Finding Nucleolus of Flow Game*

(Extended Abstract)

Xiaotie Deng† Qizhi Fang‡ Xiaoxun Sun§

Abstract

We study the algorithmic issues of finding the nucleolus of a flow game. The flow game is a cooperative game defined on a network $D = (V, E; \omega)$. The player set is E and the value of a coalition $S \subseteq E$ is defined as the value of the maximum flow from source to sink in the subnetwork induced by S. We show that the nucleolus of the flow game defined on a simple network ($\omega(e) = 1$ for each $e \in E$) can be computed in polynomial time by a linear program duality approach, settling a twenty-three years old conjecture by Kalai and Zemel. In contrast, we prove that both computation and recognition of the nucleolus are $\mathcal{NP}$-hard for flow games with general capacity.

Keywords: Flow game, nucleolus, LP duality, efficient algorithm, $\mathcal{NP}$-hard.

1 Introduction.

Cooperative game theory considers how to distribute income $v(N)$ generated by a group N to its members. The nucleolus, defined by Schmeidler [26], is a solution that lexicographically maximizes the sorted vector of excess for all nontrivial subsets. More formally, let an imputation $x : N \to R_+$ represent the income distributed to the members in N with $x(N) = v(N)$; for each subset $\emptyset \neq S \subset N$, let $v(S)$ be the revenue generated by the subset S of members; the excess is defined by $e(S, x) = x(S) - v(S)$, where $x(S) = \sum_{i \in S} x_i$. The sorted vector of excess is $e(S_1, x), e(S_2, x), \cdots, e(S_m, x)$, where $m = 2^{|N|} - 2$, such that $e(S_1, x) \leq e(S_2, x) \leq \cdots \leq e(S_m, x)$. Note that, for different imputation x, the ordered subset $S_1, S_2, \cdots, S_m$ is in general different. The nucleolus is one imputation that maximizes this sorted vector lexicographically.

Surprisingly, such a complicatedly defined solution, according to Aumann and Maschler [1], was the foundation that dictated a particular schema for the estate division problem set by Rabbi Nathan that baffled Talmudic scholars for two millennia. The problem is one of three wives married to a man who promised them 100, 200, and 300 zuz respectively upon his death. The husband died leaving an estate worth less than 600 zuz. According to the Talmud recommendation, the wives will receive an equi-partition of the estate if it is worth 100; but a proportional partition of the promised amount if it is worth 300. Such an intricacy has been made clear only after the work of Aumann and Maschler, and the Talmud rule has since been credited as anticipation of the modern cooperative game theory. The nucleolus and related solution concepts have been applied to study insurance policies by Lemaire [21], to real estate by Raghavan and Solymosi [25], to study peer group by Brânzei, Solymosi, and Tijs [2], to bankruptcy by Aumann and Maschler [1] as well as Malkevitch [22].

With the linear programming approach, the nucleolus can simply be solved in a sequence of linear programs, each exponential in the group size. Clearly, it does not provide us with an efficient solution of the nucleolus in general. The first polynomial time algorithm for nucleolus in a special cost allocation game on trees was derived by Megiddo [23], in advocation of efficient algorithms for cooperative game solutions, following the concept of good algorithms by Edmonds [5]. Subsequently, efficient algorithms have been developed for computing the nucleolus, such as, for assignment game [27], standard tree game [14] and matching game [18]. On the negative side, $\mathcal{NP}$-hardness result was obtained for minimum cost spanning tree game [8]. In general, results computing the nucleolus in polynomial time depend on specific properties of the games [13, 9, 20].

For a related problem, that of the core, the linear programming approach has been very successful in understanding the computational complexity issues. The core is a solution that possesses a sorted excess vector with all components non-negative. It can be represented by one single linear program of exponential size. A central approach is to establish an equivalent polynomial size integer program with a polynomial time solution,

*Supported by NSFG of China (No.10371114), a CERG grant (CityU 1156/04E) of Hong Kong RGC, an SRG grant (7001838) of City University of Hong Kong.

†Department of Computer Science, City University of Hong Kong, Hong Kong, P. R. China

‡Department of Mathematics, Ocean University of China, Qingdao 266003, P. R. China

§Department of Mathematics, Ocean University of China, Qingdao 266003, P. R. China

such as for assignment game of Shapley and Shubik [28], linear production game of Owen [24], partition game of Faigle and Kern [7], packing/covering game of Deng, Ibaraki and Nagamochi [3] and facility location game of Goemans and Skutella [12]. Linear program duality has played an important role in the above results characterizing the core in cooperative game theory.

A very interesting flow game was introduced by Kalai and Zemel [16, 17], which arose from the profit distribution problem related to the maximum flow in a network, in which arcs are owned by different individuals. It was shown in [16, 17] that the flow game on a simple network (edge capacity being all equal) is totally balanced, and the allocations corresponding to minimum cuts in the network always belong to core. Linear program duality was crucial for the results. They further conjectured that their approach can lead to an efficient algorithm for the computation of the nucleolus.

We study the nucleolus of flow games from the algorithmic point of view. We show that computing the nucleolus can be done in polynomial time for the flow game on a simple network. The proof is deep and, at the same time, an elegant application of linear program duality approach in Kalai and Zemel's work [17], and hence settling their conjecture. On the other hand, we prove that both the computation and the recognition of the nucleolus are $\mathcal{NP}$-hard for flow games in general cases. The $\mathcal{NP}$-hardness proof for recognizing the nucleolus also resolves a conjecture of Faigle, Kern and Kuipers [8].

In Section 2, we introduce the basic theoretic concepts and the definition of the flow game. In Section 3, we discuss the essential coalitions and dummy arcs in the flow game, which play an important role in characterization of the nucleolus. Section 4 is dedicated to the polynomial time algorithm for computing the nucleolus of the flow game defined on a simple network. Finally, in Section 5, we prove that both the problems of computing and recognizing the nucleolus are $\mathcal{NP}$-hard for flow games with general capacity.

2 Preliminary and Definition.

2.1 Cooperative game theory. The cooperative (revenue) game $\Gamma = (N, v)$ consists of a player set $N = \{1, 2, \cdots, n\}$ and a value function $v : 2^N \to R$ with $v(\emptyset) = 0$. The allocation to individual player $i \in N$ is represented by x_i, and $x = (x_1, x_2, \cdots, x_n)$ satisfying $\sum_{i \in N} x_i = v(N)$ is an allocation vector. The allocation $x \in R^n$ is called an *imputation* if $x_i \geq v(\{i\})$ holds for all $i \in N$. The set of imputations of Γ is denoted by $X(\Gamma)$. Given an allocation $x \in R^n$, the excess of a coalition S at x is defined as the number

$$e(S, x) := x(S) - v(S).$$

The *core* of a game Γ, denoted by $C(\Gamma)$, is defined as the set of all allocations whose excesses are non-negative. That is, $C(\Gamma) = \{x \in R^n : x(N) = v(N) \text{ and } x(S) \geq v(S),\ \forall S \subseteq N\}$. We use the shorthand notation $x(S) = \sum_{i \in S} x_i$ $(S \subseteq N)$ throughout this paper.

The nucleolus of a game Γ is defined in the following way. Given an allocation $x \in R^n$ of Γ, let $\Theta(x)$ denote the $(2^n - 2)$-dimensional vector of all non-trivial excesses $e(S, x)$, $\emptyset \neq S \neq N$, arranged in non-decreasing order. Let $\succeq_l$ be the "lexicographically greater than" relationship between vectors of the same dimension. The *nucleolus* of Γ, $\eta(\Gamma)$, is then defined to be the (unique) allocation $x \in R^n$ that lexicographically maximizes $\Theta(x)$ over the set of imputations $X(\Gamma)$. It is obviously that $\eta(\Gamma)$ always exists (when $X(\Gamma) \neq \emptyset$), and it is a member of $C(\Gamma)$ when $C(\Gamma)$ is non-empty. Kopelowitz [19] proposed to compute the nucleolus by recursively solving the following sequential linear programs (set $\mathcal{J}_0 = \{\emptyset, N\}$ and $\varepsilon_0 = 0$):

$$\mathrm{LP}_k : \quad \begin{array}{lll} \max \varepsilon & & \\ & x(S) = v(S) + \varepsilon_r & \forall S \in \mathcal{J}_r \\ & & r = 0, 1, \cdots, k-1 \\ & x(S) \geq v(S) + \varepsilon & \forall S \in 2^N \setminus \bigcup_{r=0}^{k-1} \mathcal{J}_r \\ & x \in X(\Gamma) & \end{array}$$

The number ε_r is the optimum value of the r-th program (LP_r), and $\mathcal{J}_r = \{S \in 2^N : x(S) = v(S) + \varepsilon_r \text{ for every } x \in X_r\}$, where $X_r = \{x \in X(\Gamma) : (x, \varepsilon_r) \text{ is an optimal solution of } \mathrm{LP}_r\}$. This sequential linear programming process for computing $\eta(\Gamma)$ is denoted by $\mathrm{SLP}(\eta(\Gamma))$.

2.2 Flow games. Consider a directed network $D = (V, E; \omega)$, where V is the vertex set, E is the arc set and $\omega : E \to R^+$ is the arc capacity function. Let s and t be two distinct vertices of D which we denote the 'source' and the 'sink' of the network, respectively. We assume that each player controls one arc in the network. The associated flow game $\Gamma = (E, v)$ is defined as:

1) The player set is E;

2) For each subset $S \subseteq E$, $v(S)$ is the value of the maximum flow from s to t in the subnetwork induced by the corresponding arc set S.

A network $D = (V, E; \omega)$ is called *simple*, if $\omega(e) = 1$ for every $e \in E$. The flow games associated to simple networks also fall into the scope of packing/covering games introduced in [3]. The following result on characterization of the core of a flow game is due to Kalai and Zemel [17].

THEOREM 2.1. *Let $\Gamma = (E, v)$ be the flow game defined on a network $D = (V, E; \omega)$. Then the core $C(\Gamma)$ is always non-empty, and a core allocation can be found in polynomial time. In the case D is a simple network, $C(\Gamma)$ is exactly the convex hull of the characteristic vectors of the minimum cuts of D.*

On the other hand, it was proved that for flow games with general capacities, the problem of checking whether a given allocation belongs to the core is *co-NP*-complete [6].

3 Essential Coalition and Dummy Arc.

In this section we restrict our attention to simple networks. We assume that each $v \in V$ is contained in some path from the source s to the sink t.

Let $\Gamma = (N, v)$ be a cooperative game. A subset $S \subseteq N$ is called an *essential* coalition of Γ if $v(S) > \sum_{T \in \mathcal{T}} v(T)$ for every non-trivial partition $\mathcal{T}$ of S, where a partition of S is called trivial if the partition consists of the coalition S itself. Also, single member coalitions are also defined to be essential, since there are only trivial partitions of such coalitions. We denote by $\mathcal{E}$ the set of all essential coalitions of Γ. It was shown in Huberman [15] that for a game Γ:

(1) the core $C(\Gamma)$ can be determined only by essential coalitions;

(2) when $C(\Gamma) \neq \emptyset$, dropping the constraints associated with inessential coalitions will not change the result of SLP($\eta(\Gamma)$) for computing the nucleolus. That is, the nucleolus can be determined completely by essential coalitions.

Let us consider the essential coalitions of flow games. An (s, t)-path in D is a simple path (it visits each node at most once) from s to t. Let $\mathcal{P}$ be the set of (s, t)-paths in D, each regarded as a subset of arcs, i.e., a coalition of E. The following result follows directly from the definition of essential coalition.

PROPOSITION 3.1. *Let $\Gamma = (E, v)$ be the flow game defined on a simple network $D = (V, E)$. Then the set of essential coalitions of Γ consists of all singletons and all coalitions with respect to (s, t)-paths. That is, $\mathcal{E} = \bigcup_{e \in E}\{e\} \bigcup \mathcal{P}$.*

In the following, we introduce dummy arcs (players) of two types, which play an important role in the computation of the nucleolus, and discuss their descriptions and related algorithm. The proofs of the following results can be found in the full version.

DEFINITION 3.1. *Let $D = (V, E)$ be a simple network. An arc $e \in E$ is called a dummy arc if $v(E \backslash \{e\}) = v(E)$. Furthermore, we distinguish the dummy arcs into two types:*

(1) A dummy arc e is called Type I, if there exists a maximum flow f of D with $f(e) > 0$;

(2) A dummy arc e is called Type II, if it holds that $f(e) = 0$ for any maximum flow f of D.

For example, e_1, e_2, e_3, e_4 are all dummy arcs in the network D depicted in Figure 1, e_1, e_2 and e_3 are dummy arcs of Type I, and e_4 is a dummy arc of Type II.

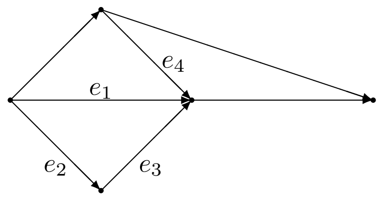

Figure 1: Network D

PROPOSITION 3.2. *An arc $e \in E$ is a dummy arc in the network D if and only if e is not contained in any minimum cut of D. Furthermore, $e \in E$ is a dummy arc if and only if $x(e) = 0$ for each core allocation $x \in C(\Gamma)$ of the associated flow game Γ.*

PROPOSITION 3.3. *Let P be an (s, t)-path in D. The following statements hold:*

(a) If P does not contain any dummy arc of Type II , then $|P \cap C| = 1$ for each minimum cut C in D;

(b) If P contains dummy arcs of Type II , then there exists a minimum cut C in D such that $|P \cap C| \geq 2$.

PROPOSITION 3.4. *The sets of dummy arcs of Type I and II can be identified efficiently.*

4 Nucleolus of Flow Game on Simple Network.

In this section we show that for the flow game defined on a simple network, the nucleolus can be computed efficiently. Let $D = (V, E)$ be a simple network with source s and sink t ($|E| = n$), and $\Gamma = (E, v)$ be the flow game defined on D. We also assume that each $v \in V$ is contained in some (s, t)-path in D. Throughout this section, we use the following notations:

$E_* = \{e \in E : e = (s, t)\}$;

E_0: the set of dummy arcs;

E_{01}: the set of dummy arcs of Type I;

E_{02}: the set of dummy arcs of Type II;

$\mathcal{P}$: the set of all (s,t)-paths in D;

$\mathcal{P}_0 = \{P \in \mathcal{P} : P$ contains dummy arc of Type II$\}$.

Recall that for the flow game Γ, the collection of essential coalitions is $\mathcal{E} = \bigcup_{e\in E}\{e\} \bigcup \mathcal{P}$. Followed from the results on essential coalitions in [15], the core of the flow game Γ is represented as

$$C(\Gamma) = \{x \in R^n : x(E) = v(E); \; x(P) \geq 1, \forall P \in \mathcal{P} \text{ and } x(e) \geq 0, \; \forall e \in E \setminus E_*\}, \tag{4.1}$$

and SLP($\eta(\Gamma)$) for computing the nucleolus can be described as LP$_k$ $(k = 1, 2, \cdots)$:

(4.2)

$$\begin{array}{lll} \max & \varepsilon & \\ & x(P) \geq 1 + \varepsilon & \forall P \in \mathcal{P} \setminus \bigcup_{r=0}^{k-1} \mathcal{P}^r \\ & x(e) \geq \varepsilon & \forall e \in (E \setminus E_*) \setminus \bigcup_{r=0}^{k-1} E^r \\ & x(e) = \varepsilon_r & \forall e \in E^r, \; r = 1, 2, \cdots, k-1 \\ & x(P) = 1 + \varepsilon_r & \forall P \in \mathcal{P}^r, \; r = 1, 2, \cdots, k-1 \\ & x(E) = v(E) & \end{array}$$

Here $\mathcal{P}^r = \{P \in \mathcal{P} : x(P) = 1 + \varepsilon_r$ for every $x \in X^r\}$ and $E^r = \{e \in E \setminus E_* : x(e) = \varepsilon_r$ for every $x \in X^r\}$, where the number ε_r is the optimum value of the r-th program (LP$_r$) and $X^r = \{x \in X(\Gamma) : (x, \varepsilon_r)$ is an optimal solution of LP$_r\}$, $X^0 = X(\Gamma)$.

In the following, we will discuss how to solve the SLP($\eta(\Gamma)$) (4.2) efficiently. (The details of the proofs in this section are referred to the full version of the paper.)

Proposition 4.1. *For LP$_1$ in SLP($\eta(\Gamma)$) (4.2), we have $X^1 = C(\Gamma)$, $\varepsilon_1 = 0$ and consequently, $\mathcal{P}^1 = \{P : P \in \mathcal{P} \setminus \mathcal{P}_0\}$ and $E^1 = \{e : e \in E_0\}$.*

Proposition 4.1 implies that if the network D does not contain any dummy arc of Type II, i.e., $\mathcal{P}_0 = \emptyset$, then SLP($\eta(\Gamma)$) (4.2) can be simplified as $(k = 1, 2, \cdots)$:

$$\begin{array}{lll} \max & \varepsilon & \\ & x(P) \geq 1 & \forall \; P \in \mathcal{P} \\ & x(e) \geq \varepsilon & \forall e \in (E \setminus E_*) \setminus \bigcup_{r=1}^{k-1} E^r \\ & x(e) = \varepsilon_r & \forall e \in E^r, \; r = 1, 2, \cdots, k-1 \\ & x(E) = v(E) & \end{array} \tag{4.3}$$

Note that in the sequential LPs (4.3), we use the constrains $x(P) \geq 1$ replacing $x(P) = 1$, $\forall P \in \mathcal{P}$. According to Proposition 3.3 (a), it is easy to see that this replacement does not change the solutions of these LPs. Based on the polynomial equivalence of optimization and separation problems (see, e.g., [11]) (here, $x(P) \geq 1$ can be checked by solving the shortest (s,t)-path w.r.t the arc length $l(e) = x(e)$ for $e \in E$), we can conclude that the sequential LPs (4.3), consequently the nucleolus $\eta(\Gamma)$, can be computed in polynomial time.

Proposition 4.2. *For the flow game Γ defined on a simple network with no dummy arc of Type II, the nucleolus $\eta(\Gamma)$ can be computed in polynomial time.*

When the network D contains dummy arcs of Type II, the number of (s,t)-paths in $\mathcal{P}_0$ may be exponential in n and identifying the (s,t)-paths in $\mathcal{P}_0$ which are binding at all optimal solutions of LP$_k$ in (4.2) can not be done in polynomial time directly. Therefore, we have to investigate a new approach to dealing with the SLP($\eta(\Gamma)$).

Let $W \subseteq V$, denote by $\delta^+(W)$ and $\delta^-(W)$ the sets of arcs leaving W and entering W, respectively. Define a function $c : E \to \{0, 1\}$ with $c(e) = 1$ if $e \in \delta^+(\{s\})$, and $c(e) = 0$, otherwise. Consider the following arc-flow formulation of the maximum flow problem in the network $D = (V, E)$:

(LP*)

$$\begin{array}{lll} \max & \sum_{e\in E} c(e)y(e) & \\ s.t. & \sum_{e\in\delta^+(\{v\})} y(e) - \sum_{e\in\delta^-(\{v\})} y(e) = 0 & \forall \; v \in V \setminus \{s,t\} \\ & 0 \leq y(e) \leq 1 & \forall \; e \in E \end{array}$$

The dual program of (LP*) is:

(DLP*)

$$\begin{array}{lll} \min & \sum_{e\in E} z(e) & \\ s.t. & z(e) + \phi(v) - \phi(w) \geq c(e) & \forall e = (v,w) \in E \\ & z(e) \geq 0 & \forall e \in E \end{array}$$

The following result is due to Kalai and Zemel [17].

Proposition 4.3. *Let $z \in C(\Gamma)$. Then there exists $\phi = \{\phi(v) : v \in V\}$ such that (z, ϕ) is an optimal solution to (DLP*).*

Kalai and Zemel [17] conjectured that this theorem can serve as a practical basis for computing the nucleolus of a flow game. In the rest of this section, we will show that Kalai and Zemel's result (Proposition 4.3) indeed plays an important role in our approach to efficiently computing the nucleolus.

Let $z \in C(\Gamma)$ and (z, ϕ) be an optimal solution for (DLP*). Proposition 4.3 leads to the following remarks:

Remark 1. By the definitions of the two types of dummy arc and LP duality theorem,

$$\begin{array}{ll}(4.4) & \begin{array}{l}\forall\ e=(v,w)\in E\setminus E_{02}:\\ \qquad z(e)+\phi(v)-\phi(w)=c(e),\\ \forall\ e=(v,w)\in E_{02}:\\ \qquad z(e)+\phi(v)-\phi(w)\geq c(e).\end{array}\end{array}$$

Also by Proposition 3.2 and the definition of c, we have

$$(4.5)\qquad \forall\ e\in E_{02},\quad z(e)=0\ \text{ and }\ c(e)=0.$$

Remark 2. Let $P=\{s,v_{i_1},\cdots,v_{i_k},t\}\in\mathcal{P}\setminus\mathcal{P}_0$. Then $c(P)=1$, and Proposition 3.3(a) implies that $z(P)=1$. Followed from the constraints of (DLP*) and formula (4.4),

$$\begin{aligned}z(P)=\ & c(P)+\phi(v_{i_1})-\phi(s)\\ & +\phi(v_{i_2})-\phi(v_{i_1})+\cdots+\phi(t)-\phi(v_{i_k})\\ =\ & 1+[\phi(t)-\phi(s)].\end{aligned}$$

It implies that

$$(4.6)\qquad \phi(t)=\phi(s).$$

Remark 3. Let $P\in\mathcal{P}_0$ be an (s,t)-path containing k dummy arcs of Type II, namely $e_1=(v_{1_1},v_{1_2}),\cdots,e_k=(v_{k_1},v_{k_2})$. Then by formulas (4.4), (4.5) and (4.6), we have

$$(4.7)\qquad \begin{aligned}z(P)-1 &= z(P)-c(P)\\ &= \textstyle\sum_{e\in P\setminus\{e_1,\cdots,e_k\}} z(e)-c(e)\\ &= \textstyle\sum_{j=1}^{k}(\phi(v_{j_2})-\phi(v_{j_1})).\end{aligned}$$

Formula (4.7) shows that, given a core allocation, the excess of an (s,t)-path $P\in\mathcal{P}_0$ is determined completely by the corresponding optimal solution of (DLP*).

The remarks given above, especially the formula (4.7), provide us a new idea to deal with SLP($\eta(\Gamma)$) of the flow game Γ. Let us define another sequential linear programs $\widetilde{\text{LP}}_k$ as follows ($k=2,3,\cdots$):

(4.8)

$$\begin{array}{lll}\max\ \varepsilon & & \\ x(e)+\phi(v)-\phi(w)\geq c(e) & \forall\ e=(v,w)\in E & \\ \phi(w)-\phi(v)=\tilde{\varepsilon}_r & \forall\ e=(v,w)\in E_{02}^r, & r=1,\cdots,k-1\\ x(e)=\tilde{\varepsilon}_r & \forall\ e\in E^r,\ r=1,\cdots,k-1 & \\ \phi(w)-\phi(v)\geq\varepsilon & \forall\ e=(v,w)\in E_{02}\setminus\bigcup_{r=1}^{k-1}E_{02}^r & \\ x(e)\geq\varepsilon & \forall\ e\in(E\setminus E_*)\setminus\bigcup_{r=1}^{k-1}E^r & \\ x(P)\geq 1 & \forall P\in\mathcal{P} & \\ x(e)\geq 0 & \forall e\in E & \\ x(E)=v(E) & & \end{array}$$

Here, $E_{02}^1=\emptyset$ and $E_{02}^r=\{e=(v,w)\in E_{02}:\phi(w)-\phi(v)=\tilde{\varepsilon}_r$ for every $x\in\widetilde{X}^r\}$, where the number $\tilde{\varepsilon}_r$ is the optimum value of the r-th linear program $\widetilde{\text{LP}}_r$ and $\widetilde{X}^r=\{x\in X(\Gamma):(x,\phi,\tilde{\varepsilon}_r)$ is an optimal solution of $\widetilde{\text{LP}}_r\}$. Notice that in (4.8), the last three constraints guarantee x must be in the core $C(\Gamma)$, and the other constraints guarantee (x,ϕ) is a solution of (DLP*). Thus, according to Proposition 4.3, each feasible solution (x,ϕ) of $\widetilde{\text{LP}}_k$ is in fact an optimal solution of (DLP*). We will show that the sequential linear programs $\widetilde{\text{LP}}_k$ is equivalent to SLP($\eta(\Gamma)$) given in (4.2).

To make our approach more clearly, we first assume that the network D satisfies the following property. And at last, we shall show how to transform D to a new network $\widehat{D}$ satisfying this property such that the corresponding flow games defined on D and $\widehat{D}$ have the same core and the same nucleolus.

Denote by $\mathcal{P}_{0*}$ be the set of (s,t)-paths containing only one dummy arc of Type II.

Property (A) *For each $e\in E_{02}$, there exists an (s,t)-path $P\in\mathcal{P}_{0*}$ containing e.*

PROPOSITION 4.4. *Let Γ be the flow game defined on $D=(V,E)$ with Property (A). In SLP($\eta(\Gamma)$) (4.2), if $P\in\mathcal{P}_0\setminus\mathcal{P}_{0*}$ satisfies that $e(P,\bar{x})=\varepsilon_k$ for some $\bar{x}\in X^{k-1}$, then there exists a path $P'\in\mathcal{P}_{0*}$ such that $e(P',\bar{x})=\varepsilon_k$. Furthermore, $P\in\mathcal{P}_0\setminus\mathcal{P}_{0*}$ need not be considered in any computation of SLP($\eta(\Gamma)$).*

Based on Proposition 4.4, we can rewrite SLP($\eta(\Gamma)$) described in (4.2) as following LPs, $\text{LP}_k(k=2,3,\cdots)$:

$$(4.9)\qquad \begin{array}{ll}\max\ \varepsilon & \\ x(P)=1+\varepsilon_r & \forall\ P\in\mathcal{P}_{0*}^r,\ r=1,\cdots,k-1\\ x(e)=\varepsilon_r & \forall\ e\in E^r,\ r=1,\cdots,k-1\\ x(P)\geq 1+\varepsilon & \forall\ P\in\mathcal{P}_{0*}\setminus\bigcup_{r=1}^{k-1}\mathcal{P}_{0*}^r\\ x(e)\geq\varepsilon & \forall\ e\in(E\setminus E_*)\setminus\bigcup_{r=1}^{k-1}E^r\\ x(P)\geq 1 & \forall P\in\mathcal{P}\\ x(e)\geq 0 & \forall e\in E\\ x(E)=v(E) & \end{array}$$

Here, $\mathcal{P}_{0*}^1=\emptyset$ and $\mathcal{P}_{0*}^r=\{P\in\mathcal{P}_{0*}:x(P)=1+\varepsilon_r$ for every $x\in X^r\}$, where the number ε_r is the optimum value of LP_r and $X^r=\{x\in X(\Gamma):(x,\varepsilon_r)$ is an optimal solution of $\text{LP}_r\}$.

PROPOSITION 4.5. *Let D be a simple network satisfying Property (A), and Γ be the associated flow game. Then the sequential linear programs $\widetilde{LP}_k$ (4.8) is equivalent to SLP($\eta(\Gamma)$) given in (4.9).*

Therefore, the nucleolus $\eta(\Gamma)$ can be obtained by proceeding to solve the sequential linear programs LP_1 and $\widetilde{\text{LP}}_k$ ($k=2,3,\cdots$). Followed from the similar discussion as in the proof of Proposition 4.2, it is shown that $\widetilde{\text{LP}}_k$ $(k = 2, 3, \cdots)$ can be solved efficiently. Therefore, we have

PROPOSITION 4.6. *Let D be a simple network satisfying Property (A), and Γ be the associated flow game. Then the nucleolus $\eta(\Gamma)$ can be calculated in polynomial time.*

The remainder problem is how to deal with a network D not possessing Property (A).

A vertex $v \in V$ is called *removable* if the arcs adjacent to it are all dummy arcs of Type II. Let R be the set of removable vertices in the network D. Suppose that the induced subnetwork $D[R]$ has l connected components. Each component corresponds to a subset of vertices, namely $R_1, R_2, \cdots, R_l$, which forms a partition of R.

We transform D to a new related network $\widehat{D}$ as follows: for each $i = 1, 2, \cdots, l$, we first delete the vertices in R_i and all the arcs adjacent to them; then add to D a set of new arcs $\{(u, w) : u \in \delta^-(R_i), w \in \delta^+(R_i)$ and there is a (u, w)-path in $D[R_i \cup \{u, w\}]\}$.

Note that the arcs deleted in D are all dummy arcs of Type II, and the new arcs added are also dummy arcs of Type II in $\widehat{D}$. It is easy to verify that

(a) $\widehat{D}$ satisfies Property (A);

(b) our construction of $\widehat{D}$ can be carried out in polynomial time;

(c) the flow games corresponding to D and $\widehat{D}$ have the same core and the same nucleolus (for edge $e \in D \setminus \widehat{D}$ and $e \in \widehat{D} \setminus D$, $x(e) = 0$ for each core allocation x of both games).

Thus, computing the nucleolus of the flow game defined on D can be transformed to the same problem defined on $\widehat{D}$. By summarizing the discuss and results given above, we therefore obtain our main result of this section.

THEOREM 4.1. *Let $D = (V, E)$ be a simple network and $\Gamma = (E, v)$ be the corresponding flow game. Then the nucleolus $\eta(\Gamma)$ can be computed in polynomial time.*

5 Computational Complexity Related to Nucleolus.

In this section, we show that for flow games with general capacity, both the computation and the recognition of the nucleolus are *NP*-hard. The technique we used is a polynomial transformation from the basic $\mathcal{NP}$-complete problem, EXACT COVER BY 3-SETS (X3C) [10].

Given a finite set $U = \{u_1, u_2, \cdots, u_{3q}\}$ and a collection $W = \{w_1, w_2, \cdots, w_{|W|}\}$ of 3-element subsets of U $(|W| \geq q)$. We say that an element $u \in U$ is covered by a subset $w \in W$ if $u \in w$. A sub-collection $C \subseteq W$ is called a cover of U if each element $u \in U$ is covered by some $w \in C$. A minimum cover of U is a cover with minimum cardinality. The problem X3C is to determine whether there exists a sub-collection $W' \subseteq W$ such that every element of U occurs in exactly one member of W', i.e., whether the minimum cover of U has the cardinality q. Throughout this section, we add a simple restriction to the instance of X3C:

(5.1) *Each element of U is included in* at least two subsets in W.

The problem of X3C with the restriction (5.1) is still $\mathcal{NP}$-complete.

Let $U' = \{u'_1, \cdots, u'_6\}$ and $W' = \{w'_1, w'_2, w'_3, w'_4\}$ be such that

$$\begin{aligned} w'_1 &= \{u'_1, u'_2, u'_3\}, \quad & w'_2 &= \{u'_3, u'_4, u'_5\}, \\ w'_3 &= \{u'_5, u'_6, u'_1\}, \quad & w'_4 &= \{u'_2, u'_4, u'_6\}. \end{aligned} \tag{5.2}$$

We call this couple (U', W') an extra module. The construction of w'_k $(k = 1, \cdots, 4)$ guarantees that in an extra module, the cardinality of a minimum cover of U' is 3.

Given an arbitrary instance of X3C (with the restriction (5.1)): $U = \{u_1, u_2, \cdots, u_{3q}\}$ and $W = \{w_1, w_2, \cdots, w_{|W|}\}$, we construct a network D in two steps.

Step 1. Construction of the expanded instance (EI).

We add three extra modules (U_1, W_1),(U_2, W_2) and (U_3, W_3) to the given instance of X3C, where U, U_1, U_2 and U_3 are disjoint, and obtain an expanded instance (EI):

$$(\text{EI}): \quad \begin{aligned} &\overline{U} = U \cup U_1 \cup U_2 \cup U_3, \\ &\overline{W} = W \cup W_1 \cup W_2 \cup W_3; \\ &\bar{q} := q + 6, \bar{f} := |W| + 12. \end{aligned}$$

Obviously, the expanded instance (EI) also satisfies the restriction (5.1).

Step 2. Construction of a network $D = (V, E; c)$.

Based on the expanded instance (EI), a network $D = (V, E; c)$ is constructed as follows. See Figure 2.

The vertex set V consists of three parts excluding the source s and the sink t:

- $\overline{U} = \{u_i : i = 1, 2, \cdots, 3\bar{q}\}$;
- $\overline{W} = \{w_j : j = 1, 2, \cdots, \bar{f}\}$;
- v_0 and v'_0 are two additional vertices.

Without confusion, we identify the elements in $\overline{U}$ and $\overline{W}$ with their corresponding vertices.

The arc set E and the associated capacity function $c : E \to R^+$ are defined as:

- $a_0 = (v'_0, t)$, $c(a_0) = 3\bar{q}$;
- $b_0 = (v_0, t)$, $c(b_0) = 3\bar{f} - 3\bar{q}$
- $A_1 = \{(s, w_i) : i = 1, \cdots, \bar{f}\}$, $\forall e \in A_1 : c(e) = 3$;
- $A_2 = \{(w_i, u_j) : u_j \in w_i, i = 1, \cdots, \bar{f}, j = 1, \cdots, 3\bar{q}\}$, $\forall e \in A_2 : c(e) = 1$;
- $A_3 = \{(u_j, v'_0) : j = 1, \cdots, 3\bar{q}\}$, $\forall e \in A_3 : c(e) = 1$;
- $B = \{(w_i, v_0) : i = 1, \cdots, \bar{f}\}$, $\forall e \in B : c(e) = 3$.

Denote by $\Gamma_D = (E, v)$ the flow game associated to the network D, and $n := |E|$. Obviously, $v(E) = 3\bar{f}$.

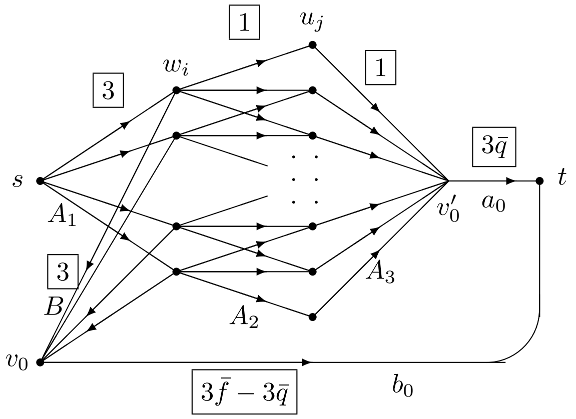

Figure 2: Network $D = (V, E; c)$.

Define an allocation $\eta^* \in R^n$ of Γ_D:

$$\eta^*(e) = \begin{cases} 3/4 & \forall e \in A_1 \\ 0 & \forall e \in A_2 \cup B \\ 1/2 & \forall e \in A_3 \\ \frac{3}{4}\bar{q} - \frac{3}{4} & e = a_0 \\ \frac{9}{4}(\bar{f} - \bar{q}) + \frac{3}{4} & e = b_0 \end{cases} \tag{5.3}$$

We show that η^* is not only a core allocation of Γ_D, but also an optimal solution of LP_1, LP_2 and LP_3 in $\mathrm{SLP}(\eta(\Gamma_D))$ for computing the nucleolus of Γ_D. Moreover, we have

Lemma 5.1. *The allocation η^* given in (5.3) belongs to $C(\Gamma)$. Furthermore, it is the nucleolus of the flow game Γ_D if and only if there is an exact cover in the instance of X3C.*

The proof of this lemma is referred to the full version of this paper. It can be seen that the construction of Γ_D can be carried out in polynomial time. Therefore, we obtain our main result of this section.

Theorem 5.1. *Computing the nucleolus of a flow game with general capacity is $\mathcal{NP}$-hard.*

Theorem 5.2. *Given a flow game $\Gamma = (E, v)$ and an allocation $y \in C(\Gamma)$, checking whether y is the nucleolus of Γ is $\mathcal{NP}$-hard.*

6 Conclusion and Remarks.

Computational complexity has recently proposed as a rationality measure for the solution concepts in cooperative game theory (Megiddo [23], Deng and Papadimitriou [4]). That is, a rational solution concept should not only be "fair" in theoretical sense but also efficiently computable. It has since been a major effort in the study of cooperative games.

The definition of the nucleolus of a cooperative game entails comparisons between vectors of exponential length. This makes it difficult to compute the nucleolus by directly following its definition. To obtain an efficient algorithm for the nucleolus of the flow game defined on a simple network, we provide an alternative characterization for the computation of the nucleolus based on the linear program duality to the arc-flow formulation of the maximum flow problem. This kind of formulation and its dual were used to characterize the core of the flow game by Kalai and Zemel [17]. They also conjectured that their result may serve as a practical basis for computing the nucleolus. Our work therefore answer in the affirmative to their conjecture. In general, linear program duality has proven itself a very powerful tool in the study of cooperative games, especially in the study of the cores. However, little work dealt with nucleolus by using the duality technique so far. Hence, our alternative formulation for computing the nucleolus may be of independent interest.

In the $\mathcal{NP}$-hardness proof given in Section 5, the flow game constructed possesses a polynomial size formulation of linear production game. Therefore, as a direct corollary, we come to the same $\mathcal{NP}$-hardness conclusion for linear production games. That is, both the problems of computing and recognizing the nucleolus of a linear production game are $\mathcal{NP}$-hard.

References

[1] R. J. Aumann and M. Maschler, *Game theoretic analysis of a bankruptcy problem from the Talmud*, Journal of Economic Theory, 36 (1985), pp. 195–396.

[2] Brânzei, R., Solymosi, T. and Tijs, S.H. *Strongly Essential Coalitions and the Nucleolus of Peer Group Games*, CentER Discussion Paper 2003-19.

[3] X. Deng, T. Ibaraki and H. Nagamochi, *Algorithmic aspects of the core of combinatorial optimization games*, Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, (1997), pp. 720–729.

[4] X. Deng and C. H. Papadimitriou , *On the complexity*

of cooperative solution concepts, Mathematics of Operations Research, 19 (1994), pp. 257–266.

[5] J. Edmonds, *Paths, Trees, and Flowers*, Canadian Journal of Mathematics, 17 (1965), pp. 449–467.

[6] Q. Fang, S. Zhu, M. Cai, X. Deng, *Membership for core of LP games and other games*, Lecture notes in computer science 2108, (2001), pp. 247–256.

[7] U. Faigle and W. Kern, *Partition games and the core of hierarchically convex cost games*, Universiteit Twente, faculteit der toegepaste wiskunde, Memorandum, No. 1269, 1995.

[8] U. Faigle, W. Kern and J. Kuipers, *Computing the nucleolus of min-cost spanning tree games is $\mathcal{NP}$-hard*, International Journal of Game Theory, 27 (1998), pp. 443–450.

[9] U. Faigle, W. Kern and J. Kuipers, *On the computation of the nucleolus of a cooperative game*, International Journal of Game Theory, 30 (2001), pp. 79–98.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness*, W. H. Freeman and Company, San Francisco, 1979.

[11] M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1993.

[12] M. X. Goemans and M. Skutella, *Cooperative facility location games*, Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, (2000), pp. 76–85.

[13] D. Granot, F. Granot and W. R. Zhu, *Characterization sets for the nucleolus*, International Journal of Game Theory, 27 (1998), pp. 359–374.

[14] D. Granot, M. Maschler, G. Owen and W. R. Zhu, *The kernel/nucleolus of a standard tree game*, International Journal of Game Theory, 25 (1996), pp. 219–244.

[15] G. Huberman, *The nucleolus and the essential coalitions*, In: Analysis and Optimizations of Systems, pp. 416–422, Springer, Berlin, 1980.

[16] E. Kalai and E. Zemel, *Totally balanced games and games of flow*, Mathematics of Operations Research, 7 (1982), pp. 476–478.

[17] E. Kalai and E. Zemel, *Generalized network problems yielding totally balanced games*, Operations Research, 30 (1982), pp. 498–1008.

[18] W. Kern and D. Paulusma, *Matching games: the least core and the nucleolus*, Mathematics of Operations Research, 28 (2003), pp. 294–308.

[19] A. Kopelowitz, *Computation of the kernels of simple games and the nucleolus of n-person games*, RM-31, Math. Dept., The Hebre University of Jerusalem, 1967.

[20] J. Kuipers T. Solymosi and H. Aarts , *Computing the nucleolus of some combinatorially structured games*, Mathematical Programming, 88 (2000), pp. 541–563.

[21] Jean Lemaire, Lemaire, Jean, *An Application of Game Theory: Cost Allocation*, ASTIN Bulletin 14, 1 (1984), pp. 61–81.

[22] Joe Malkevitch, *Resolving Bankruptcy Claims*, Feature Column, Monthly Essays on Mathematical Topics, AMS, March 2005. http://www.ams.org/featurecolumn/index.html.

[23] N. Megiddo, *Computational complexity and the game theory approach to cost allocation for ta tree*, Mathematics of Operations Research, 3 (1978), pp. 189–196.

[24] G. Owen, *On the core of linear production games*, Mathematical Programming, 9 (1975), pp. 358–370.

[25] T.E.S. Raghavan, T. Solymosi, *An Algorithm to Locate the Nucleolus Prices for a Real Estate Game*, GPI XII, 1998.

[26] D. Schmeidler, *The nucleolus of a characteristic function game*, SIAM Journal of Applied Mathematics, 17 (1969), pp. 1163–1170.

[27] T. Solymosi and T. E. S. Raghavan, *An algorithm for finding the nucleolus of assignment games*, International Journal of Game Theory, 23 (1994), pp. 119–143.

[28] L. S. Shapley and M. Shubik, *The assignment game*, International Journal of Game Theory, 1 (1972), pp. 111–130.