The nucleolus of balanced simple flow networks

Jos Potters [a,*], Hans Reijnierse [b], Amit Biswas [c]

[a] *Mathematical Department RUN, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands*
[b] *CentER and Department of Econometrics, Tilburg University, PO Box 90153, 5000 LE Tilburg, The Netherlands*
[c] *Indian Statistical Institute, 110, Nelson Manickam Road, Aminjikarai, Chennai 600 029, India*

Abstract

This paper gives an algorithm for the nucleolus of simple flow games with directed and undirected, private as well as public arcs, under the condition that the flow game has a *nonempty core*.

JEL classification: C63; C71

Keywords: Cooperative game theory; Nucleolus; Core; Balancedness; Flow network; Potential

Introduction

Flow games are introduced in Kalai and Zemel (1982). They proved that flow games (without public arcs) are totally balanced and, conversely, that every nonnegative totally balanced game can be derived from a flow network in which every arc is private. Curiel et al. (1989) studied flow networks with coalitionally controlled undirected arcs and proved that these TU-games are balanced and that each nonnegative balanced game can be obtained as a flow game with veto control. More recent sources, important for the present

* Corresponding author.
E-mail addresses: J.Potters@math.ru.nl (J. Potters), J.H.Reijnierse@uvt.nl (H. Reijnierse), amit_biswas@hotmail.com (A. Biswas).

paper, are Reijnierse et al. (1996) and Granot and Granot (1992). We repeat their results in as far as they are relevant for this paper.

(i) (Reijnierse et al., 1996) The core of a simple flow game (N, v_f) is nonempty if and only if there is a minimum cut without public arcs. In that case, the extreme points of the core are in one-to-one correspondence with the minimum cuts without public arcs: if C is a minimum cut without public arcs, the allocation assigning to each player controlling an arc in the minimum cut a payoff one and zero to the other players, is an extreme point of the core and in this way all extreme points of the core are obtained. For simple flow games without public arcs this result can already be found in Kalai and Zemel (1982).
(ii) (Reijnierse et al., 1996) A simple flow game is totally balanced if it is superadditive. This is the case if and only if the capacity of all public arcs can be increased without increasing the maximum flow for *any* coalition. Public arcs are never 'bottlenecks'. For a superadditive simple flow game *the bargaining set is equal to the core*.
(iii) (Granot and Granot, 1992) For simple flow games *without directed arcs and without public arcs*—these games are automatically totally balanced—the nucleolus is the *lexicographically maximal* element of the core. For simple flow games of this type *the kernel is a subset of the core* (by (ii)) and, in fact, a *convex polytope*. The results under (iii) are not true if the arcs are directed.

This paper shows how the nucleolus of the most general kind of a simple flow game can be computed, as long as the game has a nonempty core. For this purpose, a collection of coalitions $\mathcal{P}^{(1)}$ is given that determines the nucleolus if the set of candidates is restricted to the core. Thereafter, *potential functions* are defined on a subset of the vertices of the network. These potentials turn out to be in one-to-one correspondence with the core elements. A *modified digraph* is defined as a tool to find the *lexicographically maximal* potential function. This potential corresponds to the nucleolus of the original flow network.

The paper is organized as follows. In the next section the necessary preliminaries are given. Section 2 gives a result concerning the core, the nucleolus and the kernel of a simple flow game. Furthermore, it gives a generalization of a known result about collections determining the nucleolus. The result is used in Section 3 to show that the collection $\mathcal{P}^{(1)}$ determines the nucleolus. Section 4 introduces potentials and the modified digraph g of a simple flow network f. Section 5 defines a nucleolus on g and proves that it corresponds to the nucleolus of v_f, the cooperative game corresponding to f. The last sections have a constructive nature; Section 6 gives the construction of the digraph g and Section 7 gives an algorithm to compute $\mathrm{Nu}(g)$, the nucleolus of g. Finally, Section 8 discusses the complexity of the calculations.

1. Introduction of the basic tools

A *directed graph* or *digraph* $\langle V, E, \alpha\rangle$ consists of a set V of *vertices*, a set E of *arcs* and a map $\alpha : E \to V \times V$. If $\alpha(e) = (a, b)$, the arc e has begin-point *(tail)* a and endpoint *(head)* b. It is assumed that there are no loops: for every $e \in E$ and $a \in V$, $\alpha(e) \neq (a, a)$. To define a flow network with private control three more ingredients are added:

- $\text{cap}: E \to \mathbb{R}_{++}$ is a map assigning to each arc e a (strictly positive) capacity of $\text{cap}(e)$.
- Two different vertices s and t are indicated, called the *source* and the *sink*.
- N is a set of players and $\sigma : N \twoheadrightarrow E$ is a multifunction that assigns to each player the arcs *under his control*.

We assume that $\sigma(i) \cap \sigma(j) = \emptyset$ if $i \neq j$; an arc is not controlled by two different players. Arcs not under control of any player are called *public*; they can be used freely by any (group of) player(s). The 'inverse' of σ is called S, so for every $Q \subseteq E$, $S(Q)$ is the (possibly empty) set of players i with $\sigma(i) \cap Q \neq \emptyset$. To avoid trivial examples, the existence of a (directed) path from source to sink is assumed. So, a flow network is given by:

$$f := \langle V, s, t, E, \alpha, \text{cap}, N, \sigma \rangle.$$

If all capacities are *one* and each player controls one arc or two oppositely directed arcs between the same nodes, the flow network is called *simple*. In this context, a pair of oppositely oriented arcs between the same vertices have the same opportunities as an undirected edge between the same nodes, as we shall see.

A *flow* in a flow network $\langle V, s, t, E, \alpha, \text{cap} \rangle$ is a map $X: E \to \mathbb{R}$ with the property that $X(e) \geqslant 0$ for all arcs $e \in E$. A flow X is called *feasible* if $X(e) \leqslant \text{cap}(e)$ for all arcs $e \in E$. An arc e is called X-*used* if $X(e) > 0$. The set of X-used arcs is called E_X. The endpoints of the X-used arcs form the set V_X (so V_X is the set of vertices that X visits).

The *in-flow* of a flow X into a vertex $b \in V$ is the sum of the flow $X(e)$ over all arcs e with head b. The *out-flow* of X from a vertex a is defined analogously. It is the sum of flow $X(e)$ over all arcs e with tail a.

A vertex $a \in V$ is a *transition point* of a flow X if the in-flow of a equals its out-flow. If all vertices but s and t are transition points and if the out-flow of the source exceeds its in-flow, X is called a *flow from s to t* or *s-t-flow*. The difference between the out-flow of s and its in-flow is called the *value* of X. An s-t-flow X is called a *maximum flow* if the flow is feasible and there is no feasible s-t-flow with a larger value.

If $S \subseteq N$ is a *coalition* of players, we define $v_f(S)$ as the value of a maximum flow in the network consisting of the arcs controlled by the players in S, and the public arcs. We assume that there is no flow that uses only public arcs i.e., $v_f(\emptyset) = 0$, in order to have that the set of players N and the map v_f form the TU-game (N, v_f). Because of this assumption, every source to sink path uses at least one private arc. In the case of a simple flow network, this limits the value of a maximal flow to $|N|$. Therefore, it is harmless to assume additionally that in a simple flow network, for every pair of nodes (a, b), there are at most $|N|$ arcs with head a and tail b. This avoids an excessive amount of superfluous public arcs. Because we assume that there is at least one path from source to sink, we have $v_f(N) > 0$.

The TU-game (N, v_f) is a tool to find an appropriate allocation of the value of a maximum flow of a network f among the players in N. Solution concepts developed for general TU-games will be used. The nucleolus $\text{Nu}(v_f)$ will have the most attention but also the kernel and the core will show up in Section 2. We will not repeat the definitions of these concepts as they are well known nowadays. For the general nucleolus (Maschler et al., 1992) an exception is made.

Let Π be a compact convex subset of the pre-imputation set of a TU-game (N, v) and let $\mathcal{B}$ be a collection of coalitions in N. The *excess map* $\mathrm{Exc}: \Pi \to \mathbb{R}^{\mathcal{B}}$ is defined by its coordinates: $\mathrm{Exc}_S(x) := \mathrm{exc}(S, x) = v(S) - x(S)$ for $x \in \Pi$ and $S \in \mathcal{B}$. The map $\theta : \mathbb{R}^{\mathcal{B}} \to \mathbb{R}^{|\mathcal{B}|}$ orders the coordinates of a vector in $\mathbb{R}^{\mathcal{B}}$ in a weakly decreasing order. The *general nucleolus of* (N, v) *with respect to* Π *and* $\mathcal{B}$ is defined by:

$$\mathrm{Nu}(\Pi, \mathcal{B}) := \{x \in \Pi\colon\ \theta \circ \mathrm{Exc}(x) \preccurlyeq_{\mathrm{lex}} \theta \circ \mathrm{Exc}(y) \text{ for all } y \in \Pi\}.$$

For example, the nucleolus is (the element of) $\mathrm{Nu}(I(v), 2^N \setminus \{\emptyset\})$, in which $I(v)$ denotes the imputation set.

From here we assume that f is a simple flow network. Furthermore, we only consider integer-valued (maximum) s-t-flows without circuits i.e., if the arcs $e_1, e_2, \ldots, e_q$ form a circuit (a directed cycle) in the graph, the flow along at least one of the arcs e_i equals zero. If a flow has circuits, we can diminish the flow along the circuit without diminishing the value of the flow. Flows with circuits use the capacity of the flow network in an inefficient way. In particular, if a player has a pair of oppositely directed arcs, at most one of the arcs is used by a flow without circuits. This means that the possibilities of an undirected edge are the same as a pair of oppositely directed arcs between the same vertices.

A set of arcs $C \subseteq E$ is called a *cut* of network f, if the (value of a) maximum flow in the network obtained by deleting the arcs in C, is zero. A *minimum cut* is a cut C with minimal total capacity. By the well known theorem of Ford and Fulkerson (1956), the capacity of a minimum cut equals the value of a maximum flow in the network f.

In a simple flow network the value of the grand coalition $v_f(N)$ equals the number of arcs in a minimum cut (and is therefore an integer). The same is true for the coalitional values $v_f(S)$.

Every (integer-valued) maximum flow uses an arc at full capacity one or not at all. Furthermore, a maximum flow (without circuits) can be decomposed into $v_f(N)$ simple flows of value one with (pairwise) edge-disjoint carriers. Such a simple flow (unit flow) is using the arcs of a path from s to t (a *unit flow path*) at full capacity and no other arcs. The decomposition can be obtained successively by following the flow from the source to the sink, subtracting the unit flow obtained in this way and following the same procedure with the remaining flow of which the value is one unit lower. Given a flow X, a unit flow path is called an *X-unit flow path* if $X(e) = 1$ for every edge e on the path.

Paths are supposed not to contain circuits (are non-self-intersecting). If Q is any path from source to sink, the coalition $S(Q)$ is called a *path coalition*. $\mathcal{P}$ is the collection of path coalitions.

If X is a maximum flow and Q is an X-unit flow path, then $v_f(S(Q)) \geqslant 1$ and $v_f(N \setminus S(Q)) \geqslant v_f(N) - 1$. If v_f is balanced (or superadditive), the inequalities are in fact equalities. Hence, every core element allocates 1 to coalition $S(Q)$ and $S(Q)$ has excess 0 throughout the core. For players who do not own an X-used arc, there is nothing left; they are assigned 0 by every core element.

2. The core, nucleolus and core-kernel of a simple flow game

Theorem 1. *If f is a simple flow network and the associated TU-game (N, v_f) is balanced, then the core, the nucleolus and the intersection of the core and the kernel are determined by $\mathcal{P}$ (the collection of path coalitions), the 1-coalitions and the grand coalition.*

Proof. (About the core, cf. Reijnierse et al., 1996.) We have to prove that the core of (N, v_f) can be given by the (in)equalities:

$$x_i \geqslant 0, \qquad (i \in N)$$
$$x(N) = v_f(N),$$
$$x(T) \geqslant 1. \qquad (T \in \mathcal{P})$$

Let S be a strict subset of N. Decompose a maximum flow X_S in the flow network controlled by coalition S (inclusive the public arcs) into $v_f(S)$ X_S-unit flow paths. Then S is the disjoint union of the corresponding path coalitions and some 1-coalitions. Hence, the payoff to coalition S is at least $v_f(S)$ times one.

(About the nucleolus.) Hubermann (1980) calls a coalition S *essential* if $v(S) > \sum_{i=1}^{q} v(S_i)$ for all partitions $\{S_1, \ldots, S_q\}$ of S, i.e. the value of S exceeds the sum of the parts. In a simple flow game (N, v_f), all essential coalitions *are* path coalitions or singletons. The paper proves that the nucleolus of a balanced game is determined by the essential coalitions only. This means, if we define the general nucleolus by only using the excesses of essential coalitions and restrict the set of candidates to the core allocations, one finds the standard nucleolus.

(About the core-kernel.) Let x be a core allocation of (N, v_f) and define $\bar{s}_{ij}(x) := \max\{v_f(i) - x_i, 1 - x(T)) \mid T \in \mathcal{P} \text{ with } i \in T \text{ and } j \notin T\}$.

As every coalition S is the disjoint union of path coalitions and 1-coalitions and x is a core allocation, $\bar{s}_{ij}(x) = s_{ij}(x)$ (take the component containing i). The core-kernel is the set of core allocations where $\bar{s}_{ij}(x) = \bar{s}_{ji}(x)$ for all pairs (i, j) with $i \neq j$. Hence, it is determined by $\mathcal{P}$, the 1-coalitions and N. □

Corollary 2. *If (N, v_f) is superadditive, the kernel is determined by the path coalitions and the 1-coalitions.*

Proof. If (N, v_f) is superadditive, the bargaining set is equal to the core (Reijnierse et al., 1996) and, as the kernel is a subset of the bargaining set, the kernel coincides with the intersection of the kernel and the core. □

The part of Theorem 1 concerning the nucleolus can even be sharpened. It is possible to give a further reduction of the collection of coalitions that determines the nucleolus.

For this purpose we need a slight generalization of Theorem 1 in Reijnierse and Potters (1998). We will formulate and prove this result first. Let (N, v) be a TU-game with nonempty imputation set. Let Π be a closed convex subset of the *imputation set* with $\mathrm{Nu}(v) \in \Pi$ and let $\mathcal{B}$ be any nonempty collection of coalitions. Let:

$$\mathcal{Z}(\Pi) := \{Z \subseteq N\colon\ x(Z) \text{ is constant on } \Pi\}$$

and finally, for $S \notin \mathcal{B}$ let:

$$\mathcal{B}_S(\Pi) := \{T \in \mathcal{B}\colon \operatorname{exc}(T, y) \geqslant \operatorname{exc}(S, y) \text{ for all } y \in \Pi\}.$$

Proposition 3 (cf. Reijnierse and Potters, 1998). *Let (N, v) be a TU-game with $I(v) \neq \emptyset$. Let Π be a closed convex set with $\operatorname{Nu}(v) \in \Pi \subseteq I(v)$ and let $\mathcal{B}$ be any nonempty collection of coalitions. Then $\mathcal{B}$ determines the nucleolus inside the set Π if, for every coalition $S \notin \mathcal{B}$, the vector e_S is a nonnegative linear combination of e_T with $T \in \mathcal{B}_S(\Pi)$, plus a linear combination of e_Z with $Z \in \mathcal{Z}(\Pi)$. In formula, $\operatorname{Nu}(\Pi, \mathcal{B}) = \operatorname{Nu}(v)$, if for all $S \notin \mathcal{B}$:*

$$e_S \in \mathbb{R}_+\langle e_T\colon\ T \in \mathcal{B}_S(\Pi)\rangle + \mathbb{R}\langle e_Z\colon\ Z \in \mathcal{Z}(\Pi)\rangle. \tag{1}$$

Proof. Let $x = \operatorname{Nu}(v)$ and $y \in \operatorname{Nu}(\Pi, \mathcal{B})$. Let $\mathcal{U} = \{S \subset N\colon\ x(S) \neq y(S)\}$. We prove that $\mathcal{U} = \emptyset$. Suppose, on the contrary, that $\mathcal{U} \neq \emptyset$.

By definition $\theta \circ \operatorname{Exc}(x) \preccurlyeq_{\text{lex}} \theta \circ \operatorname{Exc}(y)$. Then $\theta \circ \operatorname{Exc}_{\mathcal{U}}(x) \preccurlyeq_{\text{lex}} \theta \circ \operatorname{Exc}_{\mathcal{U}}(y)$. Here, the map $\operatorname{Exc}_{\mathcal{U}}$ assigns to each imputation the vector of excesses of the coalitions in $\mathcal{U}$.

By definition $\theta \circ \operatorname{Exc}_{\mathcal{B}}(y) \preccurlyeq_{\text{lex}} \theta \circ \operatorname{Exc}_{\mathcal{B}}(x)$. Then also, $\theta \circ \operatorname{Exc}_{\mathcal{B}\cap\mathcal{U}}(y) \preccurlyeq_{\text{lex}} \theta \circ \operatorname{Exc}_{\mathcal{B}\cap\mathcal{U}}(x)$.[1]

Both implications rest on the property of θ and $\preccurlyeq_{\text{lex}}$:

If $a, b \in \mathbb{R}^p$ and $c \in \mathbb{R}^q$, then $\theta(a, c) \preccurlyeq_{\text{lex}} \theta(b, c)$ if and only if $\theta(a) \preccurlyeq_{\text{lex}} \theta(b)$.

These two inequalities combined give:

$$\bigvee_{S\in\mathcal{U}\cap\mathcal{B}} \operatorname{exc}(S, y) \leqslant \bigvee_{S\in\mathcal{U}\cap\mathcal{B}} \operatorname{exc}(S, x) \leqslant \bigvee_{S\in\mathcal{U}} \operatorname{exc}(S, x) \leqslant \bigvee_{S\in\mathcal{U}} \operatorname{exc}(S, y),$$

in which $\bigvee_{S\in\mathcal{U}} \operatorname{exc}(S, x)$ is the 'maximum' over the excess functions $\operatorname{exc}(S, x)$ with $S \in \mathcal{U}$. Let $E := \bigvee_{S\in\mathcal{U}} \operatorname{exc}(S, y)$. We prove that the inequalities in the previous relation are, in fact, equalities. Let $\bar{S} \in \mathcal{U}$ be a coalition with $\operatorname{exc}(\bar{S}, y) = E$. If $\bar{S} \in \mathcal{B}$, then $\bigvee_{S\in\mathcal{U}\cap\mathcal{B}} \operatorname{exc}(S, y) = E$ and we are done. If $\bar{S} \notin \mathcal{B}$, we can denote:

$$e_{\bar{S}} = \sum_{T\in\mathcal{B}_{\bar{S}}(\Pi)} \lambda_T e_T + \sum_{Z\in\mathcal{Z}(\Pi)} \mu_Z e_Z, \quad \text{with } \lambda_T \geqslant 0 \text{ and } \mu_Z \in \mathbb{R}.$$

Take the inner product with $x - y$:

$$x(\bar{S}) - y(\bar{S}) = \sum_{T\in\mathcal{B}_{\bar{S}}(\Pi)} \lambda_T\bigl(x(T) - y(T)\bigr) + \sum_{Z\in\mathcal{Z}(\Pi)} \mu_Z\bigl(x(Z) - y(Z)\bigr).$$

As $x(Z) = y(Z)$, the latter summation vanishes. Since $\bar{S} \in \mathcal{U}$, $x(\bar{S}) \neq y(\bar{S})$. Even $x(\bar{S}) > y(\bar{S})$, because $\bigvee_{S\in\mathcal{U}} \operatorname{exc}(S, x) \leqslant \bigvee_{S\in\mathcal{U}} \operatorname{exc}(S, y)$. Therefore there is a coalition $T \in \mathcal{B}_{\bar{S}}(\Pi)$ with $x(T) > y(T)$. Hence, $T \in \mathcal{U} \cap \mathcal{B}$. The definition of $\mathcal{B}_{\bar{S}}(\Pi)$ gives $\operatorname{exc}(T, y) \geqslant \operatorname{exc}(\bar{S}, y) = E$. Accordingly we find:

$$\bigvee_{S\in\mathcal{U}\cap\mathcal{B}} \operatorname{exc}(S, y) = \bigvee_{S\in\mathcal{U}} \operatorname{exc}(S, x) = \bigvee_{S\in\mathcal{U}} \operatorname{exc}(S, y).$$

[1] If $\mathcal{U} \cap \mathcal{B} = \emptyset$, it is easy to see that $\mathcal{U} = \emptyset$. Every characteristic vector e_S with $S \in \mathcal{U}$ is a linear combination of vectors e_T with $T \in \mathcal{B}$ and e_Z with $Z \in \mathcal{Z}(\Pi)$. For all these coalitions $x(T) = y(T)$ and $x(Z) = y(Z)$. Hence, $x(S) = y(S)$.

As the collections of coalitions $\{S \in \mathcal{U}\colon \operatorname{exc}(S,x) = E\}$ and $\{T \in \mathcal{U}\colon \operatorname{exc}(T,y) = E\}$ are disjoint, $\frac{1}{2}(x+y)$ has a lower highest excess in $\mathcal{U}$ and is, therefore, a better candidate for the nucleolus than x. □

Corollary 4. *If $\mathcal{B}$ satisfies the conditions of Proposition 3, every collection $\mathcal{B}' \supset \mathcal{B}$ satisfies them too.*

Proof. Suppose $S \notin \mathcal{B}'$. Then $S \notin \mathcal{B}$ and $\mathcal{B}'_S(\Pi) \supset \mathcal{B}_S(\Pi)$. If e_S can be written as the sum of a positive combination of vectors in $\{e_T\colon T \in \mathcal{B}_S(\Pi)\}$ and a linear combination of vectors in $\{e_Z\colon Z \in \mathcal{Z}(\Pi)\}$, the same linear combination can be used in $\mathcal{B}'$. □

3. Determining the nucleolus by jump coalitions and singletons

This section gives an application of Proposition 3. Let f be a balanced simple flow network and let X be an integer-valued optimal flow in f without circuits. A path J from node a to node b is called a *jump* if:

a and b are elements of V_X,
J does not visit other points in V_X, and
there is no X-unit flow path visiting both a and b.

A path Q from source to sink is a *jump path* if:

$v_f(S(Q)) = 1$, and
Q is the composition of a series Q_- of X-used arcs, a jump J_Q and a second series Q_+ of X-used arcs.

Such a composition is denoted by $Q = Q_- * J_Q * Q_+$. We use $*$ for the concatenation of paths. The players owning an edge on jump path Q form the *jump coalition* $S(Q)$. Let $\mathcal{P}^{(1)}$ be the collection of all jump coalitions and all 1-coalitions. The following theorem says that the nucleolus of (N, v_f) is determined by $\mathcal{P}^{(1)}$. It is proved by applying Proposition 3 in the case that $\Pi = \operatorname{Core}(v_f)$ and $\mathcal{B} = \mathcal{P}^{(1)}$.

Theorem 5. *Inside the core, the nucleolus of a balanced simple flow game is determined by the collection $\mathcal{P}^{(1)}$. In formula:*

$$\operatorname{Nu}(v_f) = \operatorname{Nu}\bigl(\operatorname{Core}(v_f), \mathcal{P}^{(1)}\bigr).$$

Proof. Let $\Pi := \operatorname{Core}(v_f)$ and $\mathcal{B} := \mathcal{P}^{(1)}$. Each coalition S is the disjoint union of path coalitions and 1-coalitions (cf. the proof of Theorem 1). Inside the core, each one of these have a weakly higher excess than S itself. So, we are left to prove that path coalitions are satisfying the condition (1) of Proposition 3.

Similarly, a path coalition $S(Q)$ is the disjoint union of path coalitions with value 1 and 1-coalitions, all having a weakly higher excess than $S(Q)$ itself. So, assume Q to be a path from s to t with $v_f(S(Q)) = 1$. It can be decomposed as:

$$Q = Q_0 * R_1 * Q_1 * \cdots * R_r * Q_r,$$

in which Q_i are paths consisting of X-used arcs and R_i are paths consisting of non-X-used arcs and with only the endpoints in V_X. Note that some of the paths Q_i can be empty. Then the head of R_i is the tail of R_{i+1}. The proof consists of three steps. The first step deals with paths with $r > 1$:

Step (i): Suppose that the decomposition of Q contains r parts R_i with $r \geqslant 2$. For each $k \in \{1, \ldots, r-1\}$, let h_k be the head of R_k and let U_k be an X-unit flow path visiting h_k. Decompose U_k in $U_k^- * U_k^+$, in which U_k^- is the part of U_k from the source to h_k and U_k^+ is the part from h_k to the sink. Define r (possibly self-intersecting) new source to sink paths $T_1, \ldots, T_r$ by:

$$\begin{aligned}
T_1 &:= \quad Q_0 * R_1 * U_1^+\\
T_2 &:= U_1^- * Q_1 * R_2 * U_2^+\\
&\ \vdots\\
T_{(r-1)} &:= U_{(r-2)}^- * Q_{(r-2)} * R_{(r-1)} * U_{(r-1)}^+\\
T_r &:= U_{(r-1)}^- * Q_{(r-1)} * R_r * Q_r
\end{aligned}$$

T_k has at most one series of non-X-used arcs. Each path T_k contains a non-self-intersecting subpath, say T_k'. We have:

$$\sum_{k=1}^{r} e_{S(T_k')} \leqslant e_{S(Q)} + \sum_{k=1}^{r-1} e_{S(U_k)}.$$

Therefore, there are nonnegative integers $a_1, \ldots, a_n$ such that:

$$e_{S(Q)} = \sum_{k=1}^{r} e_{S(T_k')} + \sum_{i=1}^{n} a_i e_i - \sum_{k=1}^{r-1} e_{S(U_k)}.$$

Let $x \in \mathrm{Core}(v_f)$. Then $x(S(U_k)) = 1$ for all k (see the last lines of Section 1). The inner product of the equation above and $-x$ gives:

$$-x\bigl(S(Q)\bigr) = \sum_{k=1}^{r} -x\bigl(S\bigl(T_k'\bigr)\bigr) + \sum_{i=1}^{n} -a_i x_i + r - 1.$$

Since $v_f(S(Q)) = 1$, we have:

$$v_f\bigl(S(Q)\bigr) - x\bigl(S(Q)\bigr) \leqslant \sum_{k=1}^{r} \bigl[v_f\bigl(S\bigl(T_k'\bigr)\bigr) - x\bigl(S\bigl(T_k'\bigr)\bigr)\bigr] + \sum_{i:a_i \geqslant 1} a_i\bigl[v_f(i) - x_i\bigr].$$

Hence, all coalitions at the right hand side have at least excess $\mathrm{exc}(S(Q), x)$. We can apply Proposition 3 to infer that the collection of all path coalitions with at most one series of non-X-used arcs united with the 1-coalitions determines the nucleolus.

Step (ii). Let $Q = Q_0 * R_1 * Q_1$ be a unit flow path with one series of non-X-used arcs. This time it is not allowed to assume that $v_f(S(Q)) = 1$, because the path of a subcoalition with value 1 may contain more than one non-X-used series.

If R_1 is a jump and $v_f(S(Q)) = 1$, then $S(Q)$ is an element of $\mathcal{P}^{(1)}$. If R_1 is a jump and $v_f(S(Q)) > 1$, we have for all $x \in \mathrm{Core}(v_f)$ that:

$$x\big(S(Q_0)\big) \leqslant 1, \quad \text{because } Q_0 \text{ is contained in an } X\text{-unit flow path,}$$
$$x\big(S(R_1)\big) = 0, \quad \text{because } S(R_1) \text{ consists of players without } X\text{-used arcs,}$$
$$x\big(S(Q_1)\big) \leqslant 1, \quad \text{because } Q_1 \text{ is contained in an } X\text{-unit flow path.}$$

Hence, $2 \leqslant v_f(S(Q)) \leqslant x(S(Q)) \leqslant 2$ and $S(Q) \in \mathcal{Z}(\Pi)$.

So, we can assume that R_1 is not a jump: there is an X-unit flow path U containing both endpoints of R_1. There are two options:

(a) U and R_1 have opposite directions,
(b) U and R_1 have equal directions.

In case (a), we denote $U = U_- * U_0 * U_+$ in which U_- is the path from s to the head of R_1, U_0 is the part between the head of R_1 and the tail of R_1 and U_+ is the part of U from the tail of R_1 to t. Define the X-unit flow paths Q' and Q'' by $Q' := Q_0 * U_+$ and $Q'' := U_- * Q_1$. They are not self intersecting, because X is free of circuits. We have:

$$e_{S(Q)} = e_{S(Q')} + e_{S(Q'')} - e_{S(U)} + e_{S(U_0)} + e_{S(R_1)}.$$

The coalitions $S(Q')$, $S(Q'')$ and $S(U)$ have constant excess 0 inside the core and are thereby elements of $\mathcal{Z}(\Pi)$. Let $x \in \mathrm{Core}(v_f)$ and take the inner product of the previous equation and $-x$:

$$-x\big(S(Q)\big) = -1 - 1 + 1 - x\big(S(R_1)\big) - x\big(S(U_0)\big).$$

Therefore:

$$\mathrm{exc}\big(S(Q), x\big) = v_f\big(S(Q)\big) - 1 - x\big(S(R_1)\big) - x\big(S(U_0)\big). \tag{2}$$

If $v_f(S(Q)) = 1$, we see that the excesses of all 1-coalitions contained in $S(U_0) \cup S(R_1)$ are at least the excess of $S(Q)$ and we are done. So, assume that $v_f(S(Q)) \geqslant 2$.

If $\hat{x}(S(R_1)) > 0$ for some $\hat{x} \in \mathrm{Core}(v_f)$, then R_1 must consist of one private arc of, say, player i and X uses his other arc. Then we can assume that U_0 consists of this other arc and $S(R_1) = S(U_0) = \{i\}$. Equation (2) shows that $x_i \geqslant 1/2$ for all $x \in \mathrm{Core}(v_f)$. Because the extreme points of the core are integer valued (cf. Reijnierse et al., 1996), $x_i = 1$ inside the core. Hence, $S(Q) \in \mathcal{Z}(\Pi)$. If $x(S(R_1)) = 0$ for all $x \in \mathrm{Core}(v_f)$, Eq. (2) indicates that $x(S(U_0)) = 1$ inside the core and again $S(Q) \in \mathcal{Z}(\Pi)$. This ends case (a).

In case (b), let U_0 be the X-unit flow path parallel to R_1 and let $Q' := Q_0 * U_0 * Q_1$. Then Q' is an X-unit flow path, so $x(S(Q')) = 1$ inside the core. If $\hat{x}(S(R_1)) > 0$ for some core element $\hat{x}$, then R_1 must consist of one private arc and X uses the other arc of the owner. However, this other arc forms an X-used cycle with U_0, which has been assumed not to exist. So, $x(S(R_1)) = 0$ for all $x \in \mathrm{Core}(v_f)$, resulting in:

$$1 \leqslant x\big(S(Q)\big) = x\big(S(Q_0)\big) + x\big(S(Q_1)\big) \leqslant x\big(S(Q')\big) = 1.$$

Hence, $S(Q) \in \mathcal{Z}(\Pi)$. This finishes case (b).

Step (iii). Finally, assume that $r = 0$. In this case, the coalition $S(Q)$ corresponds to an X-unit flow path and is thereby an element of $\mathcal{Z}(\Pi)$.

Apply Proposition 3 and conclude that $\mathcal{P}^{(1)}$ determines the nucleolus inside $\text{Core}(v_f)$. □

4. Potentials and the modified digraph g

Let $f = \langle V, s, t, E, \alpha, N, \sigma \rangle$ be a simple flow network and let (V, E, α) be the underlying graph. Let X be an integer-valued maximum flow in f without circuits. Take a decomposition $(P_1, \ldots, P_{v_f(N)})$ of X. Let $\mathcal{P}_X := (S(P_1), \ldots, S(P_{v_f(N)}))$; the collection of the corresponding X-unit flow path coalitions. The coalitions in $\mathcal{P}_X$ are pairwise disjoint. Let S_X be the coalition of players who own an arc in E_X. For every element x of $\text{Core}(v_f)$ we define its corresponding *potential* $p_x : V_X \to [0, 1]$ as follows:

$$p_x(a) := \{x(S) \mid S \subseteq T \in \mathcal{P}_X, S \text{ consists of the members of } T \text{ owning an arc between } s \text{ and } a\}.$$

We have to show that the definition does not depend on the choice of T. If the sum of the payments along the path of another element T' of $\mathcal{P}_X$ from s to a is smaller, we can take this path from s to a and continue along T. Then we have an X-unit flow path with total payment less than one, which cannot be inside the core. If it is larger, we switch the roles of T and T' and get the same contradiction.

Proposition 6. *For every $x \in \text{Core}(v_f)$, $p_x(s) = 0$ and $p_x(t) = 1$. Furthermore, for $a, b \in V_X$:*

(1) $p_x(a) \leqslant p_x(b)$ *if $\alpha(e) = (a, b)$ and e is an X-used arc,*
(2) $p_x(a) \geqslant p_x(b)$ *if there exists a route from a to b not using arcs of players in S_X.*

Proof. Because of the assumption of the existence of at least one source to sink path, we have $p_x(s) = 0$ and $p_x(t) = 1$. The first statement about edges is straightforward.

Suppose there are $a, b \in V_X$ and a path from a to b not using arcs of players in S_X. Let S be the union of a path coalition in $\mathcal{P}_X$, restricted from s to a, together with the players owning an arc on the route from a to b, together with a path coalition in $\mathcal{P}_X$, restricted from b to t. Because $v_f(N) = v_f(S_X)$, players with an arc on the route from a to b get zero allocated from x. This gives: $1 \leqslant v(S) \leqslant x(S) = p_x(a) + (1 - p_x(b))$, which gives $p_x(a) \geqslant p_x(b)$. □

Corollary 7. *X-used public arcs have potential difference zero.*

Any function $p : V_X \to [0, 1]$ satisfying $p(s) = 0$, $p(t) = 1$ and properties (1) and (2) is called a *potential function* on V_X.

Proposition 8. *There is a one-to-one correspondence between the core elements of v_f and the potential functions on V_X.*

Proof. Let $p: V_X \to [0, 1]$ be a potential function on V_X. Define $x_p \in \mathbb{R}^N$ by:

$$x_p(i) := 0 \qquad \text{if } i \notin S_X,$$
$$x_p(i) := p(b) - p(a) \quad \text{otherwise.} \quad (\alpha(e) = (a, b) \text{ if } e \text{ is the } X\text{-used arc of } i)$$

We prove that x_p is a core element. By property (1), $x_p(i) \geqslant 0$, so we only have to show that $x_p(T) \geqslant 1$ for every path coalition T with value 1, and that $x_p(N) = v_f(N)$ (cf. Theorem 1).

Let $T \in \mathcal{P}_X$. Let e be an arc used by T ($\alpha(e) = (a, b)$). If e is public, then $p_y(a) = p_y(b)$ for every core element y (Corollary 7). By property (1), the potential increases along each path T in $\mathcal{P}_X$. Therefore, $x_p(T) = p(t) - p(s) = 1$. Since $v_f(N) = |\mathcal{P}_X|$, we have $x_p(N) = v_f(N)$.

Now let T be a path coalition with value 1. Let e be an arc used by T (again $\alpha(e) = (a, b)$). If $p(a) < p(b)$, e must have an owner i in S_X and $x_p(i) = p(b) - p(a)$. Hence, $x_p(T)$ is at least the sum of the increments of the potentials along a unit flow path corresponding to T. Since the total increment is at least one, we have $x_p(T) \geqslant 1$. We conclude that $x_p \in \text{Core}(v_f)$.

To show the existence of a one-to-one correspondence, it is sufficient to show for all core elements x and potentials p on V_X we have: $x_{(p_x)} = x$ and $p_{(x_p)} = p$.

Let i in N and let (a, b) be an arc of i with the convention that we choose the X-used one if available. Then:

$$x_{(p_x)}(i) = p_x(b) - p_x(a) = x(i).$$

Now let v be in V_X. Let P be an X-used path visiting v. Let S be the coalition of players who own an arc on P before v. For i in S, denote the arc of i used by P by e_i. Then:

$$p_{(x_p)}(v) = \sum_{i \in P} x_p(i) = \sum_{i \in P} \big[p\big(\text{head}(e_i)\big) - p\big(\text{tail}(e_i)\big)\big] = p(v) - p(s) = p(v). \qquad \square$$

Define the following equivalence relation on V_X:

$$a \sim b \quad \text{if} \quad p(a) = p(b) \text{ for every potential on } V_X.$$

Equivalence classes are called *components*. The *modified digraph* $\langle V_g, E_g, \alpha_g \rangle$ is defined as follows. The vertices of g are the components of $\sim$.

Let $[a] \neq [b]$ and $([a], [b]) \neq ([s], [t])$. There is a directed edge from vertex $[a]$ to $[b]$ if:

- there is an X-used private arc e with $\alpha(e) = (a', b')$ such that $a' \sim a$ and $b' \sim b$, or
- there is a path consisting of non-X-used arcs in f from an element of $[b]$ to an element of $[a]$, not visiting vertices in V_X.

Because α_g is an injective function, the arcs of g can be identified with their α_g-images.

Since potentials are constant on components, they can also be defined on V_g:

$$\pi_x : V_g \to [0, 1], \qquad [a] \mapsto p_x(a) \quad \text{for every } x \in \text{Core}(v_f).$$

Proposition 6 gives that for every $x \in \mathrm{Core}(v_f)$: $\pi_x([s]) = 0$, $\pi_x([t]) = 1$ and if there is an arc from $[a]$ to $[b]$, then $\pi_x([a]) \leqslant \pi_x([b])$. In the case of an empty core, the set of potentials is empty as well and $[s] = [t]$. Section 6 gives a construction of g.

5. The potential corresponding to the nucleolus

Let Π_g be the polytope of potential functions on V_g. By Proposition 8, there is a bijection from Π_g to $\mathrm{Core}(v_f)$. Let $\mathrm{Nu}(g)$ be the collection of 'lexicographically best' potentials in Π_g, i.e. the smallest difference between two adjacent potentials is *maximized*, then the second smallest is maximized, and so on.

Formally, define the *map of differences* $\Delta : \Pi_g \to \mathbb{R}^{E_g}$ by: $\Delta_e \pi := \pi(b) - \pi(a)$, in which $e = (a, b) \in E_g$. Then:

$$\mathrm{Nu}(g) := \left\{ \pi \in \Pi_g \mid \bar{\theta} \circ \Delta(\pi) \succeq_{\mathrm{lex}} \bar{\theta} \circ \Delta(\pi') \text{ for all } \pi' \in \Pi_g \right\}.$$

Here, $\bar{\theta} : \mathbb{R}^{E_g} \to \mathbb{R}^{|E_g|}$ maps the coordinates of a vector in a weakly *increasing* order.

Theorem 9. *If (N, v_f) is balanced, then* $\mathrm{Nu}(g)$ *is a singleton and its element corresponds to the nucleolus of v_f.*

Proof. To every arc $e \in E_g$ we allocate a coalition S_e such that $\mathrm{exc}(S_e, x_\pi) = \pi([a]) - \pi([b])$ for every potential π on V_g, in which $e = ([a], [b])$. Of course we have to show that such coalitions exist. Let $e = ([a], [b]) \in E_g$.

If there exists a path coalition $T \in \mathcal{P}_X$ that visits vertices a' with $a' \sim a$ as well as b' with $b' \sim b$, let S_e be the set of players in T that have an arc situated after a' as well as before b'. Since $([a], [b]) \neq ([s], [t])$, we have $\pi(b') - \pi(a') < 1$ for *some* potential π. Hence, $v_f(S_e) \leqslant x_\pi(S_e) < 1$. This gives $v_f(S_e) = 0$ and $\mathrm{exc}(S_e, x_\pi) = -x_\pi(S_e) = \pi([a]) - \pi([b])$ for *every* potential π.

Otherwise, let a', b' be such that $a' \sim a$ and $b' \sim b$ and such that there is a path from b' to a' not visiting nodes in V_X. Take a path coalition $T_1 \in \mathcal{P}_X$ that visits b' and a path coalition $T_2 \in \mathcal{P}_X$ that visits a' (so $T_1 \cap T_2 = \emptyset$). Let S_e be the set of players owning an arc on T_1 before b', together with the players owning an arc on the path from b' to a', together with the players owning an arc on T_2 after a'. Because $([a], [b]) \neq ([s], [t])$, we have $\pi(b') + (1 - \pi(a')) < 2$ for *some* potential π. Hence, $v_f(S_e) \leqslant x_\pi(S_e) < 2$. This gives $v_f(S_e) = 1$ and $\mathrm{exc}(S_e, x_\pi) = \pi([a]) - \pi([b])$ for *every* potential on V_g.

Let $\mathcal{B} = \{S_e \mid e \in E_g\}$. Then for every $x \in \mathrm{Core}(v_f)$, $e = ([a], [b]) \in E_g$, we have $\pi_x([b]) - \pi_x([a]) = -\mathrm{exc}(S_e, x)$. Hence, $\theta \circ \mathrm{Exc}(x) = -\bar{\theta} \circ \Delta(\pi_x)$ and:

$$\begin{aligned}
&\mathrm{Nu}(\mathrm{Core}(v_f), \mathcal{B}) \\
&\quad = \left\{ x \in \mathrm{Core}(v_f) \mid \theta \circ \mathrm{Exc}(x) \preccurlyeq_{\mathrm{lex}} \theta \circ \mathrm{Exc}(y) \text{ for all } y \in \mathrm{Core}(v_f) \right\} \\
&\quad = \left\{ x \in \mathrm{Core}(v_f) \mid -\bar{\theta} \circ \Delta(\pi_x) \preccurlyeq_{\mathrm{lex}} -\bar{\theta} \circ \Delta(\pi_y) \text{ for all } y \in \mathrm{Core}(v_f) \right\} \\
&\quad = \left\{ x \in \mathrm{Core}(v_f) \mid \bar{\theta} \circ \Delta(\pi_x) \succeq_{\mathrm{lex}} \bar{\theta} \circ \Delta(\pi') \text{ for all } \pi' \in \Pi_g \right\} \\
&\quad = \left\{ x \in \mathrm{Core}(v_f) \mid \pi_x \in \mathrm{Nu}(g) \right\}.
\end{aligned}$$

Let $Q = Q_- * J * Q_+$ be a jump path with jump J. Let the head of J be a and the tail of J be b. Then either $[a] = [b]$ and the excess of $S(Q)$ is constant zero inside the core, or $e = ([a], [b])$ is an edge of g and $\text{exc}(S(Q), x) = \text{exc}(S_e, x)$ for every core element x. In both cases, $\text{Nu}(\text{Core}(v_f), \mathcal{B}) = \text{Nu}(\text{Core}(v_f), \mathcal{B} \cup \{S(Q)\})$ since it is harmless to include (or exclude) an excess function that is constant, or one that is already there (cf. Maschler et al., 1992, properties $\mathbf{P}_{12}$ and $\mathbf{P}_{13}$).

Hence, $\text{Nu}(\text{Core}(v_f), \mathcal{B}) = \text{Nu}(\text{Core}(v_f), \mathcal{B} \cup \mathcal{P}^{(1)})$. Corollary 4 and Theorem 5 give:

$$\text{Nu}(v) = \text{Nu}\big(\text{Core}(v_f), \mathcal{P}^{(1)}\big) = \text{Nu}\big(\text{Core}(v_f), \mathcal{B} \cup \mathcal{P}^{(1)}\big). \qquad \square$$

6. The construction of g

Let $f = \langle V, s, t, E, \alpha, N, \sigma \rangle$ be an arbitrary simple flow network. We shall prove that the modified graph can be found by the following procedure.

(i) Construct an integer-valued maximum flow X without cycles. This can be done by the algorithm of Ford and Fulkerson (1956). The sets V_X and E_X are found.
(ii) Identify the vertices connected by an X-used public arc. These vertices have the same potential value anyhow.
(iii) Reverse the direction of all non-X-used arcs. The potential difference over any arc in the present network is nonnegative.
(iv) Identify vertices in any circuit of the present graph. Because of (iii), all potential functions are constant on circuits. Circuits in a directed graph can be found by a well-known depth-first algorithm (see e.g. Weiss, 1999, p. 373).
(v) Let V_g be the subset of the current vertices that represent vertices in V_X. If $[a], [b] \in V_g$, if $([a], [b])$ is (at this stadium) not an edge, and if there exists a path from $[a]$ to $[b]$ in the current network visiting only nodes not in V_g, add edge $([a], [b])$. Remove all vertices not in V_g as well as all edges starting at or ending in such a point.
Paths can be detected by the *Floyd–Warshall* algorithm (Papadimitriou and Steiglitz, 1982, p. 132). Let (V_c, E_c) be the digraph after step (iv) has been performed. Let k be in V_g. Define the $V_c \times V_c$ cost matrix C^k by:

$$C^k_{ij} := \begin{cases} 0 & \text{if } (i, j) \in E_c,\ i = k \text{ and } j \notin V_g, \\ 0 & \text{if } (i, j) \in E_c \text{ and } i \notin V_g, \\ 1 & \text{otherwise.} \end{cases}$$

The Floyd–Warshall algorithm finds all cheapest paths. Arc (k, j) is added if the algorithm finds a free path from k to j ($j \in V_g \setminus \{k\}$). The algorithm has to be performed for every k in V_g.
(vi) Remove all loops, double arcs and, if present, the arc $([s], [t])$.

Note that only equivalent vertices are identified during this procedure. We will show that this procedure results into the network g, but let us first give an example.

Example 10. Consider the network f depicted in Fig. 1.

An integer maximum flow X has been chosen. Arcs in E_X and vertices in V_X have been depicted boldly. If a player controls two arcs, they are depicted by one undirected arc. There are three public X-used arcs. By (ii), these arcs are contracted (see Fig. 2).

By now, the distinction between public and private arcs is of no use anymore in the construction of g. We omit the attachments **P**. The following step is to reverse all arcs not in E_X. After this step, two arcs owned by a player in S_X are both directed according to the flow X. Figure 3, showing the network after step (iii), depicts only one of such a pair.

From here, the distinction between X-used and non-X-used arcs can be forgotten but we have to keep in mind which vertices represent elements of V_X. The network contains two circuits (an undirected arc is a circuit on itself), which must be contracted. Figure 4 arises (only one arc of a parallel set has been depicted):

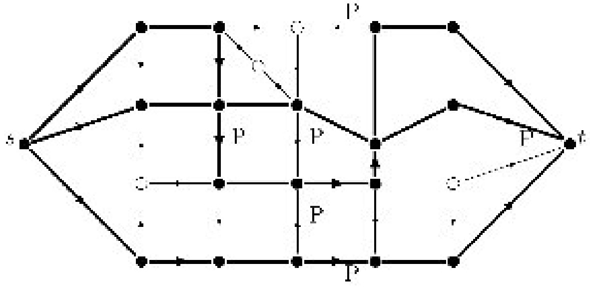

Fig. 1. Network f.

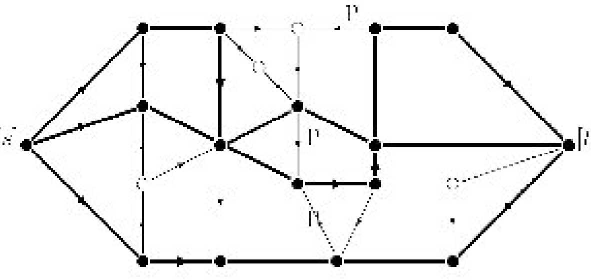

Fig. 2. Step (ii) has been performed.

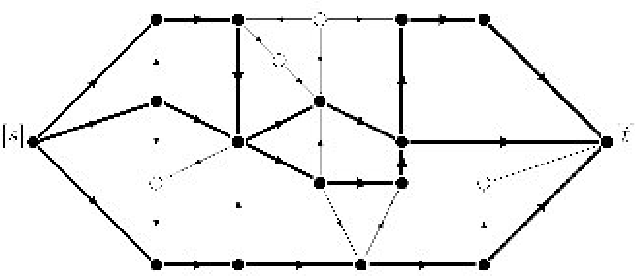

Fig. 3. Step (iii) has been performed.

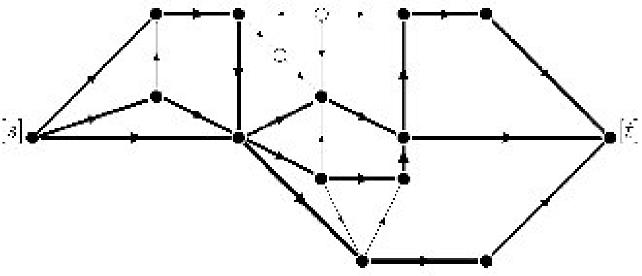

Fig. 4. Two circuits have been contracted.

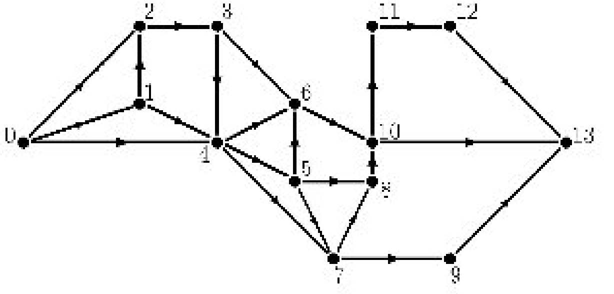

Fig. 5. Network g (with an injective labeling on the nodes).

Only one arc has to be inserted at step (v). Two nodes and their adjacent edges are removed. Finally, at step (vi) superfluous arcs are removed. Figure 5 illustrates the resulting network g.

Proposition 11. *After the operations* (i), ..., (vi), *the resulting network equals* g.

Proof. Let $\bar{f}$ be the digraph resulting from the procedure. $\bar{f}$ is a digraph without circuits.

For all $a \in V_X$, let $[a]'$ be the node of $\bar{f}$ containing a. Since nodes have been identified only if their values coincide for every potential on V_X, the class $[a]'$ is a subset of the class $[a]$.

If $[s]' = [t]'$, there are no potential functions on V_X and the core of the flow game is empty. So, by the procedure it can be decided whether the core of (N, v_f) is empty or not. Hence, we can assume that $[s]' \neq [t]'$.

In a digraph without circuits (like $\bar{f}$) the nodes can be numbered such that each node gets a different number and the number of a is smaller than the number of b whenever (a, b) is an arc. This numbering proceeds as follows:

> Let $[a]'$ be a node of $\bar{f}$ without incoming arcs. Such a node exists as there are no circuits. Define $\lambda([a]') := 0$ and let $k := 1$.
>
> As long as there are nodes without a number:
>
> > There is a node $[b]'$ without a label such that the tails of all incoming arcs have already a label. Let $\lambda([b]') := k$ and let $k := k + 1$.

Note that $\lambda([s]') < \lambda([a]') < \lambda([t]')$ for every $a \notin [s]' \cup [t]'$. By (v), every component of $[a]'$ represents an element of V_X. Hence, every node can be reached from the source of $\bar{f}$ and gets a label.

Define $p(b) := (\lambda([b]') - \lambda([s]'))/(\lambda([t]') - \lambda([s]'))$ for every $b \in V_X$. Since every node of $\bar{f}$ has been given a different label, p is a potential function on V_X that separates the nodes of $V_{\bar{f}}$. Hence, the nodes of $\bar{f}$ correspond one-to-one with the nodes of g.

If there is a private X-used arc from a to b and $[a] \neq [b]$, there is, by definition, an arc $([a],[b]) \in E_g$ and also an arc $([a]',[b]')$ in $\bar{f}$, as such an arc is only contracted if it occurs in an circuit (step (iv)) (but then $[a]' = [b]'$), or when there is a parallel arc (step (vi)).

If there is a non-X-used path in f from an element of $[b]$ to an element of $[a]$, not visiting other vertices of V_X, this path has been reversed (step (iii)) and replaced by one arc (step (v)). Hence, there is an arc from $[a]'$ to $[b]'$ in $\bar{f}$. These are all reasons for the existence of arcs in g. Hence, $\bar{f}$ contains certainly the arcs of g but maybe more.

During the steps (i)–(iv) some arcs of f disappear because their endpoints are identified (in step (ii) if they connect an X-used public arc and in step (iv) if the arc occurs in a circuit). Furthermore, the orientation of some arcs has been reversed. What is left are X-used private arcs (and these define also arcs in g) and paths of non-X-used arcs, only visiting V_X in the endpoints. In step (v) each such path is replaced by an arc between the endpoints and also in g this defines an arc. □

In the example, the injective labeling can be given as in Fig. 5.

7. The computation of Nu(g)

The input of the algorithm is the network $g = \langle V_g, E_g, [s], [t]\rangle$, given in the previous section. The computation of the potential on g corresponding to the nucleolus will be by an improving-direction method. The starting point is found by a labeling only slightly different from the one used in the previous section. In fact it is an algorithm to find all longest paths from the source to the sink.

After having verified that the source and the sink do not coincide, label the nodes of g as follows:

For all $a \in V_g$: $\lambda(a) := 0$, let $k := 0$,
As long as $[t]$ is not the only node with label k:
  $k := k + 1$.
  For all edges e of which the tail has label $(k-1)$: $\lambda(\text{head}(e)) := k$.
Normalize the labeling by dividing by $\lambda([t[)$: $\lambda(a) := \lambda(a)/\lambda([t])$. $(a \in V_g)$

Figure 6 depicts the initial labeling of Example 10 (the bold edges form the longest path).

If n is the label of the sink before normalizing, the longest path from source to sink consists of n edges. Hence, at least one edge on this path has at potential difference of $1/n$ or less. The found potential *has* a minimal increase of $1/n$. So the maximal minimal increase has been found.

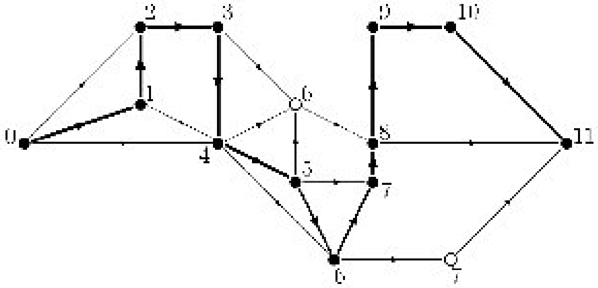

Fig. 6. The initial labeling (before normalizing).

Let F_0 be the set of nodes that are situated on a longest path. All labelings with a maximal potential difference of $1/n$ give the nodes in F_0 the value of the current labeling. Therefore, labels of nodes in F_0 will not be changed during the procedure and are called *fixed*. We can calculate F_0 as follows:

$F_0 := \{[t]\}$.
As long as $[s] \notin F_0$:
Let A be the set of nodes a for which there exists a node $b \in F_0$ such that $(a, b) \in E_g$ and $\lambda(b) - \lambda(a) = 1/n$.
Let $F_0 := F_0 \cup A$.

In Fig. 6, the elements of F_0 have been depicted boldly. Let the current *stepsize* be $\ell_0 := 1/n$.

It turns out[2] that the speed of the algorithm improves if at this stage all edges between fixed nodes are removed (they are no longer interesting):

Let $E_0 := E_g \setminus \{e \in E_g\colon$ both ends of e are situated in $F_0\}$.

Furthermore, again to improve the performance of the algorithm, it is useful to store the edges in a sequence $e_1, \ldots, e_m$ in such a way that if $i < j$, then $\lambda(\text{head}(e_i)) \leqslant \lambda(\text{head}(e_j))$. This has the advantage that every path (not only from source to sink) consists of a sequence of edges with increasing rank.

After having found the initial potential, a number of iterations will be performed to find successively the next maximal level of the smallest increase of the potential along edges that are still there. The input of iteration $i+1$ is the network $g_i := \langle V_g, F_i, E_i, [s], [t], \lambda\rangle$. If $E_i = \emptyset$, the algorithm terminates. The iteration finds the next stepsize ℓ_{i+1} and updates the labeling λ accordingly. Let $\mathcal{P}_i$ be the collection of paths with endpoints in F_i consisting of edges in E_i. Each path P in $\mathcal{P}_i$ gives a ratio of the potential increase along this path divided by the number of its edges. In fact, the next stepsize ℓ_{i+1} equals the minimum over these ratios.

$$\ell_{i+1} = \min\left\{\frac{\lambda(b) - \lambda(a)}{|P|} \;\middle|\; a, b \in F_i,\ P \in \mathcal{P}_i \text{ from } a \text{ to b}\right\}.$$

[2] We actually have written the algorithm in Matlab. For information, please send an email to J.H.Reijnierse@uvt.nl.

The iteration $i+1$ is given by:

Calculate ℓ_{i+1} (this subroutine will be given below).
Relabel as follows:
For $j = 1$ to $|E_i|$:
$\lambda(\text{head}(e_j)) := \max\{\lambda(\text{head}(e_j)), \lambda(\text{tail}(e_j)) + \ell_{i+1}\}$.
Update the set of fixed labeled vertices:
Let $F_{i+1} := F_i$.
For $j = |E_i|$ down to 1: (edges must be considered in decreasing order)
If $\text{head}(e_j) \in F_{i+1}$ and $\lambda(\text{head}(e_j)) - \lambda(\text{tail}(e_j)) = \ell_{i+1}$:
$F_{i+1} := F_{i+1} \cup \{\text{tail}(e_j)\}$.
Update the set of interesting edges:
Let $E_{i+1} := E_i \setminus \{e \in E_i : \text{both ends of } e \text{ are situated in } F_{i+1}\}$.

This ends iteration $i+1$. If $F_{i+1} \neq V_g$, let $i := i+1$ and perform the iteration again.

The subroutine calculating the stepsize ℓ_{i+1} is a generalization of the longest path algorithm of the initial step. Instead of starting at the source and ending at the sink, paths can start and end anywhere in F_i:

$\ell_{i+1} := 1$.   (or any other upper bound)
For all $a \in F_i$ with $\{e \in E_i : \text{tail}(e) = a\} \neq \emptyset$:
$\mu(a) := 0$.
For $j = 1$ to $|E_i|$:
If $\text{tail}(e_j)$ has a μ-label and $\text{head}(e_j)$ does not:
$\mu(\text{head}(e_j)) := \mu(\text{tail}(e_j)) + 1$.
If both ends have μ-labels:
$\mu(\text{head}(e_j)) := \max\{\mu(\text{head}(e_j)), \mu(\text{tail}(e_j)) + 1\}$.
If $\text{head}(e_j) \in F_i$:
$\ell_{i+1} := \min\{\ell_{i+1}, \dfrac{\lambda(\text{head}(e_j)) - \lambda(a)}{\mu(\text{head}(e_j)) - \mu(a)}\}$.
Remove the μ-labeling.

In the example, $\ell_1 = 3/22$. Figure 7 shows the new labeling. The path with minimal ratio has been depicted boldly and so are the elements of F_1.

Let us explain why an iteration finds a potential λ with the next maximal minimum level ℓ_{i+1}. Let $e \in E_i$.

If $\text{head}(e) \notin F_i$, then:

$$\lambda(\text{head}(e)) = \max\{\lambda(a) + \ell_{i+1} \mid (a, \text{head}(e)) \in E_i\} \geqslant \lambda(\text{tail}(e)) + \ell_{i+1}.$$

So, the potential difference along e is at least ℓ_{i+1}.

If $\text{head}(e) \in F_i$, then $\text{tail}(e)$ is not. We can find an edge ending at $\text{tail}(e)$ with potential ℓ_{i+1}, just take the (an) argument of $\max\{\lambda(a) + \ell_{i+1} \mid (a, \text{tail}(e)) \in E_i\}$. If the tail of this edge is still not in F_i, we can find another edge with potential ℓ_{i+1}, and again, and eventually find (backward) a path $P \in \mathcal{P}_i$ from some node a in F_i to $\text{head}(e)$, of which the

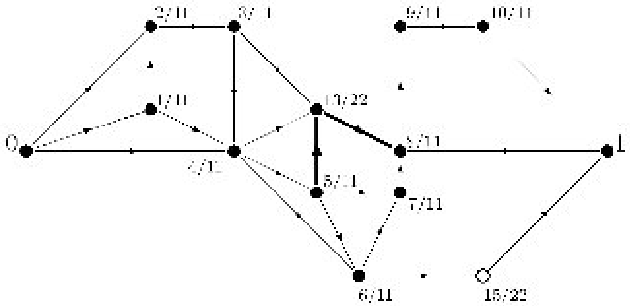

Fig. 7. The second labeling.

last edge is e. Furthermore, we know that all other edges along P have a potential ℓ_{i+1}. Because

$$\ell_{i+1} \leqslant \frac{\lambda(\mathrm{head}(e)) - \lambda(a)}{|P|} = \frac{[\lambda(\mathrm{head}(e)) - \lambda(\mathrm{tail}(e))] + (|P| - 1)\ell_{i+1}}{|P|},$$

we find that $\lambda(\mathrm{head}(e)) - \lambda(\mathrm{tail}(e))$ is at least ℓ_{i+1}.

Given that the potential values on elements of F_i are fixed, ℓ_{i+1} is an upper bound of the minimal potential increase along arcs in E_i, because of the path P in $\mathcal{P}_i$ with minimal ratio. The new potential *is* a potential of which the minimal increase along such arcs equals ℓ_{i+1}. So, the next maximal minimal potential difference has been found.

When the algorithm stops, because all vertices have a fixed potential, the final potential is the one with lexicographically maximal differences. The nucleolus $\mathrm{Nu}(g)$ has been computed. This proves the following theorem:

Theorem 12. *Let g be the modified digraph corresponding to a balanced simple flow network. Then the algorithm described above computes the element of* $\mathrm{Nu}(g)$.

In the example, all nodes but one are elements of F_1. We need one other iteration, in which the stepsize ℓ_2 equals $5/22$. The final labeling becomes is depicted in Fig. 8.

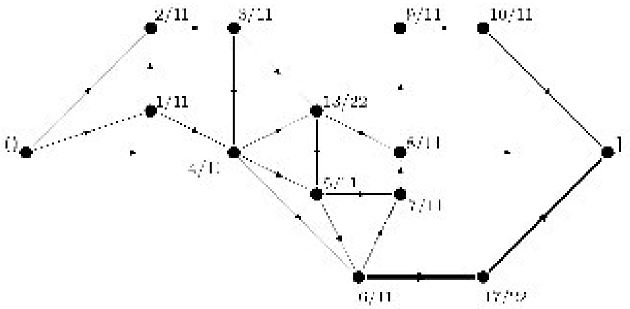

Fig. 8. The final labeling.

8. The complexity

The final section investigates the complexity of the algorithms in Sections 6 and 7. In Section 6, the construction of the modified graph g contains the Ford–Fulkerson algorithm, the algorithm to detect the strong components of a directed graph and the Floyd–Warshall algorithm to find the shortest path matrix.

The Ford–Fulkerson algorithm has a complexity of $\mathcal{O}(n\,|E|)$, as the search for a flow-increasing path can be done in $\mathcal{O}(|E|)$ steps and there are at most $v_f(N) \leqslant n$ iterations, when the core is nonempty. In fact, when a maximum flow X has been discovered, one can increase the capacity of the X-used public arcs from one to two and continue the search for another flow-increasing path. When this is possible, the core of the flow game is empty; when there is no flow-increasing path one finds a minimum cut consisting of private arcs and the core is nonempty.

The detection of the strong components of a directed graph has a complexity of $\mathcal{O}(|E|)$ by a well known depth first algorithm.

The Floyd–Warshall algorithm has a complexity of $\mathcal{O}(|V_g|^3) \leqslant \mathcal{O}(|V|^3)$. It has to be performed $|V_g|$ times.

The computation of the nucleolus in Section 7 has at most $|V_g|$ iterations, as in each iteration the value of the potential $\mathrm{Nu}(g)$ in at least one new point is discovered. The i-th iteration requires at most $\mathcal{O}(|E_i|\,|F_i|)$ steps. Because the graph g does not have *multiple connections*), we have $|E_g| \leqslant |V_g|^2$, so $\mathcal{O}(|E_i|\,|F_i|)$ can be estimated by:

$$\mathcal{O}\big(|E_i||F_i|\big) \leqslant \mathcal{O}\big(|E_g||V_g|\big) \leqslant \mathcal{O}\big(|V_g|^3\big) \leqslant \mathcal{O}\big(|V|^3\big).$$

Therefore, the entire computation of $\mathrm{Nu}(g)$ has a complexity of at most $\mathcal{O}(|V|^4)$. The total complexity is the maximum of $n\,|E|$ and $|V|^4$.

Before one can hope to obtain an estimation of the complexity in terms of $n = |N|$ only, one has to avoid the occurrence of superfluous arcs and nodes. This means that before one starts with any computation:

(i) one has to remove nodes that cannot be reached from the source or from which the sink cannot be reached,
(ii) for every arc e one has to remove parallel public arcs as long as the total number of parallel arcs is larger than n, i.e. $|E| \leqslant n\,|V|^2$,
(iii) one has to contract a public arc with its preceding arc when there is only one preceding arc or with its succeeding arc when there is only one succeeding arc. This means, paths of length 2 or more without intermediate exits should contain private arcs only.

In a preliminary $\mathcal{O}(|E|)$-algorithm one can transform a general directed graph into a graph satisfying (i), (ii) and (iii).

If there is not an overwhelming number of public arcs, e.g. $|E| = \mathcal{O}(n)$, then also $|V| = \mathcal{O}(n)$ and the algorithm has a complexity of $\mathcal{O}(n^3)$. This is in particular the case when there are no public arcs.

Let us finish this section by a final remark concerning a recent result of Fang et al. (2002). Their model differs from ours in the sense that capacities are not necessarily 1 and public arcs are not allowed. Under these assumptions the paper shows that determining whether an imputation x is a core element is an NP-hard problem. A byproduct of Proposition 8 is that in our model this test can be performed in polynomial time. It can be done as follows.

Suppose we have a simple flow network $(V, s, t, E, \alpha, N, \sigma)$ and an element x of $\mathbb{R}^N$.

- Determine a maximal cycle-free flow X and thereby $v_f(N)$.
- Check whether x is an imputation: does it hold that $x(N) = v_f(N)$ and $x \geqslant 0$?
- If so, decompose X into $v_f(N)$ edge-disjoint paths $P_1, \ldots, P_{v_f(N)}$. Define for each path separately and for each node v on such a path P_k a potential: $p_k(v) := x(S)$, in which S is the coalition corresponding to the part of P_k from the source to v.
- Check for all k, ℓ whether $p_k(v) = p_\ell(v)$ for all nodes v situated on both P_k and P_ℓ. Moreover, verify that $p_k(t) = 1$ for all k.
- If so, define a candidate-potential on V_X by: $p(v) := p_k(v)$ for an arbitrary path P_k on which v is situated. Remove all X-used edges. By now, in order to have a core element, it should be the case that all remaining paths have a decreasing potential. With the algorithm of Floyd–Warshall, one can determine the pairs of nodes (a, b) such that a path from a to b still exists.
- If this test is affirmative, the candidate-potential p is a potential indeed and the corresponding core element x_p equals the test-vector x.

The complexity of this test is of order $\mathcal{O}(\max\{|V|^3, n|E|\})$. If there are, similar to the model of Fang et al., no public arcs, then both V and E are bounded by $2n$, so we find a complexity of order $\mathcal{O}(n^3)$.

Acknowledgments

We thank the referee and associate editor for their helpful comments and suggestions.

References

Curiel, I., Derks, J., Tijs, S., 1989. On balanced games and flow games with committee control. OR Spektrum 11, 83–88.

Fang, Q., Zhu, S., Cai, M., Deng, X., 2002. On computational complexity of membership test in flow games and linear production games. Int. J. Game Theory 31 (1), 39–45.

Ford, L., Fulkerson, D., 1956. Maximal flow through a network. Can. J. Math. 8, 399–404.

Granot, D., Granot, F., 1992. On some network flow games. Math. Operations Res. 17, 792–841.

Hubermann, G., 1980. The Nucleolus and the Essential Coalitions. Analysis and Optimization of Systems. Springer, Berlin.

Kalai, E., Zemel, E., 1982. Totally balanced games and games of flow. Math. Operations Res. 7, 476–478.

Maschler, M., Potters, J., Tijs, S., 1992. The general nucleolus and the reduced game property. Int. J. Game Theory 21, 83–106.

Papadimitriou, C., Steiglitz, K., 1982. Combinatorial Optimization, Algorithms and Complexity. Prentice Hall International, Englewood Cliffs, NJ.

Reijnierse, J., Potters, J., 1998. The $\mathcal{B}$-nucleolus of TU-games. Games Econ. Behav. 24, 77–96.

Reijnierse, J., Maschler, M., Potters, J., Tijs, S., 1996. Simple flow games. Games Econ. Behav. 16, 238–260.

Weiss, M.A., 1999. Data Structures and Algorithms in C++. Addison–Wesley/Longman.