The Cost of Stability in Network Flow Games

Ezra Resnick[1], Yoram Bachrach[2], Reshef Meir[1], and Jeffrey S. Rosenschein[1]

[1] The Hebrew University of Jerusalem, Israel
{ezrar,reshef24,jeff}@cs.huji.ac.il
[2] Microsoft Research Ltd., Cambridge, UK
t-yobach@microsoft.com

Abstract. The core of a cooperative game contains all stable distributions of a coalition's gains among its members. However, some games have an empty core, with every distribution being unstable. We allow an external party to offer a supplemental payment to the grand coalition, which may stabilize the game, if the payment is sufficiently high. We consider the cost of stability (CoS)—the minimal payment that stabilizes the game.

We examine the CoS in threshold network flow games (TNFGs), where each agent controls an edge in a flow network, and a coalition wins if the maximal flow it can achieve exceeds a certain threshold. We show that in such games, it is coNP-complete to determine whether a given distribution (which includes an external payment) is stable. Nevertheless, we show how to bound and approximate the CoS in general TNFGs, and provide efficient algorithms for computing the CoS in several restricted cases.

1 Introduction

Many artificial intelligence settings involve multiple self-interested agents. Although self-interested, the agents may still benefit from cooperation. A natural tool for analyzing such strategic situations is, of course, cooperative game theory. In *cooperative games*, every subset (*coalition*) of agents can achieve a certain utility by cooperating. A natural question which arises is how to divide the gains obtained by a coalition among its members, since the *total* utility generated by the coalition is (by assumption) of little interest to each individual agent. Each possible division of the coalition's gains among its members is called an *imputation*.

Cooperative game theory solution concepts seek to define appropriate ways of distributing a coalition's gains among its members, so as to meet some desirable criteria. A prominent solution concept is the *core* [7], which is the set of all *stable* imputations—those where no subset of agents has a rational incentive to split off from the grand coalition (the set of all agents). Some games have infinitely many imputations in their core, while others have empty cores. In games where the core is empty, any imputation would be unstable. Thus, as opposed to normal-form games where the existence of a stable solution in the form of a (mixed-strategy) Nash equilibrium is guaranteed, some cooperative domains are inherently unstable.

We examine the possibility of stabilizing a cooperative game using external payments, based on a model introduced by Bachrach *et al* [1]. In this model, an external party is

R. Královič and D. Niwiński (Eds.): MFCS 2009, LNCS 5734, pp. 636–650, 2009.

interested in inducing all the agents to cooperate. This is done by offering the grand coalition a supplemental payment, given to the grand coalition as a whole, and provided only if this coalition is formed. The game's *cost of stability* (CoS) is the minimal external payment that allows a stable division of the grand coalition's adjusted gains.

In this work we consider games defined over network flow domains, where agents must cooperate to allow flow through the network. Such games can model situations where some commodity (traffic, liquid, information) flows through a network with various capacity constraints, and different entities own the different links (roads, pipes, cables) along the way. Such games have been studied in several works [8, 9, 2, 4]. We examine *threshold network flow games* (TNFGs), where each agent controls an edge in the network, and a coalition "wins" if the maximal flow it allows from the source vertex to the sink vertex exceeds a certain threshold. Computing the core of such a game enables finding a stable distribution of the rewards obtained from operating the network among the various agents. However, in many such games the core would be empty. In fact, we will see that unless there exists some *veto agent*, without which no coalition can achieve the required flow, the game's core would be empty and no imputation would be stable. There might exist some external party (e.g., a government) that would be willing to pay in order to ensure the cooperation of all agents in allowing flow through the network. Naturally, this external party would want to minimize its costs.

We explore the CoS in TNFGs. We show that it is coNP-complete to determine whether a given profit division, allowed by some external payment, makes a certain TNFG stable. Despite this hardness result, we nevertheless show how to bound and approximate the CoS in general TNFGs, and provide efficient algorithms for computing the CoS and finding optimal super-imputations in several restricted forms of TNFGs. We give an upper bound on the CoS in TNFGs based on the max-flow value of the network, which can also be used to approximate the CoS. We consider the CoS in *connectivity games*, a restricted form of TNFGs, and show that in these games the CoS is equal to the max-flow value of the network. We generalize this result, considering TNFGs with equal edge capacities. We also consider the case of *serial* TNFGs, built by serially connecting several component TNFGs. We show that the CoS of a serial TNFG is equal to the minimal CoS among the component TNFGs, and that this value may be computed efficiently if the number of edges in each component is not too large. Finally, we consider the relationship between the CoS in TNFGs and the CoS in another well-known cooperative domain—weighted voting games.

2 Preliminaries

We now define certain game-theoretic concepts necessary for our analysis of the cost of stability. We also define the domain on which we will be concentrating, the threshold network flow game.

2.1 Cooperative Games

A (transferable utility) *cooperative game* (also called a *coalitional game*) is defined by specifying the collective utility that can be achieved by every coalition of agents. In this work, the term *game* always refers to a cooperative game.

Definition 1. *A* cooperative game *consists of a finite set of agents N and a function $v : 2^N \to \mathbb{R}$. The function v is called the* characteristic function *of the game.*

The characteristic function maps every coalition of agents to the total utility that can be achieved by those agents together. In many typical cooperative games, adding more agents to a coalition never reduces the achievable utility. Such games are called *increasing*.

Definition 2. *A cooperative game $\langle N, v\rangle$ is* increasing *if $v(C') \leq v(C)$ for any $C' \subseteq C \subseteq N$.*

The TNFG domain considered in this work is one where a coalition can either win or lose. Such domains can be modeled as *simple* cooperative games.

Definition 3. *A cooperative game $\langle N, v\rangle$ is* simple *if v only takes the values 0 or 1, i.e., $v : 2^N \to \{0, 1\}$. We say a coalition $C \subseteq N$ is a* winning *coalition if $v(C) = 1$, and it is a* losing *coalition if $v(C) = 0$.*

In such games, it is usually assumed that $v(\emptyset) = 0$ and $v(N) = 1$. An agent without which no coalition can win is called a *veto agent*.

Definition 4. *In a simple cooperative game $\langle N, v\rangle$, an agent $a \in N$ is a* veto agent *if for any coalition $C \subseteq N$ it holds that $v(C \setminus \{a\}) = 0$.*

2.2 Flow Networks

Flow networks are useful for modeling systems where some fluid commodity travels through a network with capacity constraints. A flow network consists of a directed graph $\langle V, E\rangle$, with capacities on the edges $c : E \to \mathbb{R}_+$, a distinguished source vertex $s \in V$, and a distinguished sink vertex $t \in V$ ($s \neq t$). A *flow* through the network is a function $f : E \to \mathbb{R}_+$ which obeys the capacity constraints and conserves the flow at each vertex (except for the source and sink), meaning that the total flow entering a vertex must equal the total flow leaving that vertex. The *value* of a flow f (denoted $|f|$) is the net amount flowing out of the source (and into the sink). A *cut* of a flow network is a partition of the vertexes into two subsets S, T (where $S \cup T = V$ and $S \cap T = \emptyset$) such that $s \in S$ and $t \in T$. The capacity of a cut $\langle S, T\rangle$ is defined as the sum of the capacities of the edges crossing the cut (from S to T). We call a minimal capacity cut a *min-cut* and a maximal value flow a *max-flow*. The *max-flow min-cut theorem* states that in any flow network, the max-flow value is equal to the min-cut capacity.

Max-flow min-cut theorem. *The value of a flow f in a flow network is maximal if and only if there exists a cut of the network with capacity equal to $|f|$.*

Many efficient algorithms for finding a maximal value flow for a given network are known. Note that if all the edge capacities in a network are integers, the Ford-Fulkerson algorithm [6] produces an integer max-flow. This implies the following lemma:

Lemma 1. *In a flow network $\langle V, E, c, s, t\rangle$, if $c(e) \in \mathbb{N}$ for all $e \in E$ then there exists a max-flow f such that $f(e) \in \mathbb{N}$ for all $e \in E$.*

A general graph theory problem which may be solved efficiently using flow networks is that of finding the maximal number of edge-disjoint paths between two vertexes in a directed graph. This is done by assigning each edge a capacity of 1, and computing the max-flow value in the resulting flow network.

Lemma 2. *Given a flow network $\langle V, E, c, s, t\rangle$, if $c(e) = 1$ for all $e \in E$ then the maximal number of edge-disjoint paths from s to t in the directed graph $\langle V, E\rangle$ is equal to the max-flow value of the flow network.*

Proof. Since all capacities are 1, Lemma 1 implies that there exists a max-flow f such that $f(e) = 0$ or $f(e) = 1$ for all $e \in E$. This means that f must define $|f|$ edge-disjoint paths from s to t (with a flow of 1 through each). There cannot exist more than $|f|$ edge-disjoint paths, since then we could construct a flow whose value was greater than $|f|$, contradicting the assumption that f is a max-flow. □

2.3 Threshold Network Flow Games

A *threshold network flow game* (TNFG) is a cooperative game defined over a flow network, where each agent controls an edge in the network. Coalitions of agents may cooperate in order to send a certain flow from the source to the sink, and a coalition wins if the max-flow value allowed when using only the edges in the coalition exceeds a certain threshold (this threshold variant of network flow games has been studied by Kalai and Zemel [8] and Bachrach and Rosenschein [2]).

Definition 5. *A* threshold network flow domain *consists of a flow network $\langle V, E, c, s, t\rangle$ and a threshold $k \in \mathbb{R}_+$.*

Definition 6. *Given a threshold network flow domain $\langle V, E, c, s, t, k\rangle$, a* threshold network flow game (TNFG) *is the cooperative game $\langle N, v\rangle$ where $N = E$ and the characteristic function is defined as:*

$$v(C) = \begin{cases} 1 & \textit{if there exists a flow } f \textit{ in the network such that } |f| \geq k \textit{ and} \\ & \forall e \in E \setminus C : f(e) = 0 \\ 0 & \textit{otherwise} \end{cases}$$

By definition, TNFGs are simple games. They are also increasing games, since adding more edges to a coalition can only increase the value of the max-flow. It is easy to check whether a given coalition is a winning coalition by computing the max-flow value of the network which contains only the edges in the coalition and checking whether that value exceeds the threshold.

2.4 Imputations and the Core

The characteristic function of a cooperative game defines only the *total* gains a coalition achieves, but does not offer a way of distributing those gains among the agents in the coalition. Such a division is called an *imputation* (or a *payoff vector*).

Definition 7. *Given a cooperative game $\langle N, v\rangle$, an* imputation *is a vector $p \in \mathbb{R}^N_+$ such that $\sum_{a\in N} p_a = v(N)$. We call p_a the* payoff *of agent a, and denote the payoff of a coalition $C \subseteq N$ as $p(C) = \sum_{a\in C} p_a$.*

Cooperative game theory solution concepts offer ways of choosing an imputation, so as to satisfy some criteria. A basic criterion is *individual rationality*, which requires that $p_a \geq v(\{a\})$ for any agent $a \in N$—otherwise, some agent has an incentive to leave the coalition and work alone. A stronger criterion is that of *coalitional rationality*, based on the notions of *blocking* coalitions and *stable* imputations.

Definition 8. *In a cooperative game $\langle N, v\rangle$, a coalition $C \subseteq N$* blocks *an imputation p if $p(C) < v(C)$.*

Definition 9. *In a cooperative game $\langle N, v\rangle$, an imputation p is* stable *if it is not blocked by any coalition, i.e., for every coalition $C \subseteq N$, $p(C) \geq v(C)$.*

If the coalition C blocks the imputation p, the members of C could leave the grand coalition, derive the gains of $v(C)$, give each member $a \in C$ its previous gains p_a—and still some utility remains, so each agent could get more utility. If an unstable imputation is chosen, we cannot expect all agents to remain in the grand coalition. The *core* is the set of all stable imputations.

Definition 10. *The* core *of a cooperative game is the set of all imputations that are stable.*

Some games have infinitely many imputations in their core, while other games have empty cores. If we divide the gains of the grand coalition using an imputation in the core, then no subset of agents has an incentive to break off and work alone. However, if the core is empty, then any possible division of the grand coalition's gains is unstable: there will always be some coalition with an incentive to break away. In simple games, there is a well-known characterization of the core based on the game's veto agents: the core consists of all imputations which divide the grand coalition's gains only among the veto agents. Consequently, the core of a simple game (such as a TNFG) is nonempty if and only if there exists at least one veto agent. Note that we can compute the core of a TNFG in polynomial time, simply by finding all the veto agents (a given edge is a veto agent if and only if the coalition of all other edges is a losing coalition).

What should we do if we are faced with a game whose core is empty, but we still wish to ensure that no coalition has an incentive to leave the grand coalition? In the next section we suggest a solution, using external payments.

3 The Cost of Stability

We now consider the possibility of stabilizing a cooperative game using external payments, leading to the definition of the *cost of stability*, as introduced by Bachrach *et al* [1]. If a game is increasing, the maximal utility is achieved by the grand coalition. However, if the game's core is empty, it is impossible to distribute the gains of the grand coalition in a stable manner among the agents. This impedes the agents' cooperation, rendering the grand coalition unstable. Consider an external party that would like to induce all the agents to cooperate. One way to do this is by offering the grand coalition a *supplemental payment* if all agents cooperate. This external payment is offered to the grand coalition as a whole, and is provided only if this coalition is formed. The *adjusted game* is defined based on the original game and the supplemental payment.

Definition 11. *Given a cooperative game $G = \langle N, v \rangle$ and a* supplemental payment $\Delta \in \mathbb{R}_+$, *the* adjusted game *is the cooperative game $G(\Delta) = \langle N, v' \rangle$ where the characteristic function is defined as:*

$$v'(C) = \begin{cases} v(C) & \text{if } C \neq N \\ v(C) + \Delta & \text{if } C = N \end{cases}$$

We call $v'(N) = v(N) + \Delta$ the grand coalition's *adjusted gains*. We call a division of the adjusted gains in the adjusted game a *super-imputation*.

Definition 12. *Given an adjusted game $G(\Delta) = \langle N, v' \rangle$, a* super-imputation *is a vector $p \in \mathbb{R}_+^N$ such that $\sum_{a \in N} p_a = v'(N) = v(N) + \Delta$.*

We will sometimes talk about super-imputations without explicitly defining the adjusted game—in such a case the supplemental payment is implied by the sum of the super-imputation's payments.

Even if the core of the original game G was empty, the core of the adjusted game $G(\Delta)$ may not be empty—if the supplemental payment is high enough. Naturally, the external party would prefer to minimize the supplemental payment. The *cost of stability* (CoS) is defined as the *minimal* sum of payments such that a stable super-imputation exists in the adjusted game.

Definition 13. *The* cost of stability *of a cooperative game $G = \langle N, v \rangle$ is defined as follows:*

$$\mathrm{CoS}(G) = \min_{\Delta \in \mathbb{R}_+} \{v(N) + \Delta : \textit{the core of } G(\Delta) \textit{ is nonempty}\}$$

Note that for any simple game G, $\mathrm{CoS}(G) \geq 1$, and $\mathrm{CoS}(G) = 1$ if and only if the core of G is nonempty. For simple games, we can give additional lower and upper bounds on the CoS.

Theorem 1. *If there exist m pairwise-disjoint winning coalitions in a simple game $G = \langle N, v \rangle$, then $\mathrm{CoS}(G) \geq m$.*

Proof. Let $C_1, \dots, C_m$ be pairwise-disjoint winning coalitions in G. Let p be a super-imputation such that $p(N) < m$. This means there must exist a winning coalition C_i ($1 \leq i \leq m$) such that $p(C_i) < 1$ (otherwise we would get $p(N) \geq \sum_{j=1}^{m} p(C_j) \geq \sum_{j=1}^{m} 1 = m$). This means that C_i blocks p and p is unstable. Therefore, any stable super-imputation p' must satisfy $p'(N) \geq m$, so $\mathrm{CoS}(G) \geq m$. □

Theorem 2. *Let $G = \langle N, v \rangle$ be a simple game and let $S \subseteq N$ be a subset of agents. If every winning coalition C in G satisfies $C \cap S \neq \emptyset$, then $\mathrm{CoS}(G) \leq |S|$.*

Proof. We define a super-imputation p as follows:

$$\forall a \in N : p_a = \begin{cases} 1 & \text{if } a \in S \\ 0 & \text{otherwise} \end{cases}$$

Any winning coalition includes at least one agent from S, and so is paid at least 1. This means that p is stable, therefore: $\mathrm{CoS}(G) \leq p(N) = |S|$. □

4 Hardness of Determining Stability of Super-Imputations in TNFGs

This work focuses on the CoS in TNFGs. We first consider the problem of testing whether a given super-imputation, allowed by a certain supplemental payment, is actually stable in a TNFG. We show that this problem is in fact coNP-complete.

Definition 14. TNFG-SUPER-IMPUTATION-STABILITY (TNFG-SIS): *Given a TNFG $G = \langle V, E, c, s, t, k\rangle$, a supplemental payment Δ, and a super-imputation p in the adjusted game $G(\Delta)$, decide whether p is stable, i.e., whether there exists some blocking coalition for p in $G(\Delta)$.*

Theorem 3. TNFG-SIS *is* coNP*-complete.*

Proof. TNFG-SIS is in coNP, since we can easily verify instability: given a potentially blocking coalition, we can check whether it is a winning coalition and whether the sum of payments to the coalition members is less than 1, in polynomial time. We show that TNFG-SIS is coNP-hard by polynomially reducing SUBSET-SUM to the complement of TNFG-SIS. SUBSET-SUM is a well-known NP-complete problem, where we are given a set of positive integers $A = \{a_1, \ldots, a_n\}$ and a positive integer b, and are asked to determine whether there exists a subset $A' \subseteq A$ such that the sum of the elements in A' is exactly b. Given a SUBSET-SUM instance, we construct the following TNFG:

$V = \{s, t\} \cup \{v_1, \ldots, v_n\}$
$E = \{(s, v_i) : 1 \leq i \leq n\} \cup \{(v_i, t) : 1 \leq i \leq n\}$
$\forall\, 1 \leq i \leq n :\ c(s, v_i) = c(v_i, t) = a_i$
$k = b$

In other words, for each element a_i we add a path from s to t with capacity a_i, and we define the threshold to be the target sum b (see Figure 1). We now define a super-imputation p as follows:

$$\forall\, 1 \leq i \leq n :\ p_{(s,v_i)} = p_{(v_i,t)} = \frac{a_i}{2(b+1)}$$

We show that this super-imputation is unstable if and only if the given SUBSET-SUM instance is a "yes" instance.

First, assume p is unstable. This means there is some winning coalition C such that $p(C) < 1$. We can assume that if $(s, v_i) \in C$ for some $1 \leq i \leq n$ then also $(v_i, t) \in C$

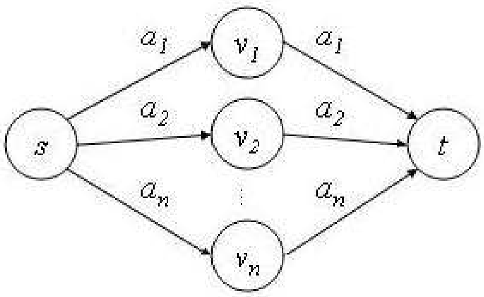

Fig. 1. Reduction of a SUBSET-SUM instance $\langle\{a_1, a_2, \ldots, a_n\}, b\rangle$ to an instance of TNFG-SIS. The game's threshold is b. We consider the stability of a super-imputation giving an edge with capacity a_i a payoff of $\frac{a_i}{2(b+1)}$.

(otherwise, we could remove (s, v_i) from C and C would still block p). Likewise, we can assume that if $(v_i, t) \in C$ for some $1 \leq i \leq n$ then also $(s, v_i) \in C$. Let $I \subseteq \{1, \ldots, n\}$ be the subset of indexes such that $C = \{(s, v_i) : i \in I\} \cup \{(v_i, t) : i \in I\}$. We assumed $p(C) < 1$, so:

$$1 > p(C) = 2\sum_{i \in I} \frac{a_i}{2(b+1)} \Rightarrow b + 1 > \sum_{i \in I} a_i$$

The max-flow value allowed by C is $\sum_{i \in I} a_i$, and we assumed C is a winning coalition, so $\sum_{i \in I} a_i \geq b$. Altogether, we get:

$$b + 1 > \sum_{i \in I} a_i \geq b$$

But since all a_i are integers, we conclude that $\sum_{i \in I} a_i = b$, so the given SUBSET-SUM instance is a "yes" instance.

On the other hand, assume the given SUBSET-SUM instance is a "yes" instance. This means there is some subset of indexes I such that $\sum_{i \in I} a_i = b$. Define the coalition $C = \{(s, v_i) : i \in I\} \cup \{(v_i, t) : i \in I\}$. The max-flow value allowed by C is $\sum_{i \in I} a_i = b$, so C is a winning coalition. However:

$$p(C) = 2\sum_{i \in I} \frac{a_i}{2(b+1)} = \frac{b}{b+1} < 1$$

So the coalition C blocks p, and p is unstable. □

Another interesting question is whether finding the CoS itself is computationally hard in TNFGs. The answer to that question is *yes*, although this result does not follow from Theorem 3. The proof is based on a reduction from the well-known PARTITION problem, and is omitted due to space constraints.

5 The Cost of Stability in TNFGs

We now show how to bound and approximate the CoS in general TNFGs, and provide efficient algorithms for computing the CoS and finding optimal super-imputations[1] in restricted classes of TNFGs.

5.1 Connectivity Games

We first show that the CoS can be computed efficiently in *connectivity games*, where a coalition wins if it contains a path from the network's source to its sink.

Definition 15. *A* connectivity game *is a TNFG where the capacities of all edges are* 1 *and the threshold is also* 1*.*

Theorem 4. *The CoS of a connectivity game is equal to the max-flow value of the underlying flow network.*

Proof. Let C be the set of edges crossing a min-cut in a connectivity game G. Notice that $|C|$ is equal to the max-flow value of the network due to the max-flow min-cut theorem (since all capacities are 1). Any winning coalition in G (containing a path from s to t) must include some edge in C, so from Theorem 2 we get $\mathrm{CoS}(G) \leq |C|$. On the other hand, Lemma 2 guarantees the existence of $|C|$ edge-disjoint paths from s to t, each of which is a winning coalition, so from Theorem 1 we get $\mathrm{CoS}(G) \geq |C|$. We conclude that $\mathrm{CoS}(G) = |C|$. □

5.2 Bounding the CoS in TNFGs

We now give an upper bound on the CoS in general TNFGs, based on the max-flow value of the underlying flow network.[2]

Theorem 5. *Let G be a TNFG with threshold k, and let F be the max-flow value of the underlying flow network. Then* $\mathrm{CoS}(G) \leq \frac{F}{k}$.

Proof. Let E be the edge set of G, and let S be the set of edges crossing a min-cut of G. We define the super-imputation p as follows:

$$\forall e \in E : p_e = \begin{cases} \frac{c(e)}{k} & \text{if } e \in S \\ 0 & \text{otherwise} \end{cases}$$

Notice that due to the max-flow min-cut theorem:

$$p(E) = \sum_{e \in S} \frac{c(e)}{k} = \frac{F}{k}$$

[1] A super-imputation is optimal if it is stable and the sum of payments is equal to the CoS.

[2] Note that there is a trivial upper bound on the CoS in any simple game—the CoS is never greater than the number of agents in the game (this is implied by Theorem 2).

Let C be a winning coalition in G. This means that:

$$\sum_{e \in C \cap S} c(e) \geq k$$

And so:

$$p(C) = \sum_{e \in C} p_e = \sum_{e \in C \cap S} \frac{c(e)}{k} \geq 1$$

So p is stable and $\mathrm{CoS}(G) \leq p(E) = \frac{F}{k}$. □

A corollary of Theorem 5 is that the ratio between the max-flow value (F) and the threshold (k) of a TNFG (which is easy to compute) can serve as an approximation for the game's CoS (an $\frac{F}{k}$-approximation). Of course, this approximation is tighter the smaller the ratio. Also, the proof of Theorem 5 shows us how to efficiently find a stable super-imputation with adjusted gains equal to this ratio.

5.3 Equal Capacity TNFGs

We now generalize Theorem 4, showing an efficient way to compute the CoS of a TNFG with equal edge capacities.

Theorem 6. *If G is a TNFG where the capacities of all edges are equal to b and the threshold is rb (for some $b \in \mathbb{R}_+$ and $r \in \mathbb{N}$), then $\mathrm{CoS}(G) = \frac{F}{rb}$, where F is the max-flow value of the underlying flow network.*

Proof. We know that $\mathrm{CoS}(G) \leq \frac{F}{rb}$ by Theorem 5, so it suffices to prove that $\mathrm{CoS}(G) \geq \frac{F}{rb}$.[3]

Denote $d = \frac{F}{b}$. Note that $d \in \mathbb{N}$, since a min-cut in G contains d edges (each with capacity b). We claim that there must exist d edge-disjoint paths from s to t in G. This follows from Lemma 2, because if we changed all the capacities in the network to 1, the max-flow value would be d (any min-cut in the original network is still a min-cut after the change).

Let $C_1, \ldots, C_d$ denote edge-disjoint paths from s to t in G. Let p be a stable super-imputation in G. Since the threshold is rb, any coalition containing r of the paths C_i ($1 \leq i \leq d$) is a winning coalition. In other words, for any subset of indexes $I \subseteq \{1, \ldots, d\}$ where $|I| = r$, it must hold that:

$$\sum_{i \in I} p(C_i) \geq 1$$

We can write $\binom{d}{r}$ such inequalities, and each $p(C_i)$ appears in an equal number of them, so summing all the inequalities yields:

$$\frac{r}{d}\binom{d}{r}\sum_{i=1}^{d} p(C_i) \geq \binom{d}{r} \Rightarrow \sum_{i=1}^{d} p(C_i) \geq \frac{F}{rb}$$

[3] The proof of Theorem 5 also provides an efficient method for finding an optimal super-imputation in this case.

Since this is true for any stable super-imputation p, we conclude that: $\mathrm{CoS}(G) \geq \frac{F}{rb}$. □

Note that Theorem 4 is actually a special case of Theorem 6, where $r = b = 1$.

5.4 Serial TNFGs

We now examine the special case of *serial* TNFGs, built by serially connecting a sequence of component TNFGs. Such games can model scenarios where the flow must pass through a series of bottlenecks. We show that in such a case, the CoS of the entire sequence is equal to the minimal CoS among the component TNFGs.

Definition 16. *Given a set of TNFGs $\{G_1, \ldots, G_n\}$ all with the same threshold k, a* serial TNFG *is the TNFG with threshold k over the flow network obtained by merging the sink of G_i with the source of G_{i+1} for every $1 \leq i < n$.*

Theorem 7. *If G is a serial TNFG composed of the TNFGs $\{G_1, \ldots, G_n\}$, then $\mathrm{CoS}(G) = \min_{1 \leq i \leq n} \mathrm{CoS}(G_i)$.*

Proof. We will prove the theorem for the case where $n = 2$, and the general case follows by induction. Assume w.l.o.g. that $\mathrm{CoS}(G_1) \leq \mathrm{CoS}(G_2)$. Denote by E_1 and E_2 the edge sets of G_1 and G_2 respectively, and denote by $E = E_1 \cup E_2$ the edge set of G. Let p' be an optimal super-imputation in G_1 (i.e., p' is stable and $p'(E_1) = \mathrm{CoS}(G_1)$). We define the super-imputation p in G as follows:

$$\forall e \in E : p_e = \begin{cases} p'_e & \text{if } e \in E_1 \\ 0 & \text{if } e \in E_2 \end{cases}$$

Notice that $p(E) = p'(E_1) = \mathrm{CoS}(G_1)$. We will show that p is optimal in G, which implies that $\mathrm{CoS}(G) = \mathrm{CoS}(G_1)$.

First, let $C \subseteq E$ be a winning coalition in G. C must contain a subset $C' \subseteq C \cap E_1$ which is a winning coalition in G_1. p' is stable in G_1, so $p(C) = p'(C') \geq 1$, meaning that p is stable in G.

On the other hand, let $\tilde{p}$ be a super-imputation in G such that $\tilde{p}(E) < p(E) = \mathrm{CoS}(G_1)$. Write $\tilde{p}(E_1) = \alpha\tilde{p}(E)$ and $\tilde{p}(E_2) = (1-\alpha)\tilde{p}(E)$ for some $0 \leq \alpha \leq 1$. Assume w.l.o.g. $\alpha > 0$. There must exist a winning coalition $C_1 \subseteq E_1$ in G_1 such that $\tilde{p}(C_1) < \alpha$, otherwise the super-imputation $\frac{1}{\alpha}\tilde{p}$ would be stable in G_1 with adjusted gains smaller than $\mathrm{CoS}(G_1)$, which would be a contradiction. Likewise, there must exist a winning coalition $C_2 \subseteq E_2$ in G_2 such that $\tilde{p}(C_2) \leq (1-\alpha)$.[4] The coalition $C_1 \cup C_2$ is then a winning coalition in G, but $\tilde{p}(C_1 \cup C_2) < \alpha + (1-\alpha) = 1$. We conclude that $\tilde{p}$ is unstable in G and so $p(E) = \mathrm{CoS}(G)$.

Altogether, this shows that p is optimal in G, which implies that $\mathrm{CoS}(G) = p(E) = \mathrm{CoS}(G_1)$. So the theorem is proved for the case where $n = 2$. The general case follows by induction. □

Using Theorem 7, we now show how the CoS of a serial TNFG can be computed efficiently, as long as the number of edges in each component TNFG is not too large.

[4] Here the inequality is not strict, since if $\alpha = 1$ then $\tilde{p}$ is 0 for any coalition in G_2.

Definition 17. *A* B-bounded serial TNFG *is a serial TNFG with components* $\{G_1, \ldots, G_n\}$ *where the number of edges in each component TNFG* G_i $(1 \leq i \leq n)$ *is bounded by some constant number* B.[5]

Theorem 8. *The CoS of a B-bounded serial TNFG can be computed in polynomial time.*

Proof. Let G be a B-bounded serial TNFG whose component TNFGs are $\{G_1, \ldots, G_n\}$. We present an algorithm for computing $\mathrm{CoS}(G)$ in time linear in n, although the runtime includes a constant factor which is exponential in B. Therefore, this algorithm is only tractable if the bound B is small.

For each TNFG G_i, we can describe $\mathrm{CoS}(G_i)$ as a linear program. Let E_i denote the set of edges in G_i. For every $e \in E_i$ we define a variable p_e. The linear program is:

Minimize: $\sum\limits_{e \in E_i} p_e$

Under the constraints:
$\forall e \in E_i : p_e \geq 0$
$\forall C \subseteq E_i : \sum\limits_{e \in C} p_e \geq v(C)$

Recall that $v(C)$ equals 1 if C is a winning coalition and 0 otherwise. The number of constraints in the linear program is exponential in $|E_i|$, but $|E_i|$ is bounded by the constant B. Linear programs can be solved efficiently, so we can calculate $\mathrm{CoS}(G_i)$ in constant time with respect to n (although exponential with respect to B).

Once we have computed $\mathrm{CoS}(G_i)$ for all n component TNFGs, we can get $\mathrm{CoS}(G)$ by using Theorem 7: $\mathrm{CoS}(G) = \min\limits_{1 \leq i \leq n} \mathrm{CoS}(G_i)$. □

6 Weighted Voting Games and TNFGs

We now examine the relationship between the CoS in TNFGs and in *weighted voting games* (WVGs), a well-known game theoretic model of cooperative decision making.[6]

Definition 18. *Given a set of agents N, a weight function $w : N \rightarrow \mathbb{R}_+$ and a threshold $q \in \mathbb{R}_+$, a* weighted voting game *is the simple cooperative game where a coalition $C \subseteq N$ is a winning coalition if and only if the sum of the weights of the agents in C exceeds the threshold q, that is $w(C) = \sum_{a \in C} w(a) \geq q$.*

We can define a WVG based on any subset of agents in a TNFG: given a TNFG $\langle V, E, c, s, t, k \rangle$ and a subset of agents $F \subseteq E$, we define the WVG $W_F = \langle F, w, k \rangle$ where $w(e) = c(e)$ for every agent $e \in F$. We also denote the CoS of the new game W_F as $\mathrm{CoS}(F)$.

We now show that the CoS of a TNFG is bounded by the CoS of any WVG induced by the set of edges crossing a cut of the flow network.

[5] Note that the number of components n is not bounded.

[6] Analysis of the CoS in WVGs is given by Bachrach *et al* [1].

Theorem 9. *Let $G = \langle V, E, c, s, t, k\rangle$ be a TNFG instance, let $F \subseteq E$ be the set of edges crossing a cut of G, and let p_F be a super-imputation in the WVG W_F. If p_F is stable in W_F, then the super-imputation p is stable in G, where $p(e) = p_F(e)$ if $e \in F$ and $p(e) = 0$ otherwise.*

As a direct corollary we get that $\mathrm{CoS}(G) \leq \mathrm{CoS}(F)$.

Proof. Let $C \subseteq E$ be a winning coalition in G, i.e., the agents in C allow a flow with value k from s to t. In particular, it must hold that $w(F \cap C) = c(F \cap C) \geq k$, so $F \cap C$ is a winning coalition in W_F. Since p_F is stable in W_F, we know that $p_F(F \cap C) \geq 1$, and therefore:

$$p(C) \geq p(F \cap C) = p_F(F \cap C) \geq 1$$

So p is stable in G. □

We can now supply alternative proofs to some of the theorems in this work using WVGs. Theorem 5 is a direct corollary of Theorem 9, if we consider the edges crossing a min-cut of a TNFG. The hardness of TNFG-SIS (Theorem 3) follows from the hardness of the equivalent problem for WVGs, since we can reduce a WVG to a TNFG: given a WVG $W = \langle N, w, q\rangle$, we define the TNFG $G_W = \langle V, E, c, s, t, k\rangle$ by setting $V = \{s, t\}$; $c = w$; $k = q$ and $E = N$, where all edges are from s to t.[7] By similar arguments to those in the proof of Theorem 9, a super-imputation is stable in W if and only if it is stable in G_W. Bachrach *et al.* [1] prove that testing for super-imputation stability in WVGs is coNP-hard, so it follows that TNFG-SIS is coNP-hard as well.

7 Related Work

The concept of the core was introduced by Gillies [7]. Similar concepts are the least-core and the nucleolus [11], which are guaranteed to be nonempty. A different solution concept is the Shapley value [12], which aims for fairness rather than stability.

Elkind *et al.* [3] discuss various solution concepts in WVGs, showing that in this domain computing the core can be done in polynomial time, while many questions relating to other solution concepts are NP-hard. Elkind and Pasechnik [5] show a pseudo-polynomial algorithm for computing the nucleolus of WVGs.

Bachrach and Rosenschein [2] examine calculating power indexes in TNFGs. Power indexes attempt to measure how much "real power" each player has in a given game. It is shown that for TNFGs, computing the Shapley-Shubik index is NP-hard and computing the Banzhaf index is #P-complete. However, an efficient algorithm for the restricted case of connectivity games over bounded layer graphs is provided. Elkind *et al.* [4] show how to compute power indexes in the special case of series-parallel TNFGs.

While our work focuses on TNFGs, much research has considered the *cardinal* network flow game (CNFG), where a coalition's utility equals the max-flow value it can achieve. Computing the core in CNFGs can be done in polynomial time; Kalai and Zemel [8, 9] show that numerous families of CNFGs have nonempty cores.

[7] This requires allowing a multigraph, but we could avoid that by splitting every edge into two equivalent edges.

Yokoo *et al.* [13] demonstrate that various cooperative solution concepts (such as the core, nucleolus and Shapley value) are vulnerable to manipulations in open anonymous environments. They use a more fine-grained model of cooperative games, where each agent has a set of skills, and values are defined for different subsets of skills (rather than subsets of agents). They show that agents may sometimes profit from manipulations such as submitting false names, collusion, and hiding skills.

Monderer and Tennenholtz [10] investigate the case of an interested party who wishes to influence the behavior of agents in a game which is not under its control. The approach taken is close to the one we take here in spirit, although that work deals with *normal-form* games, not cooperative games. In that model, the interested party may commit to making non-negative payments to the agents if certain strategy profiles are selected. Payments are given to agents individually, but they are dependent on the strategies selected by all agents. As in our work, it is assumed that the interested party wishes to minimize its expenses. Determining the optimal monetary offers to be made in order to implement a desired outcome is shown to be NP-hard in general, but becomes tractable under certain modifications.

The CoS concept that we use here was first defined by Bachrach *et al.* [1], who examined the CoS in WVGs. It was shown that it is coNP-complete to test whether a given super-imputation in such a game is stable, but the CoS may be computed efficiently if either the player weights or payments are bounded. An efficient approximation algorithm for the CoS in general WVGs was also given.

8 Conclusion

We examined stabilizing cooperative games using external payments, and considered the CoS—the minimal total payment that allows a stable division of the grand coalition's gains among the agents, in the context of network flow games (TNFGs). We showed that it is coNP-complete to determine whether a given super-imputation in a TNFG is stable. We provided an upper bound on the CoS based on the network's max-flow, which can be used to approximate the CoS. We showed that in connectivity games and in equal capacity TNFGs , both the CoS and an optimal super-imputation may be found efficiently. We also showed how to compute the CoS in serial TNFGs with a small number of edges per component. Finally, we showed that the CoS of any TNFG can be bounded by the CoS of a WVG induced by some cut of the flow network.

In future work, we could examine the CoS in various other cooperative games. Additionally, it might be interesting to define the CoS for any coalition (not only for the grand coalition), and perhaps for various coalitional structures. Finally, we could investigate the relationship between the CoS and other cooperative solution concepts such as the least-core, nucleolus, and Shapley value.

References

[1] Bachrach, Y., Meir, R., Zuckerman, M., Rothe, J., Rosenschein, J.S.: The cost of stability in weighted voting games (extended abstract). In: The International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), Budapest, Hungary (May 2009)

[2] Bachrach, Y., Rosenschein, J.S.: Power in threshold network flow games. Journal of Autonomous Agents and Multi-Agent Systems 18(1), 106–132 (2009)

[3] Elkind, E., Goldberg, L.A., Goldberg, P.W., Wooldridge, M.: Computational complexity of weighted threshold games. In: The National Conference on Artificial Intelligence (AAAI), pp. 718–723. AAAI Press, Menlo Park (2007)

[4] Elkind, E., Goldberg, L.A., Goldberg, P.W., Wooldridge, M.: A tractable and expressive class of marginal contribution nets and its applications. In: The International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 1007–1014 (2008)

[5] Elkind, E., Pasechnik, D.V.: Computing the nucleolus of weighted voting games. In: The ACM-SIAM Symposium on Discrete Algorithms, SODA (2009)

[6] Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canadian Journal of Mathematics 8, 399–404 (1956)

[7] Gillies, D.B.: Some theorems on n-person games. PhD thesis. Princeton University, Princeton (1953)

[8] Kalai, E., Zemel, E.: On totally balanced games and games of flow. Discussion Papers 413, Northwestern University (January 1980)

[9] Kalai, E., Zemel, E.: Generalized network problems yielding totally balanced games. Operations Research 30, 998–1008 (1982)

[10] Monderer, D., Tennenholtz, M.: K-implementation. In: EC 2003: Proceedings of the 4th ACM conference on electronic commerce, pp. 19–28. ACM, New York (2003)

[11] Schmeidler, D.: The nucleolus of a characteristic function game. SIAM Journal on Applied Mathematics 17(6), 1163–1170 (1969)

[12] Shapley, L.S.: A value for n-person games. Contributions to the Theory of Games, 31–40 (1953)

[13] Yokoo, M., Conitzer, V., Sandholm, T., Ohta, N., Iwasaki, A.: Coalitional games in open anonymous environments. In: The National Conference on Artificial Intelligence (AAAI), pp. 509–515 (2005)