On the core and f-nucleolus of flow games

Walter Kern[1] and Daniël Paulusma[2*]

[1] Faculty of Electrical Engineering, Mathematics and Computer Science
Department of Applied Mathematics
University of Twente, P.O.Box 217, NL-7500 AE Enschede
w.kern@math.utwente.nl

[2] Department of Computer Science, Durham University,
Science Laboratories, South Road, Durham DH1 3LE, U.K.
daniel.paulusma@durham.ac.uk

Abstract. Using the ellipsoid method, both Deng, Fang, and Sun, and Potters, Reijnierse, and Biswas show that the nucleolus of simple flow games (where all edge capacities are equal to one) can be computed in polynomial time. Our main result is a combinatorial method based on eliminating redundant $s-t$ path constraints such that only a polynomial number of constraints remains. This leads to efficient algorithms for computing the core, nucleolus and nucleon of simple flow games. Deng, Fang, and Sun also prove that computing the nucleolus for (general) flow games is NP-hard. We generalize this by proving that computing the f-nucleolus of flow games is NP-hard for all priority functions $f : 2^E \to \mathbb{R}^+$ that satisfy $f(A) > 0$ whenever $v(A) > 0$ (so, including the priority functions corresponding to the nucleolus, nucleon and per-capita nucleolus).

Keywords: flow game, computational complexity, core, nucleolus

MSC Subject Classification: 91A12, 03D15

OR/MS Classification: Games/group decisions, cooperative

1 Introduction

A *cooperative game* is given by a set E of *players* and a *characteristic function* $v : 2^E \to \mathbb{R}$ with $v(\emptyset) = 0$. A *coalition* is any subset $A \subseteq E$. We refer to $v(A)$ as the *worth* of coalition $A \subseteq E$, interpreted as the gain that the members of A can achieve by "cooperating" with each other. The worth $v(E)$ is also called the *total worth* of the game. The v-worths of many cooperative games are derived from solving an underlying discrete optimization problem (cf. [1]). For example, in a *matching game*, the underlying discrete structure is an undirected graph G with edge weights $w \in \mathbb{R}_+^E$. If this graph is bipartite we get an *assignment game* [28]. The players are represented by the vertices of G, and the worth $v(A)$ of a coalition A is defined as the size of a maximum weight matching in the subgraph of G induced by A. Another example is the class of *min-cost spanning tree games* [2] defined on an undirected complete graph G with supply node s

* Research supported by the EPSRC, project EP/D053633/1.

and edge weighting w. The players are the vertices of G without s, and the (cost) worth $v(A)$ of a coalition A is equal to the weight of a minimum spanning tree in the subgraph of G induced by $A \cup \{s\}$.

In cooperative game theory it is often assumed that the *grand coalition* E is formed. The central problem is then to *allocate* the total worth $v(E)$ to the individual players $i \in E$ in a way that might be called "fair" (in a certain sense). An *allocation* is a vector $x \in \mathbb{R}^E$ with $x(E) = v(E)$. Here, we adopt the standard notation $x(A) = \sum_{i \in A} x_i$. A *solution concept* $\mathcal{S}$ for a class of cooperative games Γ is a function that maps each game $(E, v) \in \Gamma$ to a set $\mathcal{S}(v)$ of allocations for (E, v). These allocations are called $\mathcal{S}$*-allocations* and we say that $\mathcal{S}$ *prescribes* them to (E, v).

The choice for a specific solution concept depends on the notion of "fairness" that has been specified within the decision model. Well-known solution concepts in the literature are the core [12] which might be empty, and the nucleolus, which consists of a unique allocation [26]. See [23] for a survey. Multiplicative variants of the nucleolus are the nucleon [7] (also called the proportional nucleolus [34]), and the per-capita nucleolus [17, 33] (also called the weak nucleolus [27]). Both variants can be more natural to model situations in which taxation is imposed proportionally to the worth (e.g. interest or sales tax). Later on, we give precise definitions of these concepts and discuss them in the more general framework of f-nucleoli, where $f : 2^E \to \mathbb{R}^+$ is a so-called priority function which has been introduced in [31]. Priority functions are closely related to the concept of taxation functions [29, 32] and have been studied in [9, 24] as well.

Two natural questions regarding the complexity of a certain solution concept $\mathcal{S}(v)$ are determining the computational complexity of

1. testing membership in $\mathcal{S}(v)$, i.e., checking whether a given allocation is a member of $\mathcal{S}(v)$;
2. computing an allocation in $\mathcal{S}(v)$.

For answering these questions, one takes the size of the underlying discrete structure as input size as this is more natural than taking the $2^{|E|}$ v-worths themselves. Both positive and negative results exist regarding these two questions. As an illustration, consider the earlier mentioned matching games and min-cost spanning tree games. In [6] it has been shown that testing membership in the core of min-cost spanning tree games is NP-hard while this can be done in polynomial time for matching games [7]. For assignment games [30] and *simple* matching games (i.e. with unit edge weights) [21] the nucleolus can be computed in polynomial time. For (general) matching games an allocation in the nucleon [7] can be computed in polynomial time. For min-cost spanning tree games computing the nucleolus [8] and computing an allocation in the nucleon [9] are NP-hard problems. Note that in these games the v-worths themselves can be computed efficiently.

This paper studies *flow games* introduced by Kalai and Zemel [19] to model profit allocation of integrated production systems with alternative production routes. The underlying structure of a flow game (E, v) is a *(flow) network*, i.e., a directed graph $G = (V, E)$ with *source* $s \in V$, *sink* $t \in V$ and positive edge capacities $c \in \mathbb{R}^E$. Note that we allow multiple edges. Each player in this game owns exactly one edge. Players cooperate with each other in order to allow a flow going from s to t. So, players are represented by (directed) edges and the v-worths are given by

$$v(A) := \text{maximum flow in } (V, A), \qquad A \subseteq E.$$

Note that v-worths can be efficiently computed (see e.g. [5]) and that $v(A) = 0$ for a coalition A that does not contain both s and t. Kalai and Zemel [20] show that every flow game has a nonempty core and that it is a trivial task to compute a core allocation. However, Fang et al. [10] show that testing membership in the core of a flow game is NP-hard. Deng, Fang and Sun [4] show that computing the nucleolus of a flow game is NP-hard. On the positive side, Granot and Granot [15] show that the nucleolus of a flow game can be computed in polynomial time if the underlying flow network is an augmented tree.

A *simple* flow game is a flow game on a *simple* network, i.e., with unit capacities ($c_e = 1$ for all $e \in E$). Granot and Granot [15] obtain a parametric description of the core of a simple flow game and use this result to obtain an easier description of its nucleolus. However, their work does not lead to a polynomial time algorithm for computing the nucleolus. The main obstacle is that their approach requires checking for pairs of edges a, b if there exists an $s - t$ path that goes through a but not through b. Here, a $v_0 - v_\ell$ *path* in a network $G = (V, E)$ is a directed path from $v_0 \in V$ to $v_\ell \in V$, i.e., a sequence $v_0 v_1 \ldots v_\ell$ of different vertices such that (v_{i-1}, v_i) is a (directed) edge in G for $i = 1, \ldots, \ell$. By writing $a = (x, y)$, the above problem is equivalent to checking if there exist an $s - x$ path and a $y - t$ path in $G' = (V, E \backslash \{b\})$ that are vertex-disjoint. The problem of deciding if such paths exist is NP-complete for directed graphs [14].

Fourteen years later, both Deng, Fang and Sun [4] and Potters, Reijnierse and Biswas [25] independently show that the nucleolus of a simple flow game can be computed in polynomial time. Both papers use the ellipsoid method as opposed to the approach in [15] that is based on removing redundant $s - t$ path constraints in the sequence of linear programs that determine the nucleolus (we will explain this sequence of linear programs later on). Potters, Reijnierse and Biswas [25] also make use of a polynomial description of the core in terms of so-called potential functions defined on a (modified) network.

Other properties of simple flow games such as core stability, core largeness are studied in [13], while [16] studies the reactive bargaining set.

Our results

We continue the study on flow games. First we focus on simple flow games in Section 2. We attack these games by a similar approach as followed in [15]: in the linear program descriptions of the solution concepts under consideration we try to find as many redundant $s - t$ path constraints as possible. However, our analysis is very different than the one performed in [15] as in the end we are left with a polynomial number of constraints (not necessarily corresponding to $s - t$ paths) describing the following three solution concepts:

- In Section 2.1 we will exhibit a new, polynomial size description of the core of a simple flow game.
- In Section 2.2 we show that the nucleolus of a simple flow game can be computed in polynomial time.
- In Section 2.3 we show that the nucleon of a simple flow game can be computed in polynomial time.

We would like to emphasize that our method does not rely on the ellipsoid method, as opposed to the "dual" approach in [4, 25] for computing the nucleolus in polynomial time, and (hopefully) provides some additional insight into the structure of the problem.

In Section 3 we study the f-nucleolus of a (general) flow game. We give a relatively short proof that shows that computing an allocation in the f-nucleolus is NP-hard for all priority functions f with $f(A) > 0$ if $v(A) > 0$. As for min-cost spanning tree games this is only known for a smaller subset of priority functions [9], this result was not expected beforehand. The nucleolus, per-capita nucleolus and nucleon all correspond to a priority function satisfying this property. Hence we immediately obtain the NP-hardness results for these three concepts.

Section 4 contains the conclusions. There we also mention some open problems.

2 Simple flow games

Consider a simple network (V, E). Throughout this section we use the following terminology. For two nonempty vertex-disjoint sets $S, T \subset V$ with $S \cup T = E$, $s \in S$ and $t \in T$ we write $[S : T]$ for the *cut* $\{(i, j) \in E \mid i \in S, j \in T\}$ and $[T : S]$ for $\{(j, i) \in E \mid i \in S, j \in T\}$. A *min cut* is a cut with the smallest number of edges. By the well-known Max-Flow Min-Cut theorem (cf. [3]), $v(E)$ is equal to the number of edges in a min cut.

A *min cut edge* is an edge $e \in E$ that is contained in some min cut $[S : T] \subseteq E$. A *reverse edge* is an edge $e \in [T : S]$ for some min cut $[S : T] \subseteq E$. We let M denote the set of min cut edges and we let R denote the set of reverse edges. Note that $M \cap R = \emptyset$ (we make this more explicit in Observation 1). In general, there may exist edges which are neither reverse nor min cut edges. We let $D := E \backslash (M \cup R)$ denote the set of *dummy edges*. See Figure 1 for two examples. In this figure the letters D, M, R denote the set to which the edges in the example networks belong. Both R and M (and consequently D) can be computed in polynomial time as follows. We say that we *identify* two vertices i, j if we replace i, j by a new vertex adjacent to all edges formerly adjacent to i or j. Then, an edge $e = (i, j)$ is in M if and only if identifying i with s and j with t (still) yields a min cut value of $v(E)$. Similarly, an edge $e = (i, j)$ is in R if and only if identifying i with t and j with s yields a min cut value of $v(E)$.

Let $\mathcal{P} \subseteq 2^E$ denote the family of *$s - t$ paths*. Recall that these are directed paths starting in s and ending in t. We define a *max flow set* in a simple network as a set of k pairwise edge-disjoint paths $P_1, \ldots, P_k$ in $\mathcal{P}$ that form a max flow of worth $k = v(E)$.

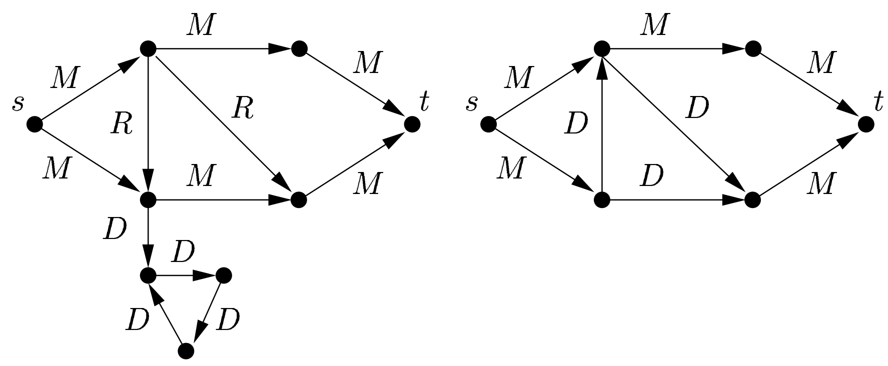

Fig. 1. Two examples of simple networks.

Observation 1 *Let $\{P_1, \ldots, P_k\}$ be a max flow set in a simple network (V, E). Then $M \subseteq (P_1 \cup \ldots \cup P_k)$ and $(P_1 \cup \ldots \cup P_k) \cap R = \emptyset$.*

Proof. Let $\{P_1, \ldots, P_k\}$ be a max flow set in (V, E). Every P_i $(i = 1, \ldots, k)$ passes through every (min) cut of G. Furthermore, every min cut contains k edges. These two facts together imply our claim. □

From now on we fix some max flow set $\{P_1, \ldots, P_k\}$. Note that besides s, t these paths may have other common vertices. An $i - j$ path P is denoted by P^{ij} and an $i - j$ subpath of a path P_ℓ is denoted by P_ℓ^{ij}. A subpath $P' \subseteq P_\ell \cap M$ joining $i \in V(M)$ to $j \in V(M)$ is denoted by M_ℓ^{ij}. A path $P' \subseteq D \cup R$ joining $i \in V$ to $j \in V$ is denoted by Q^{ij}. However, we write R^{ij} if we want to indicate that $Q^{ij} \cap R \neq \emptyset$ (so a path R^{ij} may contain edges from D). Note that this is exactly the case when i and j are separated by a min cut. Otherwise all edges of any path Q^{ij} belong to D, and if we want to indicate this we denote Q^{ij} by D^{ij}. We can then decompose an arbitrary $P \in \mathcal{P}$ as

$$P = D^{si_1} \cup M_{\ell_1}^{i_1 j_1} \cup Q^{j_1 i_2} \cup M_{\ell_2}^{i_2 j_2} \cup \ldots \cup Q^{j_{q-1} i_q} \cup M_{\ell_q}^{i_q j_q} \cup D^{j_q t}. \tag{2.1}$$

We observe that P cannot pass any reverse edges before passing any min cut, so P indeed starts and ends with (possibly empty) D^{si_1} respectively, $D^{j_q t}$.

We will now define two polynomially bounded sets $\mathcal{P}_0 \subseteq \mathcal{P}$ and $\mathcal{P}_1 \subset 2^E$ that play a crucial role in all our proofs in this section. We do this as follows. Whenever $(i, j) \in V(M)^2$ are joined by a path in $D \cup R$, we fix such a path $\overline{Q}^{ij}$. If $\overline{Q}^{ij} \cap R \neq \emptyset$ then we write $\overline{R}^{ij}$. In the other case $\overline{Q}^{ij} \subseteq D$ holds, and we denote $\overline{Q}^{ij}$ by $\overline{D}^{ij}$. Then $\mathcal{P}_0$ consists of all paths of the form

$$P = P_\ell^{si} \cup \overline{D}^{ij} \cup P_m^{jt}.$$

Here, we allow $i = j$. Then, as $\overline{D}^{ii}$ is empty for all $i \in V$, we ensure $P_1, \ldots, P_k \in \mathcal{P}_0$. The set $\mathcal{P}_1$ consists of all s-t *walks* of the form

$$P = P_\ell^{si} \cup \overline{R}^{ij} \cup P_m^{jt}.$$

Note that it is indeed possible that a coalition $P \in \mathcal{P}_1$ is not a path coalition as we might visit a vertex or edge twice by going from s to t along P, e.g., when $\ell = m$. We also note that $\overline{R}^{ii}$ does not exist.

Observation 2 *Both $\mathcal{P}_0$ and $\mathcal{P}_1$ have polynomial size and can be computed in polynomial time.*

Proof. Clearly, $|\mathcal{P}_0| = |\mathcal{P}_1| = O(|V|^2k^2) = O(|V|^2||E|^2)$. The second claim immediately follows from the fact that we can compute D, R, and a max flow set in polynomial time. □

The following lemma is crucial for every solution concept in this section. Note that the coalitions $\widetilde{P}_\rho$ defined in this lemma are in $\mathcal{P}_0 \cup \mathcal{P}_1$.

Lemma 1. *Let $P \in \mathcal{P}$ be decomposed as in (2.1) and let*

$$\widetilde{P}_\rho = P^{sj_\rho}_{\ell_\rho} \overline{Q}^{j_\rho i_{\rho+1}} P^{i_{\rho+1}t}_{\ell_{\rho+1}} \text{ for } \rho = 1, \ldots, q-1.$$

Let $x \equiv 0$ on $D \cup R$ for some $x \in \mathbb{R}^E$. Then the following holds:

- *If $q = 1$, then $x(P) = x(P_{\ell_1})$.*
- *If $q = 2$, then $x(P) = x(\widetilde{P}_1)$.*
- *If $q \geq 3$, then $x(P) = x(\widetilde{P}_1) + \sum_{\rho=2}^{q-1}(x(\widetilde{P}_\rho) - x(P_{\ell_\rho}))$.*

Proof. As $x \equiv 0$ on $D \cup R$ we may without loss of generality assume $D^{si_1} = P^{si_1}_{\ell_1}$ and $D^{j_qt} = P^{j_qt}_{\ell_q}$. For the same reason we may without loss of generality assume that $Q^{j_\rho i_{\rho+1}}$ equals $\overline{Q}^{j_\rho i_{\rho+1}}$.

If $q = 1$ we then obtain $x(P) = x(D^{si_1}) + x(M^{i_1j_1}_{\ell_1}) + x(D^{j_1t}) = x(P^{si_1}_{\ell_1}) + x(P^{i_1j_1}_{\ell_1}) + x(P^{j_1t}_{\ell_1}) = x(P_{\ell_1})$.

If $q = 2$ we obtain $x(P) = x(D^{si_1}) + x(M^{i_1j_1}_{\ell_1}) + x(Q^{j_1i_2}) + x(M^{i_2j_2}_{\ell_2}) + x(D^{j_2t}) = x(P^{sj_1}_{\ell_1}) + x(\overline{Q}^{j_1i_2}) + x(P^{i_2t}_{\ell_2}) = x(\widetilde{P}_1)$.

If $q \geq 3$, we identify each path P with its incidence vector in $\mathbb{Z}^E_+$. Then we can write $P + P_{\ell_2} + \ldots + P_{\ell_{q-1}} = \widetilde{P}_1 + \ldots + \widetilde{P}_{q-1}$, and the claim follows. □

2.1 The core of simple flow games

The *core* of a game (E, v), denoted by core(v), is the set of allocations that are fair in the sense that every coalition A gets at least its worth $v(A)$ (assuming that the v-worths represent profits):

$$\begin{array}{rll} \text{core}(v): & x(A) \geq v(A) & A \in 2^E \backslash \{\emptyset, E\} \\ & x(E) = v(E) & \end{array}$$

We first state two elementary results known in the literature.

Proposition 1 **([20]).** *Let (E, v) be a simple flow game, then*

$$\begin{array}{lll} \text{core}(v): & x(P) \geq 1 & P \in \mathcal{P} \\ & x_e \geq 0 & e \in E \\ & x(E) = v(E). & \end{array}$$

Theorem 1 **([20]).** *For simple flow games, core(v) is the convex hull of the incidence vectors of min cuts.*

By Observation 2 the next theorem gives the desired core result. We formulate it a bit stronger as this is useful for proving the polynomial time results for the nucleolus and nucleon in Section 2.2 and 2.3, respectively.

Theorem 2. *Let (E, v) be a simple flow game with max flow set $\{P_1, \ldots, P_k\}$, then*

$$\begin{array}{llll} \text{core}(v): & x(P) \geq 1 & P \in \mathcal{P}_1 \\ & x(P) = 1 & P \in \mathcal{P}_0 \\ & x_e \quad \geq 0 & e \in M \\ & x_e \quad = 0 & e \in D \cup R. \end{array}$$

All inequalities can be satisfied strictly.

Proof. Due to Observation 1, the above constraints imply $x(E) = \sum x(P_i) = k = v(E)$. By Lemma 1, these constraints also imply $x(P) \geq 1$ for each $P \in \mathcal{P}$. Hence, by Proposition 1, we have proven the above core characterization is correct.

We now show that all inequalities can be satisfied strictly. Let $[S^{(i)} : T^{(i)}]$, $i = 1, \ldots, n$ be a complete list of all min cuts and let $x^{(i)} \in \mathbb{R}^E$ denote their incidence vectors. Due to Theorem 1, every $x \in \text{core}(v)$ can be written as $x = \sum_i \lambda_i x^{(i)}$ such that $\sum_i \lambda_i = 1$. Hence $\overline{x} = \frac{1}{n} \sum_i x^{(i)}$ is a core allocation that allocates $\overline{x}_e \geq \frac{1}{n}$ to every $e \in M$ and $\overline{x}(P) \geq 1 + \frac{1}{n}$ to every $P \in \mathcal{P}_1$ as $P \cap R \neq \emptyset$ by definition and hence P must pass at least one min cut twice. $\square$

2.2 The nucleolus of simple flow games

Given an allocation $x \in \mathbb{R}^E$ for some game (E, v), we define the *excess* of a nonempty coalition $A \subsetneq E$ as $e(A, x) := x(A) - v(A)$. We first order all excesses $e(A, x)$ into a non-decreasing sequence to obtain the *excess vector* $\theta(x) \in \mathbb{R}^{2^{|E|}-2}$. The *nucleolus* of (E, v) is then defined as the set of allocations that lexicographically maximize $\theta(x)$ over all *imputations*, i.e., over all allocations $x \in \mathbb{R}^E$ with $x_e \geq v(\{e\})$ for all $e \in E$. Note that the nucleolus is not defined if the set of imputations is empty. Otherwise, due to a result by Schmeidler [26], the nucleolus consists of exactly one allocation.

We use the following alternative procedure (cf. [22]) for computing the nucleolus of games with a nonempty core such as flow games. Let (E, v) be a game with core$(v) \neq \emptyset$. Then we might seek for an allocation $x \in \mathbb{R}^E$ satisfying all coalitions (core constraints) as much as possible by solving

$$(\text{LP}_1) \qquad \begin{array}{ll} \varepsilon_1 := \max \varepsilon & \\ x(A) \geq v(A) + \varepsilon & A \in 2^E \backslash \{\emptyset, E\} \\ x(E) = v(E). & \end{array}$$

Note that $\varepsilon_1 \geq 0$, as core(v) is nonempty. The set of allocations $x \in \mathbb{R}^E$ for which (x, ε_1) is an optimum solution of (LP_1) is known as the *least core* of (E, v). Note that the least core consists of those allocations that maximize the smallest excess. This idea may be further pursued. Let $\mathcal{A}_0 := \{\emptyset, E\}$ and let $\mathcal{A}_1 \subseteq 2^E \backslash \mathcal{A}_0$ denote the set of coalitions A that *get tight* in (LP_1) in the sense that $x(A) = v(A) + \varepsilon_1$ holds for every optimal solution (x, ε_1) of (LP_1). For $\mathcal{A} \subseteq 2^E$ let $\langle \mathcal{A} \rangle \subseteq 2^E$ consist of all coalitions whose incidence vectors are linearly generated by the incidence vectors of coalitions in $\mathcal{A}$. So, for example, any $A \in \langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle$ has a fixed value $x(A)$ for each x in the least core of v. We may thus seek to further increase $x(A)$ for $A \in 2^E \backslash \langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle$ etc., leading to a sequence of linear programs

$$
(\mathrm{LP}_r) \qquad \begin{aligned} \varepsilon_r := \max \varepsilon & \\ x(A) \geq v(A) + \varepsilon & \quad A \in 2^E \backslash \langle \mathcal{A}_0 \cup \ldots \cup \mathcal{A}_{r-1} \rangle \\ x(A) = v(A) + \varepsilon_i, & \quad A \in \mathcal{A}_i \quad (i = 0, \ldots, r-1). \end{aligned}
$$

with $\varepsilon_0 := 0$ and $\mathcal{A}_r \subseteq 2^E \backslash \langle \mathcal{A}_0 \cup \ldots \cup \mathcal{A}_{r-1} \rangle$ being recursively defined as the set of coalitions A that *get tight* in (LP_r) in the sense that $x(A) = v(A) + \varepsilon_r$ holds for every optimal solution (x, ε_r) of (LP_r). We say that (LP_r) *fixes* $A \subseteq E$ if $A \in \langle \mathcal{A}_0 \cup \ldots \cup \mathcal{A}_r \rangle \backslash \langle \mathcal{A}_0 \cup \ldots \cup \mathcal{A}_{r-1} \rangle$. Note that a coalition A can be fixed by some (LP_r) without ever getting tight.

The dimension of the feasible regions of (LP_r) decreases in each step, so we end up with a unique optimum solution $x^* = x^*(v)$, the nucleolus of (E, v), after at most $m = |E|$ iterations. The nucleolus is in general difficult to compute, due to the exponentially many constraints in each (LP_r). However, for simple flow games, this can be done in polynomial time. Our main idea is the following. We fix a max flow set $\{P_1, \ldots, P_k\}$. After excluding the case when E is an $s-t$ path, we deduce that $\varepsilon_1 = 0$ and that the set of optimal solutions for (LP_1) simply coincides with the core, i.e., $\langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle \subseteq 2^E$ consists exactly of all coalitions generated by single edges $\{e\} \subseteq D \cup R$ and paths $P \in \mathcal{P}_0$ (cf. Theorem 2). As $x \equiv 0$ on $D \cup R$ in linear programs (LP_r) with $r \geq 2$, we may apply Lemma 1. This enables us to disregard all coalitions not in $D \cup R \cup \mathcal{P}_0 \cup \mathcal{P}_1$. As the proof for the nucleon in Section 2.3 goes along the same lines and the result for the nucleon is new, we leave out the proof details of Theorem 3, which has been shown in [4, 25] as well.

Theorem 3. *The nucleolus of a simple flow game can be computed in polynomial time.*

2.3 The nucleon of simple flow games

A multiplicative variant of the nucleolus, the so-called *nucleon*, has been introduced in [7]. Assuming $v \geq 0$ (and v-worths representing profits) this paper proposes to solve

$$
(\widehat{\mathrm{LP}}_r) \qquad \begin{aligned} \varepsilon_r := \max \varepsilon & \\ x(A) \geq (1+\varepsilon) v(A) & \quad A \in 2^E \backslash \langle \mathcal{A}_0 \cup \ldots \cup \mathcal{A}_{r-1} \rangle \\ x(A) = (1+\varepsilon_i) v(A) & \quad A \in \mathcal{A}_i \quad (i = 0, \ldots, r-1), \end{aligned}
$$

(where $\varepsilon_0 = 0$ and $\mathcal{A}_0 = \{\emptyset, E\}$) for $r = 1, 2, \ldots$ until $v(A) = 0$ holds for all $A \in 2^E \backslash \langle \mathcal{A}_0 \cup \ldots \cup \mathcal{A}_{r-1} \rangle$. Strictly speaking the above sequence of linear programs leads to the *prenucleon* (since we do not restrict ourselves to imputations only). However, if we assume that the core of a game (E, v) is nonempty, $\varepsilon_1 \geq 0$ holds, and then the nucleon and prenucleon coincide. Since flow games have a nonempty core, we can indeed make this assumption.

In contrast to the nucleolus, the nucleon is not necessarily a single point. The complexity of the nucleolus and nucleon may differ for a specific class of games, but in general also the nucleon is difficult to compute, due to the exponentially many constraints in each $(\widehat{\mathrm{LP}}_r)$. However, we can efficiently compute the nucleon of a simple flow game by applying our method. To start with, we exclude the trivial exception that E is an $s-t$ path and first observe that

$$(\widehat{\mathrm{LP}}_1) \qquad \begin{aligned} \varepsilon_1 := \max \varepsilon & \\ x(A) \geq (1+\varepsilon) v(A) & \quad A \in 2^E \backslash \{\emptyset, E\} \\ x(E) = v(E) & \end{aligned}$$

yields $\varepsilon_1 = 0$. This can be seen as follows. Theorem 2 implies $\varepsilon_1 \geq 0$. Let $\{P_1, \ldots, P_k\}$ be a max flow set and let (x, ε_1) be an optimal solution for $(\widehat{\mathrm{LP}}_1)$. Then $x \geq 0$ and hence $k = x(E) \geq \sum_{i=1}^k x(P_i) \geq (1+\varepsilon_1)k$, which implies $\varepsilon_1 \leq 0$. So, the set of optimal solutions for $(\widehat{\mathrm{LP}}_1)$ simply coincides with the core as was the case for (LP_1). By Theorem 2, $\langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle \subseteq 2^E$ consists exactly of all coalitions generated by single edges $\{e\} \subseteq D \cup R$ and paths $P \in \mathcal{P}_0$. Next let us turn to

$$(\widehat{\mathrm{LP}}_2) \qquad \begin{aligned} \varepsilon_2 := \ & \max \varepsilon & & \\ & x(A) \geq (1+\varepsilon) v(A) & & A \in 2^E \backslash \langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle \\ & x(P) = 1 & & P \in \mathcal{P}_0 \\ & x_e \quad = 0 & & e \in D \cup R. \end{aligned}$$

Which coalitions may possibly get tight in $(\widehat{\mathrm{LP}}_2)$? First we fix a max flow set $\{P_1, \ldots, P_k\}$. We call $A \subseteq E \backslash \langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle$ *critical* if A is of the form

$$\begin{aligned} A_e &= \{e\} \cup \bigcup_{h \neq m} P_h, & & \text{with } e \in P_m \cap M \\ A_{\ell m}^{ij} &= P_\ell^{si} \cup \overline{R}^{ij} \cup P_m^{jt} \cup \bigcup_{h \neq \ell, m} P_h, & & \text{with } \ell \neq m. \end{aligned}$$

Note that a coalition of the form A_e with $e \in P_m \cap M$ is not critical if and only if P_m consists of e. Furthermore, each critical coalition of the form $A_{\ell m}^{ij}$ corresponds to exactly one coalition $P_\ell^{si} \cup \overline{R}^{ij} \cup P_m^{jt} \in \mathcal{P}_1$. Hence the number of critical coalitions is $O(|E|) + |\mathcal{P}_1| = O(|V|^2|E|^2)$ and we can find al critical coalitions in polynomial time due to Observation 2.

We now show that for computing an allocation in the nucleon, it suffices to consider the constraints for the critical coalitions together with the single edge and path constraints (for paths in $\mathcal{P}_0$).

Theorem 4. *An allocation in the nucleon of a simple flow game can be computed in polynomial time.*

Proof. Let (E, v) be a simple flow game with $v(E) = k$. If $k = 1$, the nucleon coincides with the core (we leave the proof to the reader). Suppose $k \geq 2$. Fix some max flow set $\{P_1, \ldots, P_k\}$ of (E, v). Note that $v(A) = k-1$ if A is a critical coalition. We first show that $(\widehat{\mathrm{LP}}_2)$ is equivalent to

$$(\mathrm{LP}_2^*) \qquad \begin{aligned} \varepsilon_2 := \ & \max \varepsilon \\ & x(A) \geq (1+\varepsilon)(k-1) && A \text{ is critical} \\ & x(P) = 1 && P \in \mathcal{P}_0 \\ & x_e \quad = 0 && e \in D \cup R \\ & x_e \quad \geq 0 && e \in M. \end{aligned}$$

We prove this by showing that any feasible solution (x, ε) of (LP_2^*) is also feasible for $(\widehat{\mathrm{LP}}_2)$ (the reverse implication is clear). As $\varepsilon_2 > \varepsilon_1 = 0$, we may assume $\varepsilon > 0$. Let (x, ε) be a solution of (LP_2^*). We need to show that $x(A) \geq (1+\varepsilon)v(A)$ for all $A \in 2^E \backslash \langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle$.

Assume $A \in 2^E \backslash \langle \mathcal{A}_0 \cup \mathcal{A}_1 \}$, say $v(A) = \ell$, i.e., there exist ℓ edge-disjoint $s-t$ paths $\overline{P}_1, \ldots, \overline{P}_\ell \subseteq A$. If $\ell = k$, then $\overline{P}_1, \ldots, \overline{P}_k$ form a max flow set. By Observation 1, we obtain $M \subseteq P_1 \cup \ldots \cup P_k$ and $M \subseteq \overline{P}_1 \cup \ldots \cup \overline{P}_k$. The latter implies $M \subseteq A$. Since $x \equiv 0$ on $D \cup R$, we then find $x(A) = x(M) = x(\bigcup_{i=1}^k P_i) = k$. Then $A \in \langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle$ holds, a contradiction. Hence we may assume $\ell \leq k-1$.

If $\ell = 0$, there is nothing to show (as $v(A) = \ell = 0$). Hence assume $\ell \geq 1$. First, we prove a lower bound on $x(\widetilde{P})$ for all $\widetilde{P} \in \mathcal{P}_1$. Recall that $\widetilde{P}$ makes part of a critical coalition: assume $\widetilde{P} = P_\ell^{si} \cup \overline{R}^{ij} \cup P_m^{jt}$, then

$$x(A_{\ell m}^{ij}) = x(\widetilde{P} \cup \bigcup_{h \neq \ell, m} P_k) = x(\widetilde{P}) + k - 2 \geq (1+\varepsilon)(k-1).$$

So we find that $x(\widetilde{P}) \geq 1 + \varepsilon(k-1)$ for all $\widetilde{P} \in \mathcal{P}_1$. We use this information to deduce a lower bound on $x(P)$ for each $P \in \{\overline{P}_1, \ldots, \overline{P}_\ell\}$. We can decompose such a P as in (2.1) and apply Lemma 1. If $q = 1$ we find

$$x(P) = 1. \tag{2.2}$$

If $q \geq 2$ we find

$$x(P) \begin{cases} = 1 & \text{if and only if all } \widetilde{P}_\rho \in \mathcal{P}_0 \\ \geq 1 + \varepsilon(k-1) & \text{if exactly one } \widetilde{P}_\rho \in \mathcal{P}_1 \\ \geq 1 + 2\varepsilon(k-1) & \text{else.} \end{cases} \tag{2.3}$$

First assume none of $\overline{P}_1, \ldots, \overline{P}_\ell$ intersect R. Then, by (2.2) and (2.3), $x(\overline{P}_i) = 1$ and $\overline{P}_i \in \langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle$ for all i. Hence $A \backslash (\overline{P}_1 \cup \ldots \cup \overline{P}_\ell)$ must contain an edge $e \in M$ (as $A \notin \langle \mathcal{A}_0 \cup \mathcal{A}_1 \rangle$ and $x \equiv 0$ on $D \cup R$). Since $v(A) = \ell$, we find that

e is not an edge from s to t. Then A_e is a critical coalition. Since x satisfies all critical constraints, we have $x(A_e) \geq (1+\varepsilon)(k-1)$, i.e.,

$$x_e + k - 1 = x_e + \sum_{h \neq m} x(P_h) \geq (1+\varepsilon)(k-1),$$

so $x_e \geq \varepsilon(k-1) > 0$ (as a matter of fact we have proven this way that $x > 0$ on M) and, as required,

$$x(A) \geq x_e + x(\overline{P}_1) + \ldots + x(\overline{P}_\ell) \geq \varepsilon(k-1) + \ell \geq (1+\varepsilon)\ell.$$

Second, assume that at least one of the paths, say, $\overline{P}_1$ intersects R. By (2.3), we obtain $x(\overline{P}_1) \geq 1 + \varepsilon(k-1)$ and $x(\overline{P}_i) \geq 1$ for $i = 2, \ldots, \ell$. Then, as required,

$$x(A) \geq x(\overline{P}_1) + x(\overline{P}_2) + \ldots + x(\overline{P}_\ell) \geq \ell + \varepsilon(k-1) \geq (1+\varepsilon)\ell.$$

The above actually shows more than what is claimed. Indeed, we see that the coalition A can get tight only if $\ell = k-1$. Furthermore, in the first case, i.e., when $(\overline{P}_1 \cup \ldots \cup \overline{P}_{k-1}) \cap R = \emptyset$, the set $A \backslash (\overline{P}_1 \cup \ldots \cup \overline{P}_{k-1})$ may contain only a unique edge $e \in M$ (otherwise $x > 0$ on M would imply $x(A) > (1+\varepsilon)(k-1)$). So A gets tight (and fixed) exactly when A_e gets tight, either in $(\widehat{\mathrm{LP}}_2)$ or in subsequent $(\widehat{\mathrm{LP}}_r)$ for $r \geq 3$. A similar argument applies when $(\overline{P}_1 \cup \ldots \cup \overline{P}_{k-1}) \cap R \neq 0$. In this case, A can only get tight if there is exactly one path $P \in \{\overline{P}_1, \ldots, \overline{P}_k\}$ that intersects R and, in addition, there is exactly one $\widetilde{P}_\rho$ in (2.3) with $\widetilde{P}_\rho \in \mathcal{P}_1$. Finally, A may not contain any edge $e \in M$ outside $\overline{P}_1 \cup \ldots \cup \overline{P}_{k-1}$. So, again, A can only get tight when the corresponding critical coalition $A^{ij}_{\ell m}$ gets tight (in some $(\widehat{\mathrm{LP}}_r)$, $r \geq 2$). But when $A^{ij}_{\ell m}$ gets tight in, say $(\widehat{\mathrm{LP}}_r)$, then $\widetilde{P}_\rho$ gets fixed to $1 + \varepsilon_r(k-1)$, fixing A to $x(A) = (1+\varepsilon_r)(k-1)$.

Summarizing, our arguments show that we may completely disregard all constraints $x(A) \geq (1+\varepsilon)v(A)$ in $(\widehat{\mathrm{LP}}_r)$, $r \geq 2$, for which A is non-critical. This amounts to saying that each $(\widehat{\mathrm{LP}}_r)$ has only polynomially many constraints and hence is efficiently solvable. □

3 The f-nucleolus of a flow game

For a game (E, v) we define a *priority function* as a mapping $f : 2^E \to \mathbb{R}^+$ to express a *priority* $f(A)$ given to a coalition A (cf. [9]). Assuming $v \geq 0$ (and v-worths representing profits), we solve the following sequence of linear programs

$$(\mathrm{LP}^f_r) \qquad \begin{aligned} \varepsilon_r &:= \max \varepsilon \\ & x(A) \geq v(A) + \varepsilon f(A) \quad A \in 2^E \backslash \langle \mathcal{A}_0 \cup \ldots \cup \mathcal{A}_{r-1} \rangle \\ & x(A) = v(A) + \varepsilon_i f(A) \quad A \in \mathcal{A}_i \quad (i = 0, \ldots, r-1), \end{aligned}$$

(where $\varepsilon_0 = 0$ and $\mathcal{A}_0 = \{\emptyset, E\}$) for $r = 1, 2, \ldots$ until $f(A) = 0$ holds for all $A \in 2^E \backslash \langle \mathcal{A}_0 \cup \ldots \cup \mathcal{A}_{r-1} \rangle$. We then have found the *f-nucleolus* of (E, v) if $\mathrm{core}(v) \neq \emptyset$ (as otherwise, by definition we have to add in each (LP^f_r) the extra condition that x is an imputation.). Note that the f-nucleolus is not necessarily a single point. We obtain the nucleolus if we choose $f \equiv 1$, the nucleon if $f = v$ and the per-capita nucleolus if $f(A) = |A|$ for all $A \subseteq E$. Observe that these three priority functions have the property that $f(A) > 0$ whenever $v(A) > 0$. This is a natural condition and we call such a priority function *suitable*.

Below we present the main result of this section. The gadget used in the NP-hardness reduction has been inspired by the gadget Deng, Fang and Sun [4] used for proving NP-hardness of computing the nucleolus, but the arguments we use are completely different.

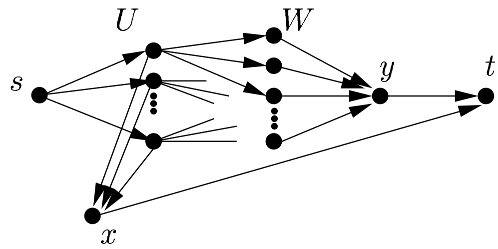

Fig. 2. The graph G given an instance (U, W) of Exact 3-Cover.

Theorem 5. *For any suitable priority function f, computing an allocation in the f-nucleolus is* NP*-hard for the class of flow games.*

Proof. We use reduction from the NP-complete problem Exact 3-Cover [11]. So we are given an undirected bipartite graph I defined by bipartite classes U, W with $|U| = m > q$ and $|W| = 3q$ for some integer q such that each vertex $u \in U$ is adjacent to exactly three vertices in W. The problem is to decide if U contains an *exact cover*, i.e., a set $C \subseteq U$ with $|C| = q$ such that each $w \in W$ is adjacent to some $u \in C$. We may assume $|N(w)| \geq 2$ for all $w \in W$ (cf. [9]).

From I we construct a flow network $G = (V, E)$ as follows (cf. Figure 2). We add four new vertices: a source s, a sink t and two other vertices x, y. We let $E = E_0 \cup E_1 \cup E_2 \cup E_3 \cup \{a, b\}$, where $a = (y, t)$ and $b = (x, t)$ and

$E_0 = \{(s, u) \mid u \in U\}$;
$E_1 = \{(u, w) \mid uw \in E_I\}$; (we directed the edges in E_I from U to W)
$E_2 = \{(w, y) \mid w \in W\}$;
$E_3 = \{(u, x) \mid u \in U\}$.

We define capacities $c(a) = 3q$, $c(b) = 3m - 3q$, $c \equiv 3$ on $E_0 \cup E_3$ and $c \equiv 1$ on $E_1 \cup E_2$. We observe that $v(E) = 3m$. Furthermore, $\varepsilon_1 = 0$, and $x \equiv 0$ on $E_1 \cup E_3$ for any feasible solution $(x, 0)$ of (LP_1^f). This can be seen as follows. Let e be an arbitrary edge in $E_1 \cup E_3$. By construction of G (recall each $w \in W$ has degree at least two in I), we can choose a coalition $A \subseteq E \backslash \{e\}$ with $v(A) = 3m$. We deduce $3m = x(E) \geq x(A) \geq 3m + \varepsilon_1 f(A)$. Since $v(A) > 0$, by definition, $f(A) > 0$, and we obtain $\varepsilon_1 \le 0$. Since flow games have a nonempty core [20], we then find $\varepsilon_1 = 0$. Now let $(x, 0)$ be a feasible solution of (LP_1^f). Then $x(e) = 0$ follows from $x \ge 0$ together with $x(\{e\}) = x(\{e\} \cup A) - x(A) \le x(E) - v(A) = 3m - 3m = 0$. Hence, indeed $x \equiv 0$ on $E_1 \cup E_3$.

We are now ready to formulate our claim which immediately shows that computing an allocation in the f-nucleolus is NP-hard. For an allocation x, we define $\min^*(x) = x_b + \min\{x(E_0') \mid E_0' \subset E_0 \text{ and } |E_0'| = m - q\}$.

Claim. The set U contains an exact cover if and only if $\min^*(x^f) = 3m - 3q$ for each allocation x^f in the f-nucleolus of (E, v).

Note that computing $\min^*(x)$ for a given allocation x is easy. So indeed we are done after proving this claim. The proof goes as follows. Suppose U contains an exact cover C. Let $E_C \subset E_0$ consist of all edges (s, u) for $u \in C$, and let $E_{\overline{C}} = E_0 \backslash E_C$. Note $|E_{\overline{C}}| = m - q$. Define $A = E_C \cup E_1 \cup E_2 \cup \{a\}$ and $B = E_{\overline{C}} \cup E_3 \cup \{b\}$. Then $v(A) = 3q$ and $v(B) = 3m - 3q$. Now let x^f be an allocation in the f-nucleolus of (E, v). Then $(x^f, 0)$ is a feasible solution of (LP_1^f), and consequently $x^f \equiv 0$ on $E_1 \cup E_3$. Then $3m = x^f(E) = x^f(A) + x^f(B) \ge v(A) + v(B) = 3q + 3m - 3q = 3m$. Hence $x^f(A) = 3q$ and $x^f(B) = 3m - 3q$. Since $x^f \equiv 0$ on E_3, we obtain $x_b^f + x^f(E_{\overline{C}}) = x^f(B) = 3m - 3q$. For any other $E_0' \subset E_0$ with $|E_0'| = m - q$ we find $x_b^f + x^f(E_0') = x^f(E_0' \cup E_3 \cup \{b\}) \ge v(E_0' \cup E_3 \cup \{b\}) = 3m - 3q$. Hence $\min^*(x^f) = 3m - 3q$.

To prove the reverse implication suppose U does not contain an exact cover. We show that (LP_2^f) can be formulated as

$$
(\mathrm{LP}_2^f) \qquad
\begin{array}{ll}
\varepsilon_2 := \max \varepsilon & \\
x(A) \ge v(A) + \varepsilon f(A) & A \in 2^E \backslash \langle E, \{e\}_{e \in E_1 \cup E_3} \rangle \\
x(A) = 0 & A \subseteq E_1 \cup E_3 \\
x(E) = 3m. &
\end{array}
$$

Then, for any x^f in the f-nucleolus of (E, v) and any $E_0' \subset E_0$ with $|E_0'| = m - q$, we find

$$
\begin{aligned}
x_b^f + x^f(E_0') &= x^f(E_0' \cup E_3 \cup \{b\}) \\
&\ge v(E_0' \cup E_3 \cup \{b\}) + \varepsilon_2 f(E_0' \cup E_3 \cup \{b\}) \\
&= 3m - 3q + \varepsilon_2 f(E_0' \cup E_3 \cup \{b\}) \\
&> 3m - 3q,
\end{aligned}
$$

and consequently $\min^*(x^f) > 3m - 3q$ (note $f(E_0' \cup E_3 \cup \{b\}) > 0$ because $v(E_0' \cup E_3 \cup \{b\}) > 0$).

We now define allocation x by $x_a = 3q - 3q\beta - q\alpha - \delta$, $x_b = 3m - 3q - (m - q)\alpha + \delta$, $x \equiv \alpha$ on E_0, $x \equiv 0$ on $E_1 \cup E_3$ and $x \equiv \beta$ on E_2 for sufficiently small $\alpha > 0$ and sufficiently small $0 < \beta, \delta < \alpha$ and are done after showing that $(x, 0)$ is a feasible solution of (LP_1^f) with $x(A) > v(A)$ for all $A \notin \langle E, \{e\}_{e \in E_1 \cup E_3} \rangle$.

So, let A be a coalition not in $\langle E, \{e\}_{e \in E_1 \cup E_3} \rangle$. Then $x(A) > 0$ for sufficiently small $\alpha, \beta, \delta > 0$. So, if $v(A) = 0$ then $x(A) > 0 = v(A)$. Suppose $v(A) > 0$. Then $\{a, b\} \cap A \neq \emptyset$. First assume $a \in A, b \notin A$. Since $x^f \equiv 0$ on E_1, we may without loss of generality assume $E_1 \subset A$. Let $E_0 \cap A = E_0'$ and let $E_2 \cap A = E_2'$. If $|E_0'| \leq q$ then $v(A) \leq 3q - 1$ (as otherwise U would have an exact cover) and we find

$$\begin{aligned} x(A) - v(A) &\geq \alpha|E_0'| + \beta|E_2'| + 3q - 3q\beta - q\alpha - \delta - (3q - 1) \\ &= 1 + \alpha(|E_0'| - q) + \beta(|E_2'| - 3q) - \delta > 0, \end{aligned}$$

if we have chosen $\alpha, \beta, \delta > 0$ sufficiently small. If $|E_0'| \geq q + 1$ then $v(A) \leq 3q$, and we find

$$\begin{aligned} x(A) - v(A) &\geq \alpha|E_0'| + \beta|E_2'| + 3q - 3q\beta - q\alpha - \delta - 3q \\ &= \alpha(|E_0'| - q) + \beta(|E_2'| - 3q) - \delta > 0, \end{aligned}$$

if we have chosen $\alpha > 0$ and $0 < \beta, \delta < \alpha$ sufficiently small.

Secondly, assume $a \notin A, b \in A$. Since $x^f \equiv 0$ on E_3, we may without loss of generality assume $E_3 \subset A$. Let $E_0 \cap A = E_0'$. If $|E_0'| \leq m - q - 1$, then

$$\begin{aligned} x(A) - v(A) &= \alpha|E_0'| + 3m - 3q - (m - q)\alpha + \delta - 3|E_0'| \\ &\geq 3 + \alpha(|E_0'| - m + q) + \delta > 0, \end{aligned}$$

if we have chosen $\alpha > 0$ sufficiently small. If $|E_0'| \geq m - q$, then

$$x(A) - v(A) = \alpha|E_0'| + 3m - 3q - (m - q)\alpha + \delta - (3m - 3q) \geq \delta > 0.$$

If both $a, b \in A$, then $v(A) \leq 3m - 1$ as otherwise $A \in \langle E, \{e\}_{e \in E_1 \cup E_3} \rangle$. Hence

$$\begin{aligned} x(A) - v(A) &\geq x_a + x_b - 3m + 1 \\ &= 3q - 3q\beta - q\alpha - \delta + 3m - 3q - (m - q)\alpha + \delta - 3m + 1 \\ &= -3q\beta - m\alpha + 1 > 0 \end{aligned}$$

for α, β sufficiently small. This completes our proof. $\square$

Due to Theorem 5 we immediately find the following.

Corollary 1. *Computing the nucleolus, the per-capita nucleolus and the nucleon are* NP*-hard problems for the class of flow games.*

4 Conclusions

We presented a new combinatorial method by which we obtain efficient algorithms for computing the core, nucleolus, and nucleon of simple flow games. We also showed that for (general) flow games computing an allocation in the f-nucleolus is NP-hard for all suitable priority functions f. As a consequence, computing the nucleolus, per-capita nucleolus and nucleon are NP-hard problems for the class of flow games. This generalizes the NP-hardness result in [4] for the nucleolus. The following questions are interesting.

1. Is computing the per-capita nucleolus, or more generally, the f-nucleolus of a simple flow game polynomially solvable for all suitable priority functions f?

2. Can the class of simple flow games be extended to a larger subclass of flow games for which efficient algorithms exist for computing the core, nucleolus, or nucleon?

Answering question 1 might not be an easy task. A similar study for matching games is still unfinished. Computing the nucleon of a matching game can be done in polynomial time [7]. However, the complexity of computing the nucleolus of a matching game is still a wide open problem.

As the nucleolus for matching games defined on an undirected graph $G = (V, E)$ with edge weights $w(u, v) = w(u) + w(v)$ for vertex weights $w \in \mathbb{R}_+^V$ can be efficiently computed [24], a candidate for an answer to question 2 might be the subclass of flow games that are defined on a network $G = (V, E)$ with edge capacities $c(u, v) = c(u) + c(v)$ for vertex capacities $c \in \mathbb{R}_+^V$.

We would like to thank Eilon Solan and the two anonymous referees for all their useful comments which much helped us to improve the paper. We especially thank the one anonymous referee for pointing us at the fact that coalitions in $\mathcal{P}_1$ do not have to be $s-t$ path coalitions.

References

1. J.M. Bilbao, *Cooperative Games on Combinatorial Structures*, Kluwer Academic, Norwell, Massachusetts, 2000.
2. C.G. Bird, On cost allocation for a spanning tree, a game theoretic approach, *Networks 6* (1976) 335–350.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein (2001). Introduction to Algorithms, 2nd edition, the MIT Press, Cambridge, MA.
4. X. Deng, Q. Fang, and X. Sun Finding Nucleolus of Flow Game, In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms* (SODA 2006), 124–131, 2006.
5. J. Edmonds and R.M. Karp Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM 19* (1972) 248–264.
6. U. Faigle, W. Kern, S, Fekete, and W. Hochstättler, On the complexity of testing membership in the core of min-cost spanning tree games, *Int. J. Game Theory 26* (1997) 361–365.
7. U. Faigle, W. Kern, S, Fekete, and W. Hochstättler, The nucleon of cooperative games and an algorithm for matching games, *Math. Program. 83* (1998) 195–211.
8. U. Faigle, W. Kern, and J. Kuipers, Computing the nucleolus of min-cost spanning tree games is NP-hard, *Int. J. Game Theory 26* (1997) 361–366.
9. U. Faigle, W. Kern and D. Paulusma, Note on the computational complexity of least core concepts for min-cost spanning tree games, *Math. Methods Oper. Res. 52* (2000) 23–38.
10. Q. Fang, S. Zhu, M. Cai, and X. Deng, On the computational complexity of membership test in flow games and linear production games, *Int. J. Game Theory 31* (2002) 39–45.
11. M. Garey and D. Johnson, *Computers and Intractability*, Freeman, San Francisco, 1979.

12. D.B. Gillies, Solutions to general non-zero-sum games, In *Contributions to the Theory of Games IV*, Annals of Mathematics Studies 40, Princeton University Press, Princeton, New Jersey, 47–85, 1959.
13. Q. Fang, R. Fleischer, J. Li, and X. Sun, Algorithms for core stability, core largeness, exactness, and extendability of flow games, In *Proceedings of the 13th Annual International Computing and Combinatorics Conference* (COCOON 2007), Lecture Notes of Computer Science 4598, 439–447, 2007.
14. S. Fortune, J.E. Hopcroft, J. Wyllie, The directed subgraph homeomorphism problem, *Theoret. Comput. Sci. 10* (1980) 111-121.
15. D. Granot, F. Granot, On some network flow games, *Math. Oper. Res. 17* (1992) 792–841.
16. D. Granot, F. Granot, and W.R. Zhu, The reactive bargaining set of some flow games and of superadditive simple games, *Int. J. Game Theory 26* (1997) 207–214.
17. J.H. Grotte, *Computations of and computations on the nucleolus and the central games*, M.Sc. Thesis, Cornell University, 1970.
18. G. Huberman, The nucleolus and the essential coalitions, In *Analysis and Optimization of Systems*, Springer, Berlin, 416-422, 1980.
19. E. Kalai and E. Zemel, Totally Balanced Games and Games of Flow, *Math. Oper. Res. 7* (1982) 476–478.
20. E. Kalai and E. Zemel, Generalized network problems yielding totally balanced games, *Oper. Res. 30* (1982) 498–1008.
21. W. Kern and D. Paulusma, Matching games: the least core and the nucleolus, *Math. Oper. Res. 28* (2003) 294–308.
22. M. Maschler, B. Peleg, L.S. Shapley, Geometric properties of the kernel, nucleolus, and related solution concepts, *Math. Oper. Res. 4* (1979) 303-338.
23. G. Owen, *Game Theory*, Academic Press, San Diego, California, 1995.
24. D. Paulusma, *Complexity Aspects of Cooperative Games*, PhD-Thesis, University of Twente, Enschede, the Netherlands, 2001.
25. J. Potters, H. Reijnierse, and A. Biswas, The nucleolus of balanced simple flow networks, *Games Econom. Behav. 54* (2006) 205–225.
26. D. Schmeidler, The nucleolus of a characteristic function game, *SIAM J. Appl. Math. 17* (1969) 1163–1170.
27. L.S. Shapley and M. Shubik, Quasi-cores in a monetary economy with nonconvex preferences, *Econometrica 34* (1966) 805–827.
28. L.S. Shapley and M. Shubik, The assignment game I: the core. *Int. J. Game Theory 1* (1972) 111–130.
29. S.H. Tijs and T.S.H. Driessen, Extensions of solution concepts by means of multiplicative ε-tax games, *Math. Social Sci. 12* (1986) 9–20.
30. T. Solymosi and T.E.S. Raghavan, An algorithm for finding the nucleolus of assignment games, *Int. J. Game Theory 23* 119–143.
31. E. Wallmeier, *Der f-Nukleolus und ein Dynamisches Verhandlungsmodell als Lösungskonzepte für kooperative n-Personenspiele*, Ph.D. Thesis, Münster, 1983.
32. G.J. Woeginger, On the rate of taxation in a cooperative bin packing game, *Methods Models Oper. Res. 42* (1995) 313–324.
33. H.P. Young, Monotonic solutions to cooperative games, *Int. J. Game Theory 14* (1985) 65–72.
34. H.P. Young, N. Okada, T. Hashimoto, Cost allocation in water resources development, *Water Res. Res. 18* (1982) 463–475.