to zero.

Thus to each prime set P_i we associate the variable y_i, and we define the following linear program.

$$\max \epsilon \tag{29}$$

$$\sum_i \Big(\pi(P_i) - 1\Big) y_i = 1 \tag{30}$$

$$y_i \geq y_j + \epsilon \quad \text{if } P_i \succ P_j \tag{31}$$

$$y_i \geq \epsilon \quad \text{if } P_i \text{ is a minimal element of the partial order.} \tag{32}$$

Equation (30) is based on condition (25) and correspond to the equations associated with trees in $\mathscr{T}_0$. Inequalities (31) come from condition (27). Inequalities (32) correspond to edges not in E_0.

Consider a vector $(\bar{y}, \bar{\epsilon})$ satisfying (30)–(32). We can define a vector $\bar{x} \in \mathbb{R}^E$, so that $\bar{x}(e) = 0$, for $e \in E_0$, and $\bar{x}(e) = \bar{y}_{i(e)}$, for $e \in E \backslash E_0$. Here $i(e)$ is the prime set containing the edge e. Lemma 22 shows that $(\bar{x}, \bar{\epsilon})$ satisfies (21)–(24). Thus the linear program above is equivalent to (20)–(24). The following lemma shows how to solve it.

Lemma 23 *Let $(\hat{y}, \hat{\epsilon})$ be an optimal solution of (29)–(32). Then for each index i, at least one inequality $y_i \geq y_j + \epsilon$, or $y_i \geq \epsilon$ holds as equation for $(\hat{y}, \hat{\epsilon})$.*

Proof Assume that for the index i_0 none of these inequalities is tight. Then for a small value $\beta > 0$ we can define $\bar{y}_{i_0} = \hat{y}_{i_0} - \beta$ and $\bar{y}_i = \hat{y}_i$ for $i \neq i_0$. Then $(\bar{y}, \hat{\epsilon})$ satisfies (31)–(32), but $\sum_i \Big(\pi(P_i) - 1\Big)\bar{y}_i < 1$. Then there is a value $\lambda > 1$ so that $(\lambda\bar{y}, \lambda\hat{\epsilon})$ satisfies (30)–(32). This contradicts the optimality of $(\hat{y}, \hat{\epsilon})$. □

Based on this lemma we derive the algorithm below.

Step 0 Set $k = 1$.

Step 1 Set $y_i = k\epsilon$, for every minimal element P_i of the partial order. Remove all minimal elements from the partial order.

Step 2 If the partial order is empty go to Step 3, otherwise set $k \leftarrow k + 1$, and go to Step 1.

Step 3 At this point, for each prime set P_i we have $y_i = k_i\epsilon$, for some positive integer k_i. Then we have to set ϵ so that (30) is satisfied.

Thus Lemma 23 and the algorithm above show that the linear program (29)–(32) has a unique optimal solution, thus this gives the nucleolus. The algorithm above is similar to the algorithm given in [1,2] for the Spanning Connectivity Game, but the mathematical derivation is different. Now we can state our main result.

Theorem 24 *If the core is non-empty, the nucleolus of the strength game can be computed in $O(n^5 m)$ time.*

Proof Computing the prime sets dominates the computing time of all other parts of the algorithm. The preflow algorithm takes $O(n^3)$ time, and the strength problem requires n applications of the preflow algorithm, so the strength problem requires $O(n^4)$ time. Computing one prime set requires solving at most m strength problems. Since there are $O(n)$ prime sets, we obtain the required bound. □

8 Concluding remarks

We have given a polynomial combinatorial algorithm for computing the nucleolus of the Network strength game, when the core is non-empty. We conjecture that when the core is empty the nucleolus can also be computed in polynomial time.

We could define a slightly different game, where for an edge-set S, we denote by $v'(S)$ the value of a fractional packing of spanning trees with capacities equal to one for each edge in S. With this value function, the core is non-empty for every connected graph, and the algorithm of Sect. 7 gives the nucleolus.

Acknowledgements We are grateful to both referees. Their comments helped us to improve the presentation.

References

1. Aziz, H., Lachish, O., Paterson, M., Savani, R.: Wiretapping a hidden network. In: Leonardi, S. (ed.) Internet and Network Economics, pp. 438–446. Springer (2009)
2. Aziz, H., Lachish, O., Paterson, M., Savani, R.: Wiretapping a hidden network. arXiv preprint arXiv:0909.5293 (2009)
3. Baïou, M., Barahona, F.: On the nucleolus of shortest path games. In: Bilò, V., Flammini, M. (eds.) Algorithmic Game Theory, pp. 55–66. Springer International Publishing, Cham (2017)
4. Barahona, F.: Packing spanning trees. Math. Oper. Res. **20**(1), 104–115 (1995)
5. Cheng, E., Cunningham, W.H.: A faster algorithm for computing the strength of a network. Inf. Process. Lett. **49**(4), 209–212 (1994)
6. Cunningham, W.H.: Optimal attack and reinforcement of a network. J. ACM (JACM) **32**(3), 549–561 (1985)
7. Deng, X., Fang, Q.: Algorithmic cooperative game theory. In: Pardalos, P.M., Migdalas, A., Pitsoulis, L. (eds.) Pareto Optimality, Game Theory and Equilibria, pp. 159–185. Springer (2008)
8. Deng, X., Fang, Q., Sun, X.: Finding nucleolus of flow game. J. Comb. Optim. **18**(1), 64–86 (2009)
9. Deng, X., Ibaraki, T., Nagamochi, H.: Algorithmic aspects of the core of combinatorial optimization games. Math. Oper. Res. **24**(3), 751–766 (1999)
10. Deng, X., Papadimitriou, C.H.: On the complexity of cooperative solution concepts. Math. Oper. Res. **19**(2), 257–266 (1994)
11. Faigle, U., Kern, W.: On some approximately balanced combinatorial cooperative games. Z. Oper. Res. **38**(2), 141–152 (1993)
12. Faigle, U., Kern, W., Fekete, S.P., Hochstättler, W.: On the complexity of testing membership in the core of min-cost spanning tree games. Int. J. Game Theory **26**(3), 361–366 (1997)
13. Gabow, H.N., Manu, K.: Packing algorithms for arborescences (and spanning trees) in capacitated graphs. Math. Program. **82**(1–2), 83–109 (1998)
14. Goemans, M.X., Skutella, M.: Cooperative facility location games. J. Algorithms **50**(2), 194–214 (2004)
15. Granot, D., Granot, F.: Computational complexity of a cost allocation approach to a fixed cost spanning forest problem. Math. Oper. Res. **17**(4), 765–780 (1992)
16. Granot, D., Huberman, G.: On the core and nucleolus of minimum cost spanning tree games. Math. Program. **29**(3), 323–347 (1984)
17. 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)
18. Gueye, A., Walrand, J.C., Anantharam, V.: Design of network topology in an adversarial environment. In: Alpcan, T., Buttyán, L., Baras, J.S. (eds.) Decision and Game Theory for Security, pp. 1–20. Springer (2010)
19. Gusfield, D.: Connectivity and edge-disjoint spanning trees. Inf. Process. Lett. **16**(2), 87–89 (1983)
20. Kalai, E., Zemel, E.: Totally balanced games and games of flow. Math. Oper. Res. **7**(3), 476–478 (1982)
21. Kern, W., Paulusma, D.: Matching games: the least core and the nucleolus. Math. Oper. Res. **28**(2), 294–308 (2003)
22. 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)
23. Megiddo, N.: Computational complexity of the game theory approach to cost allocation for a tree. Math. Oper. Res. **3**(3), 189–196 (1978)
24. Nash-Williams, C.S.J.: Edge-disjoint spanning trees of finite graphs. J. Lond. Math. Soc. **1**(1), 445–450 (1961)
25. Potters, J., Reijnierse, H., Biswas, A.: The nucleolus of balanced simple flow networks. Games Econ. Behav. **54**(1), 205–225 (2006)

26. Schmeidler, D.: The nucleolus of a characteristic function game. SIAM J. Appl. Math. **17**(6), 1163–1170 (1969)
27. Shapley, L.S.: On balanced sets and cores. Nav. Res. Logist. Q. **14**(4), 453–460 (1967)
28. Solymosi, T., Raghavan, T.E.: An algorithm for finding the nucleolus of assignment games. International Journal of Game Theory **23**(2), 119–143 (1994)
29. Tutte, W.T.: On the problem of decomposing a graph into n connected factors. Journal of the London Mathematical Society **1**(1), 221–230 (1961)