The Cost of Stability and Its Application to Weighted Voting Games

Yoram Bachrach[1], Edith Elkind[2], Reshef Meir[3], Dmitrii Pasechnik[2], Michael Zuckerman[3], Jörg Rothe[4], and Jeffrey S. Rosenschein[3]

[1] Microsoft Research, Cambridge, United Kingdom
[2] Nanyang Technological University, Singapore
[3] School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel
[4] Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Germany

Abstract. A key question in cooperative game theory is that of coalitional stability, usually captured by the notion of the *core*—the set of outcomes such that no subgroup of players has an incentive to deviate. However, some coalitional games have empty cores, and any outcome in such a game is unstable.
In this paper, we investigate the possibility of stabilizing a coalitional game by using external payments. We consider a scenario where an external party, which is interested in having the players work together, offers a supplemental payment to the grand coalition (or, more generally, a particular coalition structure). This payment is conditional on players not deviating from their coalition(s). The sum of this payment plus the actual gains of the coalition(s) may then be divided among the agents so as to promote stability. We define the *cost of stability (CoS)* as the minimal external payment that stabilizes the game.
We provide general bounds on the cost of stability in several classes of games, and explore its algorithmic properties. To develop a better intuition for the concepts we introduce, we provide a detailed algorithmic study of the cost of stability in weighted voting games, a simple but expressive class of games which can model decision-making in political bodies, and cooperation in multiagent settings. Finally, we extend our model and results to games with coalition structures.

1 Introduction

In recent years, algorithmic game theory, an emerging field that combines computer science, game theory and social choice, has received a lot of attention from the multiagent community [15, 6, 18, 16]. The reason for this interest is that multiagent systems research focuses on designing intelligent agents, i.e., entities that can coordinate, cooperate and negotiate without requiring human intervention. In many application domains, such agents are *selfish*, i.e., they are built to maximize the rewards obtained by their creators and can therefore be modeled by means of game theory. Moreover, as agents often have to function in rapidly changing environments, computational considerations are of great concern to their designers as well.

In many settings, such as online auctions and other types of markets, agents act individually. In this case, the standard notions of noncooperative game theory, such as *Nash equilibrium* or *dominant-strategy equilibrium*, provide a credible prediction of the outcome of the interaction. However, another frequently occurring type of scenario is that agents need to form teams to achieve their individual goals. In such domains, the focus turns from the interaction between single agents to the capabilities of subsets, or *coalitions*, of the agents. Thus, a more appropriate modeling toolkit for this setting would be that of *cooperative*, or *coalitional*, game theory [2], which studies what coalitions are most likely to arise and how their members distribute the gains from cooperation. When the agents are selfish, the latter question is obviously of great importance. Indeed, the *total* utility generated by the coalition is of little interest to individual agents; rather, each agent aims to maximize her own utility. Thus, a *stable* coalition can be formed only if the gains from cooperation can be distributed in a way that satisfies all agents.

The most prominent solution concept that aims to formalize the idea of stability in coalitional games is the *core*. Informally, an *outcome* of a coalitional game is a *payoff vector* which for each agent lists her share of the profit of the *grand coalition*, i.e., the coalition that includes all agents. An outcome is said to be in the core if it distributes the gains so that no subset of agents has an incentive to abandon the grand coalition and form a coalition of their own. It can be argued that the concept of the core captures the intuitive notion of stability in cooperative settings. However, it has an important drawback: the core of a game may be empty. In games with empty cores, any outcome is unstable, and therefore there is always a group of agents that is tempted to abandon the existing plan. This observation has triggered the invention of less demanding solution concepts, such as ε-core and the least core, as well as an interest in noncooperative approaches to identifying stable outcomes in coalitional games [3, 14]

In this paper, we approach this issue from a different perspective. Specifically, we examine the possibility of stabilizing the outcome of a game using external payments. Under this model, an external party, which can be seen as a central authority interested in stable functioning of the system, attempts to incentivize a coalition of agents to cooperate in a stable manner. This party does this by offering the members of a coalition a supplemental payment if they cooperate. This external payment is given to the coalition as a whole, and is provided only if this coalition is formed.

Clearly, when the supplemental payment is large enough, the resulting outcome is stable: the profit that the deviators can make on their own is dwarfed by the subsidy they could receive by sticking to the prescribed solution. However, normally the external party would want to minimize its expenditure. Thus, in this paper we define and study the *cost of stability*, which is the minimal supplemental payment that is required to ensure stability in a coalitional game. We start by considering this concept in the context where the central authority aims to ensure that *all* agents cooperate, i.e., it offers a supplemental payment in order to stabilize the grand coalition. We then extend our analysis to the setting where the goal of the center is the stability of a *coalition structure*, i.e., a partition of all agents into disjoint coalitions. In this setting, the center does not expect the agents to work as a single team, but nevertheless wants each individual team to be immune to deviations. Finally, we consider the scenario where the center is concerned with the stability of a particular coalition within a coalition structure. This model is appropriate when the central authority wants a particular group of agents to work together, but is indifferent to other agents switching coalitions.

We first provide bounds on the cost of stability in general coalitional games. We then show that for some interesting special cases, such as super-additive games, these bounds can be improved considerably. We also propose a general algorithmic technique for computing the cost of stability. Then, to develop a better understanding of the concepts proposed in the paper, we apply them in the context of *weighted voting games* (WVGs), a simple but powerful class of games that have been used to model cooperation in settings as diverse as, on the one hand, decision-making in political bodies such as the United Nations Security Council and the International Monetary Fund and, on the other hand, resource allocation in multiagent systems. For such games, we are able to obtain a complete characterization of the cost of stability from an algorithmic perspective.

The paper is organized as follows. In Section 2, we provide the necessary background on coalitional games. In Section 3, we formally define the cost of stability for the setting where the desired outcome is the grand coalition, prove bounds on the cost of stability and outline a general technique for computing it. We then focus on the computational aspects of the cost of stability in the context of our selected domain, i.e., weighted voting games. In Section 4.1, we demonstrate that computing the cost of stability in such games is coNP-hard if the weights are given in binary. On the other hand, for unary weights, we provide an efficient algorithm for this problem. We also investigate if the cost of stability can be efficiently approximated. In Section 4.2, we answer this question positively by describing a fully polynomial-time approximation scheme (FPTAS) for our problem. We complement this result by showing that, by distributing the payments in a very natural manner, we get within a factor of 2 of the optimal adjusted gains, i.e., the sum of the value of the grand coalition and the external payments. While this method of allocating payoffs does not necessarily minimize the center's expenditure, the fact that it is both easy to implement and has a bounded worst-case performance may make it an attractive proposition in certain settings. In Section 5, we extend our discussion to the setting where the center aims to stabilize an arbitrary coalition structure, or a particular coalition within it, rather than the grand coalition. We end the paper with a discussion of related work and some conclusions.

2 Preliminaries

Throughout this paper, given a vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ and a set $C \subseteq \{1, \ldots, n\}$ we write $x(C)$ to denote $\sum_{i \in C} x_i$.

Definition 1. *A* (transferable utility) coalitional game $G = (I, v)$ *is given by a set of* agents $I = \{1, \ldots, n\}$ *and a* characteristic function $v : 2^I \to \mathbb{R}^+$ *that for any subset (coalition) of agents lists the total utility these agents achieve by working together.*

A coalitional game $G = (I, v)$ is called *increasing* if for all coalitions $C' \subseteq C$ we have $v(C') \leq v(C)$, and *super-additive* if for all disjoint coalitions $C, C' \subseteq I$ we have $v(C) + v(C') \leq v(C \cup C')$. Note that since $v(C) \geq 0$ for any $C \subseteq I$, all super-additive games are increasing. Many multiagent domains can be viewed as super-additive coalitional games. Further, a coalitional game $G = (I, v)$ is called *simple* if $v(C) \in \{0, 1\}$ for all $C \subseteq I$. In a simple game, we say that a coalition $C \subseteq I$ *wins* if $v(C) = 1$, and *loses* if $v(C) = 0$. Finally, a coalitional game is called *anonymous* if $v(C) = v(C')$ for any $C, C' \subseteq I$ such that $|C| = |C'|$.

A particular class of simple games considered in this paper is that of *weighted voting games* (WVGs). In these games, each agent has a weight, and a coalition of agents wins the game if the sum of the weights of its members meets or exceeds a certain threshold.

Definition 2. *A* weighted voting game *is a simple coalitional game given by a set of agents* $I = \{1, \dots, n\}$*, a vector of agents'* weights $\mathbf{w} = (w_1, \dots, w_n)$ *and a* threshold q*. The* weight of a coalition $C \subseteq I$ *is* $w(C) = \sum_{i \in C} w_i$*. A coalition* C wins the game *(i.e.,* $v(C) = 1$*) if* $w(C) \geq q$*, and* loses the game *(i.e.,* $v(C) = 0$*) if* $w(C) < q$*.*

We denote the WVG with the weights $\mathbf{w} = (w_1, w_2, \dots, w_n)$ and the threshold q as $[\mathbf{w}; q]$ or $[w_1, w_2, \dots, w_n; q]$. Also, we set $w_{\max} = \max_{i \in I} w_i$. It is easy to see that WVGs are simple increasing games; however, they are not necessarily super-additive. Throughout this paper, we assume that $w(I) \geq q$, i.e., the grand coalition wins.

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

Definition 3. *Given a coalitional game* $G = (I, v)$*, a vector* $\mathbf{p} = (p_1, \dots, p_n) \in \mathbb{R}^n$ *is called an* imputation *for* G *if it satisfies* $p_i \geq 0$ *for each* i*,* $1 \leq i \leq n$*, and* $\sum_{i=1}^{n} p_i = v(I)$*. We call* p_i *the* payoff of agent i*; the* total payoff *of a coalition* $C \subseteq I$ *is given by* $p(C)$*. We write* $\mathcal{I}(G)$ *to denote the set of all imputations for* G*.*

For an imputation to be stable, it should be the case that no subset of players has an incentive to deviate. Formally, we say that a coalition C *blocks* an imputation $\mathbf{p} = (p_1, \dots, p_n)$ if $p(C) < v(C)$. The *core* of a coalitional game G is defined as the set of imputations not blocked by any coalition, i.e., $\mathrm{core}(G) = \{\mathbf{p} \in \mathcal{I}(G) \mid p(C) \geq v(C)$ for each $C \subseteq I\}$. An imputation in the core guarantees the stability of the grand coalition. However, the core can be empty, in which case every possible gain division is blocked by some coalition and the grand coalition is inherently unstable.

In WVGs, and, more generally, in simple games, one can characterize the core using the notion of veto players, i.e., players that are indispensable for forming a winning coalition. Formally, given a simple coalitional game $G = (I, v)$, a player $i \in I$ is said to be a *veto* player if for all coalitions $C \subseteq I \setminus \{i\}$ we have $v(C) = 0$. The following is a folklore result regarding nonemptiness of the core.

Theorem 1. *Let* $G = (I, v)$ *be a simple coalitional game. If there are no veto agents in* G*, then the core of* G *is empty. Otherwise, let* $I' = \{i_1, \dots, i_m\}$ *be the set of veto agents in* G*. Then the core of* G *is the set of imputations that distribute all the gains among the veto agents only, i.e.,* $\mathrm{core}(G) = \{\mathbf{p} \in \mathcal{I}(G) \mid p(I') = 1\}$*.*

So far, we have tacitly assumed that the only possible outcome of a coalitional game is the formation of the grand coalition. However, often it makes more sense for the agents to form several disjoint coalitions, each of which can focus on its own task. For example, WVGs can be used to model the setting where each agent has a certain amount of resources (modeled by her weight), and there is a number of identical tasks each of which requires a certain amount of these resources (modeled by the threshold) to be completed. In this setting, the formation of the grand coalition means that only one task will be completed, even if there are enough resources for several tasks.

The situation when agents can split into teams to work on several tasks simultaneously can be modeled using the notion of a coalition structure, i.e., a partition of the set of agents into disjoint coalitions. Formally, we say that $CS = (C^1, \dots, C^m)$ is a *coalition structure* over a set of agents I if $\cup_{i=1}^m C^i = I$ and $C^i \cap C^j = \emptyset$ for all $i \neq j$; we write $CS \in \mathcal{CS}(I)$. Also, we overload notation by writing $v(CS)$ to denote $\sum_{C^j \in CS} v(C^j)$. If coalition structures are allowed, an outcome of a game is not just an imputation, but a pair $(CS, \mathbf{p})$, where $\mathbf{p}$ is an imputation for the coalition structure CS, i.e., $\mathbf{p}$ distributes the gains of every coalition in CS among its members. Formally, we say that $\mathbf{p} = (p_1, \dots, p_n)$ is an *imputation for a coalition structure* $CS = (C^1, \dots, C^m)$ in a game $G = (I, v)$ if $p_i \geq 0$ for all i, $1 \leq i \leq n$, and $p(C^j) = v(C^j)$ for all j, $1 \leq j \leq m$; we write $\mathbf{p} \in \mathcal{I}(CS, G)$. We can also generalize the notion of the core introduced earlier in this section to games with coalition structures. Namely, given a game $G = (I, v)$, we say that an outcome $(CS, \mathbf{p})$ is in the *CS-core* of G if CS is a coalition structure over I, $\mathbf{p} \in \mathcal{I}(CS, G)$ and $p(C) \geq v(C)$ for all $C \subseteq I$; we write $(CS, \mathbf{p}) \in$ *CS-core*(G). Note that if $\mathbf{p}$ is in the core of G then $(I, \mathbf{p})$ is in the CS-core of G; however, the converse is not necessarily true.

3 The Cost of Stability

In many games, forming the grand coalition maximizes social welfare; this is the case, for example, if the game in question is super-additive. However, the core of such games may still be empty. In this case, it would be impossible to distribute the gains of the grand coalition in a stable way, so it may fall apart despite being socially optimal. Thus, an external party, such as a benevolent central authority, may want to incentivize the agents to cooperate, e.g., by offering the agents a supplemental payment Δ if they stay in the grand coalition. This situation can be modeled as an *adjusted coalitional game* derived from the original coalitional game G.

Definition 4. *Given a coalitional game $G = (I, v)$ and $\Delta > 0$, the* adjusted coalitional game *$G(\Delta) = (I, v')$ is given by $v'(C) = v(C)$ if $C \neq I$, and $v'(C) = v(C) + \Delta$ if $C = I$.*

We call $v'(I) = v(I) + \Delta$ the *adjusted gains* of the grand coalition. We say that a vector $\mathbf{p} \in \mathbb{R}^n$ is a *super-imputation* for a game $G = (I, v)$ if $p_i \geq 0$ for all $i \in I$ and $p(I) \geq v(I)$. Furthermore, we say that a super-imputation $\mathbf{p}$ is *stable* if $p(C) \geq v(C)$ for all $C \subseteq I$. A super-imputation $\mathbf{p}$ with $p(I) = v(I) + \Delta$ distributes the adjusted gains, i.e., it is an imputation for $G(\Delta)$; it is stable if and only if it is in the core of $G(\Delta)$. We say that a supplemental payment Δ *stabilizes* the grand coalition in a game G if the adjusted game $G(\Delta)$ has a nonempty core. Clearly, if Δ is large enough (e.g., $\Delta = n \max_{C \subseteq I} v(C)$), the game $G(\Delta)$ will have a nonempty core. However, usually the central authority wants to spend as little money as possible. Hence, we define the cost of stability as the *smallest* external payment that stabilizes the grand coalition.

Definition 5. *Given a coalitional game $G = (I, v)$, its* cost of stability *$CoS(G)$ is defined as $CoS(G) = \inf\{\Delta \mid \Delta \geq 0 \text{ and } \mathrm{core}(G(\Delta)) \neq \emptyset\}$.*

We have argued that the set $\{\Delta \mid \Delta \geq 0 \text{ and } \mathrm{core}(G(\Delta)) \neq \emptyset\}$ is nonempty. Therefore, $G(\Delta)$ is well-defined. Now, we prove that this set contains its greatest lower bound $CoS(G)$, i.e., that the game $G(CoS(G))$ has a nonempty core. While this can be shown using a continuity argument, we will now give a different proof, which will also be useful for exploring the cost of stability from an algorithmic perspective. Fix a coalitional game $G = (I, v)$ and consider the following linear program $\mathcal{LP}^*$ with variables $p_1, \ldots, p_n, \Delta$:

$$
\begin{aligned}
&\min \Delta \text{ subject to:} \\
&\quad \Delta \geq 0 && (1)\\
&\quad p_i \geq 0 \text{ for each } i, 1 \leq i \leq n, && (2)\\
&\sum_{i \in I} p_i = v(I) + \Delta && (3)\\
&\sum_{i \in C} p_i \geq v(C) \text{ for all } C \subseteq I && (4)
\end{aligned}
$$

It is not hard to see that the optimal value of this linear program is exactly $CoS(G)$. Moreover, any optimal solution of $\mathcal{LP}^*$ corresponds to an imputation in the core of $G(CoS(G))$ and therefore the game $G(CoS(G))$ has a nonempty core.

Example 1. Consider a WVG $G = [\mathbf{w}; q]$ that satisfies $w_1 = \cdots = w_n = w$. For such games, we will now provide an explicit formula for the cost of stability.

Theorem 2. *For a WVG $G = [w, w, \ldots, w; q]$, we have $CoS(G) = \frac{n}{\lceil q/w \rceil} - 1$.*

For example, if $w(n-1) < q \leq wn$, then $CoS(G) = 0$, i.e., G has a nonempty core. On the other hand, if $w = 1$, $n = 3k$ and $q = 2k$ for some integer $k > 0$, i.e., $q = \frac{2}{3}n$, we have $CoS(G) = \frac{3}{2} - 1 = \frac{1}{2}$.

3.1 Bounds on $CoS(G)$ in General Coalitional Games

Consider an arbitrary coalitional game $G = (I, v)$. Clearly, $CoS(G) = 0$ if and only if G has a nonempty core. Further, we have argued that $CoS(G)$ is upper-bounded by $n \max_{C \subseteq I} v(C)$, i.e., $CoS(G)$ is finite for any fixed coalitional game. Moreover, the bound of $n \max_{C \subseteq I} v(C)$ is (almost) tight. To see this, consider a (monotone, simple) game G' given by $v'(\emptyset) = 0$ and $v'(C) = 1$ for all $C \neq \emptyset$. Clearly, we have $CoS(G') = n-1$: any super-imputation that pays some agent less than 1 will not be stable, whereas setting $p_i = 1$ for all $i \in I$ ensures stability. Thus, the cost of stability can be quite large relative to the value of the grand coalition.

On the other hand, we can provide a lower bound on $CoS(G)$ in terms of the values of coalition structures over I. Indeed, for an arbitrary coalition structure $CS \in \mathcal{CS}(I)$, we have $CoS(G) \geq v(CS) - v(I)$. To see this, note that if the total payment to the grand coalition is less than $(v(CS) - v(I)) + v(I)$, then for some coalition $C \in CS$ it will be the case that $p(C) < v(C)$. It is tempting to conjecture that $CoS(G) = \max_{CS \in \mathcal{CS}(I)} v(CS) - v(I)$. However, a counterexample is provided by Example 1 with $w = 1$, $q = \frac{2}{3}n$: indeed, in this case we have $CoS(G) = \frac{1}{2}$, $\max_{CS \in \mathcal{CS}(I)} v(CS) - v(I) = 0$. We can summarize these observations as follows.

Theorem 3. *For any coalitional game $G = (I, v)$, we have*

$$\max_{CS \in \mathcal{CS}(I)} v(CS) - v(I) \leq CoS(G) \leq n \max_{C \subseteq I} v(C).$$

For super-additive games, we can strengthen the upper bound considerably. Note that in such games the grand coalition maximizes social welfare, so its stability is particularly desirable. Yet, as the second part of Theorem 4 implies, ensuring stability may turn out to be quite costly even in this restricted setting.

Theorem 4. *For any super-additive game $G = (I, v)$, $|I| = n$, we have $CoS(G) \leq (\sqrt{n} - 1)v(I)$, and this bound is asymptotically tight.*

For anonymous super-additive games, further improvements are possible.

Theorem 5. *For any anonymous super-additive game $G = (I, v)$, we have $CoS(G) \leq 2v(I)$, and this bound is asymptotically tight.*

A somewhat similar stability-related concept is the *least core*, which is the set of all imputations $\mathbf{p}$ that minimize the maximal *deficit* $v(C) - p(C)$. In particular, the *value* of the least core $\varepsilon(G)$, defined as $\varepsilon(G) = \min\{\max_{C \subseteq I} v(C) - p(C) \mid \mathbf{p} \in \mathcal{I}(G)\}$, is strictly positive if and only if the cost of stability is strictly positive. We can generalize this observation as follows.

Proposition 1. *For any coalitional game $G = (I, v)$ with $v(\emptyset) = 0$ and $|I| = n$ such that $\varepsilon(G) \geq 0$, we have $CoS(G) \leq n\varepsilon(G)$, and this bound is tight.*

3.2 Algorithmic Properties of $CoS(G)$

The linear program $\mathcal{LP}^*$ provides a way of computing $CoS(G)$ for any coalitional game G. However, this linear program contains exponentially many constraints (one for each subset of I). Thus, solving it directly would be too time-consuming for most games. Note that for general coalitional games, this is, in a sense, inevitable: in general, a coalitional game is described by its characteristic function, i.e., a list of 2^n numbers. Thus, to discuss the algorithmic properties of $CoS(G)$, we need to restrict our attention to games with compactly representable characteristic functions.

A standard approach to this issue is to consider games that can be described by polynomial-size circuits. Formally, we say that a class $\mathcal{G}$ of games has a *compact circuit representation* if there exists a polynomial p such that for every $G \in \mathcal{G}$, $G = (I, v)$, $|I| = n$, there exists a circuit $\mathcal{C}$ of size $p(n)$ with n binary inputs which on input $(b_1, \ldots, b_n)$ outputs $v(C)$, where $C = \{i \in I \mid b_i = 1\}$.

Unfortunately, it turns out that having a compact circuit representation does not guarantee efficient computability of $CoS(G)$. Indeed, it is easy to see that WVGs with integer weights have such a representation. However, in the next section we will show that computing $CoS(G)$ for such games is computationally intractable (Theorem 7). We can, however, provide a *sufficient* condition for $CoS(G)$ to be efficiently computable. To do so, we will first formally state the relevant computational problems. In the next two descriptions, we assume that G is given by its compact circuit representation.

SUPER-IMPUTATION-STABILITY: Given a coalitional game G, a supplemental payment Δ and an imputation $\mathbf{p} = (p_1, \ldots, p_n)$ in the adjusted game $G(\Delta)$, decide whether $\mathbf{p} \in \mathrm{core}(G(\Delta))$.

COS: Given a coalitional game G and a parameter Δ, decide whether $CoS(G) \leq \Delta$, i.e., whether $\mathrm{core}(G(\Delta)) \neq \emptyset$.

Consider first SUPER-IMPUTATION-STABILITY. Fix a game $G = (I, v)$. For any super-imputation $\mathbf{p}$ for G, let $d(G, \mathbf{p}) = \max_{C \subseteq I}\{v(C) - p(C)\}$ be the maximum deficit of a coalition under $\mathbf{p}$. Clearly, $\mathbf{p}$ is stable if and only if $d(G, \mathbf{p}) \leq 0$. Observe also that for any $\Delta > 0$ it is easy to decide whether $\mathbf{p}$ is an imputation for $G(\Delta)$. Thus, a polynomial-time algorithm for computing $d(G, \mathbf{p})$ can be converted into a polynomial-time algorithm for SUPER-IMPUTATION-STABILITY. Further, we can compute COS via solving $\mathcal{LP}^*$ by the ellipsoid method. The ellipsiod method runs in polynomial time given a polynomial-time *separation oracle*, i.e., a procedure that takes as input a candidate feasible solution, checks if it indeed is feasible, and if this is not the case, returns a violated constraint. Now, given a vector $\mathbf{p}$ and a parameter Δ, we can easily check if they satisfy constraints (1)–(3), i.e., if $\mathbf{p}$ is an imputation for $G(\Delta)$. To verify constraint (4), we need to check if $\mathbf{p}$ is in the core of $G(\Delta)$. As argued above, this can be done by checking whether $d(G, \mathbf{p}) \leq 0$. We summarize these results as follows.

Theorem 6. *Consider a class of coalitional games $\mathcal{G}$ with a compact circuit representation. If there is an algorithm that for any $G \in \mathcal{G}$, $G = (I, v)$, $|I| = n$, and for any super-imputation $\mathbf{p}$ for G computes $d(G, \mathbf{p})$ in time $\mathrm{poly}(n, |\mathbf{p}|)$, where $|\mathbf{p}|$ is the number of bits in the binary repesentation of $\mathbf{p}$, then for any $G \in \mathcal{G}$ the problems* SUPER-IMPUTATION-STABILITY *and* COS *are polynomial-time solvable.*

4 Computing the Cost of Stability in Weighted Voting Games Without Coalition Structures

In this section, we focus on computing the cost of stabilizing the grand coalition in WVGs. We start by considering the complexity of exact algorithms for this problem.

4.1 Exact Algorithms

In what follows, unless specified otherwise, we assume that all weights and the threshold are integers given in binary, whereas all other numeric parameters, such as the supplemental payment Δ and the entries of the payoff vector $\mathbf{p}$, are rationals given in binary. Standard results on linear threshold functions [13] imply that WVGs with integer weights have a compact circuit representation. Thus, we can define the computational problems SUPER-IMPUTATION-STABILITY-WVG and COS-WVG by specializing the problems SUPER-IMPUTATION-STABILITY and COS to WVGs. Both of the resulting problems turn out to be computationally hard.

Theorem 7. *The problem* SUPER-IMPUTATION-STABILITY-WVG *is* coNP*-complete and the problem* COS-WVG *is* coNP*-hard.*

The reductions in the proof of Theorem 7 are from PARTITION. Consequently, our hardness results depend in an essential way on the weights being given in binary. Thus, it is natural to ask what happens if the agents' weights are polynomially bounded (or given in unary). It turns out that in this case SUPER-IMPUTATION-STABILITY-WVG and COS are in P. This can be shown using the technique of Section 3.2, i.e., by proving that for WVGs with small weights one can compute $d(G, \mathbf{p})$ in polynomial time.

Theorem 8. SUPER-IMPUTATION-STABILITY-WVG *and* COS-WVG *are in* P *when the agents' weights are polynomially bounded (or given in unary).*

4.2 Approximating the Cost of Stability in Weighted Voting Games

For large weights, the algorithms outlined in the end of the previous section may not be practical. Thus, the center may want to trade off its payment and computation time, i.e., provide a slightly higher supplemental payment for which the corresponding stable super-imputation can be computed efficiently. It turns out that this is indeed possible, i.e., $CoS(G)$ can be efficiently approximated to an arbitrary degree of precision.

Theorem 9. *There exists an algorithm $\mathcal{A}(G, \varepsilon)$ that, given a WVG $G = [\mathbf{w}; q]$ in which the weights of all players are nonnegative integers given in binary and a parameter $\varepsilon > 0$, outputs a value Δ that satisfies $CoS(G) \leq \Delta \leq (1 + \varepsilon)CoS(G)$ and runs in time $\mathrm{poly}(n, \log w_{\max}, 1/\varepsilon)$. That is, there exists a fully polynomial-time approximation scheme (FPTAS) for $CoS(G)$.*

Moreover, one can get a 2-approximation to the adjusted gains simply by paying each agent in proportion to her weight. Formally, fix a weighted voting game $G = (I, v)$ and let $\mathbf{p}^* = (p_1^*, \ldots, p_n^*)$ be a super-imputation given by $p_i^* = \min\{1, \frac{w_i}{q}\}$.

Theorem 10. *For any weighted voting game $G = (I, v)$ with $CoS(G) = \Delta$ and any $\mathbf{p} \in \mathrm{core}(G(\Delta))$, we have $p^*(I) \leq 2\mathbf{p}(I)$.*

It can also be shown that this bound is tight (see Appendix C).

5 Cost of Stability in Games with Coalition Structures

If a coalitional game is not super-additive, the formation of the grand coalition is not necessarily the most desirable outcome: for example, it may be the case that by splitting into several teams the agents can accomplish more tasks than by working together. In such settings, the central authority may want to stabilize a coalition structure, i.e., a partition of agents into teams. We now generalize our definition of the cost of stability to such settings.

5.1 Stabilizing a Fixed Coalition Structure

We first consider the setting where the central authority wants to stabilize a particular coalition structure.

Given a coalitional game $G = (I, v)$, a coalition structure $CS = (C^1, \dots, C^m)$ over I and a vector $\boldsymbol{\Delta} = (\Delta^1, \dots, \Delta^m)$, let $G(\boldsymbol{\Delta})$ be the game with the set of agents I and the characteristic function v' given by $v'(C^i) = v(C^i) + \Delta^i$ for $i = 1, \dots, m$ and $v'(C) = v(C)$ for any $C \notin \{C^1, \dots, C^m\}$. We say that the game $G(\boldsymbol{\Delta})$ is *stable with respect to* CS if there exists an imputation $\mathbf{p} \in \mathcal{I}(CS, G(\boldsymbol{\Delta}))$ such that $(CS, \mathbf{p})$ is in the CS-core of $G(\boldsymbol{\Delta})$. Also, we say that an external payment Δ *stabilizes* a coalition structure CS with respect to a game G if there exist $\Delta^1 \geq 0, \dots, \Delta^m \geq 0$ such that $\Delta = \Delta^1 + \dots + \Delta^m$ and the game $G(\boldsymbol{\Delta})$ is stable with respect to CS. We are now ready to define the cost of stability of a coalition structure CS in G.

Definition 6. *Given a coalitional game $G = (I, v)$ and a coalition structure $CS = (C^1, \dots, C^m)$ over I, the* cost of stability $CoS(CS, G)$ *of the coalition structure CS in G is the smallest external payment needed to stabilize CS, i.e.,*

$$CoS(CS, G) = \inf\{\sum_{i=1}^{m} \Delta^i \mid \Delta^i \geq 0 \text{ for } i = 1, \dots, m \quad \text{and} \quad \exists \mathbf{p} \in \mathcal{I}(CS, G(\boldsymbol{\Delta})) \quad \text{s.t.} \quad (CS, \mathbf{p}) \in \textit{CS-core}(G(\boldsymbol{\Delta}))\}.$$

Fix a game $G = (I, v)$ and set $v_{\max} = \max_{C \subseteq I} v(C)$. It is easy to see that for any coalition structure $CS = (C^1, \dots, C^m)$ the game $G(\boldsymbol{\Delta})$, where $\Delta_i = |C^i| v_{\max}$, is stable with respect to CS, and therefore $CoS(CS, G)$ is well-defined and satisfies $CoS(CS, G) \leq n v_{\max}$. Moreover, as in the case of games without coalition structures, the value $CoS(CS, G)$ can be obtained as an optimal solution to a linear program. Indeed, we can simply take the linear program $\mathcal{LP}^*$ and replace the constraint $\sum_{i \in I} p_i = v(I) + \Delta$ with the constraint $\sum_{i \in I} p_i = v(CS) + \Delta$. It is not hard to see that the resulting linear program, which we will denote by $\mathcal{LP}^*_{CS}$, computes $CoS(CS, G)$: in particular, the constraints $\Delta^i \geq 0$ for $i = 1, \dots, m$ are implicitly captured by constraints $\sum_{i \in C^i} p_i \geq v(C^i)$ in line (4) of $\mathcal{LP}^*_{CS}$.

We now turn to the question of computing the cost of stability of a given coalition structure in WVGs. To this end, we will modify the decision problems stated in Section 4.1 as follows.

COS-WVG-CS: Given a WVG $G = [\mathbf{w}; q]$ with the set of agents I, a coalition structure CS over I and a parameter Δ, decide whether $CoS(CS, G) \leq \Delta$.

SUPER-IMPUTATION-STABILITY-WVG-CS: Given a WVG $G = [\mathbf{w}; q]$ with the set of agents I, a coalition structure $CS = (C^1, \dots, C^m)$ over I, a vector $\boldsymbol{\Delta} = (\Delta^1, \dots, \Delta^m)$ and an imputation $\mathbf{p} \in \mathcal{I}(CS, G(\boldsymbol{\Delta}))$, decide if $(CS, \mathbf{p})$ is in the CS-core of $G(\boldsymbol{\Delta})$.

The results of Section 4.1 immediately imply that both of these problems are computationally hard even for $m = 1$. Moreover, using the results of [7], we can show that SUPER-IMPUTATION-STABILITY-WVG-CS remains coNP-complete even if $\boldsymbol{\Delta}$ is fixed to be $(0, \dots, 0)$. On the other hand, when weights are integers given in unary, both COS-WVG-CS and SUPER-IMPUTATION-STABILITY-WVG-CS are polynomial-time solvable. Indeed, to solve SUPER-IMPUTATION-STABILITY-WVG-CS, one needs to check if there is a coalition C with $w(C) \geq q$, $p(C) < 1$. This can be done using the dynamic programming algorithm from the proof of Theorem 8. Moreover, to solve CoS-WVG-CS, we can simply run the ellipsoid algorithm on the linear program $\mathcal{LP}^*_{CS}$ described earlier in this section, using the algorithm for SUPER-IMPUTATION-STABILITY-WVG-CS as a separation oracle. Thus, we obtain the following result.

Theorem 11. *When all players' weights are integers given in unary, the problems* CoS-WVG-CS *and* SUPER-IMPUTATION-STABILITY-WVG-CS *are in* P.

Finally, it is not hard to see that we can adapt the approximation algorithm presented in Section 4.2 to this setting.

Theorem 12. *There exists an FPTAS for $CoS(CS, G)$ in WVGs.*

5.2 Finding the Cheapest Coalition Structure to Stabilize

So far, we have focused on the setting where the external party wants to stabilize a particular coalition structure. However, it can also be the case that the central authority simply wants to achieve stability, and does not care which coalition structure arises, as long as it can be made stable using as little as money as possible. We will now introduce the notion of *cost of stability for games with coalition structures* to capture this type of setting. Recall that $\mathcal{CS}(I)$ denotes the set of all coalition structures over I.

Definition 7. *Given a coalitional game $G = (I, v)$, let the* cost of stability for G with coalition structures*, denoted by $CoS_{CS}(G)$, be* $\min\{CoS(CS, G) \mid CS \in \mathcal{CS}(I)\}$.

Clearly, one can compute $CoS_{CS}(G)$ by enumerating all coalition structures over I and picking the one with the smallest value of $CoS(CS, G)$. Alternatively, note that the linear program $\mathcal{LP}^*_{CS}$ depends only on the value of the coalition structure CS. Hence, stabilizing all coalition structures with the same total value has the same cost. Moreover, this implies that the cheapest coalition structure to stabilize is the one that maximizes social welfare. Hence, if we could compute the value of the coalition structure CS^* that maximizes social welfare, we could find $CoS_{CS}(G)$ by solving $\mathcal{LP}^*_{CS^*}$.

For WVGs, Theorem 2 in [7] shows that if weights are given in binary, it is NP-hard to decide whether a given game has a nonempty CS-core. As this question is equivalent to asking whether $CoS_{CS}(G) = 0$, the latter problem is NP-hard, too.

One might hope that computing $CoS_{CS}(G)$ is easy if the weights of all players are given in unary. However, this does not seem to be the case. Indeed, our algorithms for computing the cost of stability in other settings relied on solving the corresponding linear program. To implement this approach in our scenario, we would need to compute the value of the coalition structure that maximizes social welfare. However, a straightforward reduction from 3-PARTITION, a classic problem that is known to be NP-hard even for unary weights, shows that the latter problem is NP-hard even if weights are given in unary. While this does not immediately imply that computing $CoS_{CS}(G)$ is hard for small weights, it means that finding the cheapest-to-stabilize outcome is NP-hard even if weights are given in unary.

5.3 Stabilizing a Particular Coalition

We now consider the case where the central authority wants a particular group of agents to work together, but does not care about the stability of the overall game. Thus, it wants to identify a coalition structure containing a particular coalition C and the minimal subsidy to the players that ensures that no set of players that includes members of C wants to deviate. We skip the formal definition of the corresponding cost-of-stability concept, as well as its algorithmic analysis due to space constraints. However, we would like to mention several subtle points that arise in this context.

First, one might think that the optimal way to stabilize a coalition is to offer payments to members of this coalition only. However, this turns out to be not true (see Example 2 in Appendix D). Second, stabilizing a given coalition may be strictly cheaper than stabilizing *any* of the coalition structures that contain it (see Example 3 in Appendix D). Thus choosing a good definition of the cost of stability of an individual coalition is a nontrivial issue.

6 Related Work

The computational complexity of various stability concepts in coalitional games has been studied in a number of papers (see, e.g., [4, 11, 5, 19]). In particular, paper [8] analyzes computational aspects of stability in WVGs, proving a number of results on the computational complexity of the ε-core, the least core and the nucleolus. The computational complexity of the CS-core in WVGs is studied in [7].

Paper [12] is quite similar to ours in spirit. It considers the setting where an external party intervenes in order to achieve a certain outcome using monetary payments. However, [12] deals with the very different domain of *non*cooperative games.

One can also draw a parallel between the subject of this paper and the computational analysis of bribery in elections [9], i.e., the setting where an external party, whose goal it is to get a given candidate elected, pays the voters to change their preferences. However, while in the context of our work it is natural to view the central authority as benevolent, the usual interpretation of bribery is much less benign.

7 Conclusion

We have examined the possibility of stabilizing a coalitional game by offering the agents additional payments in order to discourage them from deviating, and defined the cost of stability as the minimal total payment that allows a stable division of the gains. We then focused on the computational aspects of this concept for weighted voting games. In the setting where the outcome to be stabilized is the grand coalition, we provided a complete picture of the computational complexity of the related decision problems. We then showed how to extend our results to settings where agents can form a coalition structure.

There are several lines of possible future research. First, while the focus of this paper was on weighted voting games, the notion of the cost of stability is defined for any coalitional game. Therefore, a natural research direction is to study the cost of stability in other classes of games. Second, we would like to develop a better understanding of the relationship between the cost of stability of a game, and its least core and nucleolus. Finally, it would be interesting to extend the notion of the cost of stability to games with nontransferable utility and partition function games.

References

1. R. J. Aumann and S. Hart, editors. *Handbook of Game Theory, with Economic Applications, Vol. 2*. North-Holland, Amsterdam, 1994.
2. R. Branzei, D. Dimitrov, and S. Tijs. *Models in Cooperative Game Theory*. Springer, 2008.
3. K. Chatterjee, B. Dutta, and K. Sengupta. A noncooperative theory of coalitional bargaining. *Review of Economic Studies*, 60:463–477, 1993.
4. V. Conitzer and T. Sandholm. Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proc. of AAAI-04*, 2004.
5. V. Conitzer and T. Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6-7):607–619, 2006.
6. E. Ephrati and J. S. Rosenschein. The Clarke Tax as a consensus mechanism among automated agents. In *Proc. of IJCAI-91*, 1991.
7. E. Elkind, G. Chalkiadakis, and N. R. Jennings. Coalition structures in weighted voting games. In *Proc. of ECAI-08*, 2008
8. E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge. Computational complexity of weighted threshold games. In *Proc. of AAAI-07*, 2007.
9. P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of bribery in elections. In *Proc. of AAAI-06*, 2006.
10. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.
11. S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *Proc. of EC-05*, 2005.
12. D. Monderer and M. Tennenholtz. K-implementation. *Journal of Artificial Intelligence Research*, 21:37–62, 2004.
13. S. Muroga. Threshold Logic and its Applications. John Wiley & Sons, 1971.
14. A. Okada. A Noncooperative coalitional bargaining game with random proposers. *Games and Economic Behavior*, 16:97–108, 1996.
15. J. S. Rosenschein and M. R. Genesereth. Deals among rational agents. In *Proc. of IJCAI-85*, 1985.
16. T. Sandholm and V. Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *Proc. of ICMAS-95*, 1995.
17. A. Taylor and W. Zwicker. *Simple Games: Desirability Relations, Trading, Pseudoweightings*. Princeton University Press, Princeton, 1999.
18. M. P. Wellman. The economic approach to artificial intelligence. *ACM Computing Surveys*, 27:360–362, 1995.
19. M. Yokoo, V. Conitzer, T. Sandholm, N. Ohta, and A. Iwasaki. Coalitional games in open anonymous environments. In *Proc. of AAAI-05*, 2005.

A Proofs for Section 3

Theorem 2 For a WVG $G = [w, w, \dots, w; q]$, we have $CoS(G) = \frac{n}{\lceil q/w \rceil} - 1$.

Proof. First, note that by scaling w and q we can assume that $w = 1$.

Set $\Delta = \frac{n}{\lceil q \rceil} - 1$ and consider the imputation $\mathbf{p} = (p_1, \dots, p_n)$ given by $p_i = \frac{1}{\lceil q \rceil}$ for i, $1 \le i \le n$. Clearly, we have $p(I) = \frac{n}{\lceil q \rceil}$, so $\mathbf{p} \in G(\Delta)$. Moreover, for any winning coalition C, we have $|C| \ge \lceil q \rceil$, so $p(C) \ge \lceil q \rceil \frac{1}{\lceil q \rceil} = 1$. Therefore, $\mathbf{p}$ is in the core of $G(\Delta)$, and hence $CoS(G) \le \Delta$.

On the other hand, consider any stable super-imputation $\mathbf{p}$. Set $s = \lceil q \rceil$. Clearly, for any coalition C with $|C| = s$ we have $p(C) \ge 1$. Now, consider a collection of coalitions $C^1, \dots, C^n$, where $C^i = \{i \mod n, i+1 \mod n, \dots, i+s-1 \mod n\}$: for example, we have $C^{n-1} = \{n-1, n, 1, \dots, s-2\}$. We have $|C^i| = s$ for all i, $1 \le i \le n$, so $p(C^1) + \dots + p(C^n) \ge n$. On the other hand, each player i occurs in exactly s of these coalitions, so we have $p(I)s = p(C^1) + \dots + p(C^n)$. Hence, $p(I) \ge n/s = \frac{n}{\lceil q \rceil}$ and therefore $CoS(G) \ge \Delta$. □

Theorem 4 For any super-additive game $G = (I, v)$, $|I| = n$, we have $CoS(G) \le (\sqrt{n} - 1)v(I)$, and this bound is asymptotically tight.

Proof. Fix an arbitrary monotone super-additive game $G = (I, v)$ with $v(\emptyset) = 0$ and $|I| = n$. Consider the corresponding linear program $\mathcal{LP}^*$. Observe that it can be rewritten as

$$\begin{aligned} \min \textstyle\sum_{i \in I} p_i \quad & \text{subject to:} \\ p_i \ge \quad 0 \quad & \text{for } i = 1, \dots, n, \\ \sum_{i \in C} p_i \ge \quad v(C) \quad & \text{for all } C \subseteq I \end{aligned}$$

The dual to this linear program has 2^n variables $\{\lambda_C\}_{C \subseteq I}$ and is given by

$$\begin{aligned} \max \textstyle\sum_{C \subseteq I} v(C)\lambda_C \quad & \text{subject to:} \\ \lambda_C \ge \quad 0 \quad & \text{for all } C \subseteq I \\ \sum_{C \ni i} \lambda_C \le \quad 1 \quad & \text{for } i = 1, \dots, n, \end{aligned}$$

That is, we have to assign "weights" λ_C to all coalitions so that the total weight of all coalitions covering any given point is at most 1. Our goal is to maximize $\sum_{C \subseteq I} v(C)\lambda_C$ subject to this condition.

First, we claim that there exists an optimal solution to this maximization problem that satisfies $S \cap T \ne \emptyset$ for any S, T such that $\lambda_S > 0$, $\lambda_T > 0$. Indeed, suppose that this is not the case. Fix an arbitrary order $\prec$ on coalitions in 2^I such that $|S| < |T|$ implies $S \prec T$, and extend it to a lexicographic order on tuples of subsets of I in the standard manner. For every optimal solution $(\lambda_C)_{C \subseteq I}$ to the dual program, consider the vector $\xi_{(\lambda_C)_{C \subseteq I}}$ whose entries are the subsets $C \subseteq I$ with $\lambda_C = 0$, ordered according to $\prec$ (from the smallest to the largest). Among all optimal solutions to the dual linear program, pick one with the lexicographically largest such vector and denote it by $(\lambda^*_C)_{C\subseteq I}$. By our assumption, there exists a pair of nonempty sets (S,T) such that $\lambda^*_S > 0$, $\lambda^*_T > 0$, but $S \cap T = \emptyset$. Let $\varepsilon = \min\{\lambda^*_S, \lambda^*_T\}$. Consider the vector $(\lambda^{**}_C)_{C\subseteq I}$ given by

$$\lambda^{**}_C = \begin{cases} \lambda^*_C & \text{for } C \neq S, T, S\cup T, \\ \lambda^*_C - \varepsilon & \text{for } C = S, T, \\ \lambda^*_C + \varepsilon & \text{for } C = S \cup T. \end{cases}$$

First, observe that since S and T are disjoint, $(\lambda^{**}_C)_{C\subseteq I}$ is also a feasible solution to the dual program. Furthermore, by super-additivity we have

$$\sum_{C\subseteq I} v(C)\lambda^{**}_C = \sum_{C\subseteq I} v(C)\lambda^*_C - v(S)\varepsilon - v(T)\varepsilon + (v(S)+v(T))\varepsilon \geq \sum_{C\subseteq I} v(C)\lambda^*_C,$$

so $(\lambda^{**}_C)_{C\subseteq I}$ is an optimal solution to the dual program, too. Finally, observe that $\xi_{(\lambda^{**}_C)_{C\subseteq I}}$ is lexicographically greater than $\xi_{(\lambda^*_C)_{C\subseteq I}}$. Indeed, assume that $\varepsilon = \lambda^*_S$ (a similar argument works for $\varepsilon = \lambda^*_T$). Then, for $C \neq S, T, S\cup T$, we have $\lambda^*_C = \lambda^{**}_C$, and, moreover, $\lambda^*_S \neq 0$, $\lambda^{**}_S = 0$ and $|S\cup T| > |S|$. This is a contradiction with our choice of $(\lambda^*_C)_{C\subseteq I}$.

Thus, there is an optimal solution $(\lambda_C)_{C\subseteq I}$ in which any two sets C and C' with $\lambda_C \neq 0$ and $\lambda_{C'} \neq 0$ intersect. Now, suppose that there is a set S with $|S| \leq \sqrt{n}$, $\lambda_S > 0$. Any set T with $\lambda_T > 0$ contains one of the points in S. Thus, we have

$$\sum_{C\subseteq I} \lambda_C v(C) \leq v(I) \sum_{i\in S} \sum_{T: i\in T} \lambda_T \leq v(I) \sum_{i\in S} 1 \leq \sqrt{n}v(I).$$

On the other hand, if for any C with $\lambda_C > 0$ it holds that $|C| > \sqrt{n}$, we have

$$\sqrt{n}\sum_{C\subseteq I} \lambda_C v(C) \leq \sum_{C\subseteq I} \lambda_C v(C)|C| \leq v(I) \sum_{i\in I} \sum_{C: i\in C} \lambda_C \leq nv(I),$$

so $\sum_{C\subseteq I} \lambda_C v(C) \leq nv(I)/\sqrt{n} = \sqrt{n}v(I)$. Consequently, in both cases we have $\sum_{C\subseteq I} \lambda_C v(C) \leq \sqrt{n}v(I)$. Now, since the optima of the dual and the original linear programs are equal, the optimal solution $(p_1, \dots, p_n)$ to the original linear program satisfies $\sum_{i\in I} p_i \leq \sqrt{n}v(I)$, and hence $CoS(G) \leq (\sqrt{n}-1)v(I)$, as required.

To see that this bound is tight, consider a finite projective plane P of order q, where q is a prime number. It has q^2+q+1 points and the same number of lines, every line contains $q+1$ points, any two lines intersect, and any point belongs to exactly $q+1$ lines. Now, consider a simple coalitional game whose players correspond to points in P and whose winning coalitions correspond to sets of points in P that contain a line. Observe that this game is super-additive: since any two lines intersect, there do not exist two disjoint winning coalitions. Hence, for any $S, T \subseteq I$ such that $S \cap T = \emptyset$ either $v(S) = 0$ or $v(T) = 0$, and therefore $v(S) + v(T) \leq v(S\cup T)$, as required. On the other hand, for each line C, we have $\sum_{i\in C} p_i \geq 1$. Summing over all q^2+q+1 lines, and using the fact that each point belongs to $q+1$ lines, we obtain $(q+1)\sum_{i\in I} p_i \geq q^2+q+1$, i.e., $p(I) = \frac{q^2+q+1}{q+1} = q + \frac{1}{q+1}$. Since $n = |I| = q^2+q+1$, we have $q \geq \sqrt{n}-1$, i.e., $CoS(G) \geq (\sqrt{n}-2)v(I)$. □

Theorem 5 For any anonymous super-additive game $G = (I, v)$, we have $CoS(G) \leq 2v(I)$, and this bound is asymptotically tight.

Proof. Fix an anonymous super-additive game $G = (I, v)$ with $|I| = n$. Consider a super-imputation $\mathbf{p} = (p_1, \ldots, p_n)$ given by $p_i = \frac{2v(I)}{n}$. Clearly, we have $p(I) = 2v(I)$. It remains to show that $\mathbf{p}$ is in the core of the adjusted game $G(v(I))$.

For any coalition $C \subset I$, there exists an integer k, $1 \leq k \leq n-1$, such that $\frac{n}{k+1} \leq |C| < \frac{n}{k}$. For this value of k, one can construct k pairwise disjoint coalitions $C_1, \ldots, C_k$ with $C_1 = C$ and $|C_1| = \cdots = |C_k|$. Super-additivity then implies that $v(C) \leq \frac{v(I)}{k}$. On the other hand, we have

$$p(C) = |C|\frac{2v(I)}{n} \geq \frac{n}{k+1} \cdot \frac{2v(I)}{n} = \frac{2v(I)}{k+1}.$$

Since $\frac{2v(I)}{k+1} \geq \frac{v(I)}{k}$ for any $k \geq 1$, it follows that $p(C) \geq v(C)$ for all $C \subset I$, so $\mathbf{p}$ is stable.

To see that this bound is tight, consider a game $G = (I, v)$ with $|I| = n = 2k + 1$ given by $v(C) = 0$ if $|C| \leq k$, and $v(C) = 1$ if $|C| \geq k + 1$. Clearly, this game is anonymous. Moreover, as any two winning coalitions intersect, this game is also super-additive. Consider any stable super-imputation $\mathbf{p}$ for this game. For any C with $|C| = k + 1$, we have $\sum_{i \in C} p_i \geq 1$. There are exactly $\binom{n}{k+1}$ coalitions of this size, and each agent participates in exactly $\binom{n-1}{k}$ such coalitions. Thus, summing all these inequalities, we obtain $\binom{n-1}{k} p(I) \geq \binom{n}{k+1}$, or, canceling, $p(I) \geq \frac{n}{k+1} = 2 - \frac{1}{k+1}$. □

Proposition 1 For any coalitional game $G = (I, v)$ with $v(\emptyset) = 0$ and $|I| = n$ such that $\varepsilon(G) \geq 0$, we have $CoS(G) \leq n\varepsilon(G)$, and this bound is tight.

Proof. Clearly, if $\varepsilon(G) = 0$, we have $CoS(G) = 0$. Now, assume $\varepsilon(G) > 0$. Let $\mathbf{p}$ be an imputation in the least core of G. For any $C \subseteq I$ we have $p(C) \geq v(C) - \varepsilon(G)$. Consider a super-imputation $\mathbf{p}^*$ given by $p^*_i = p_i + \varepsilon(G)$. Clearly, we have $p^*(C) \geq v(C)$ for any $C \subseteq I$ such that $C \neq \emptyset$, i.e., $\mathbf{p}^*$ is stable. Further, it is easy to see that $p^*(I) = v(I) + n\varepsilon(G)$, so $CoS(G) \leq n\varepsilon(G)$.

To see that this bound is tight, reconsider the game $G = (I, v)$ with $|I| = n$, $v(\emptyset) = 0$, and $v(C) = 1$ for all $C \neq \emptyset$. It is easy to see that $\varepsilon(G) = \frac{n-1}{n}$, since the imputation $(\frac{1}{n}, \ldots, \frac{1}{n})$ is in the least core of G. On the other hand, as mentioned above, $CoS(G) = n - 1 = n\varepsilon(G)$. □

B Proofs for Section 4.1

Theorem 7 The problem SUPER-IMPUTATION-STABILITY-WVG is coNP-complete and the problem COS-WVG is coNP-hard.

Proof. Both of our reductions will be from PARTITION, a well-known NP-complete problem [10], which is defined as follows: given a list $A = (a_1, \ldots, a_n)$ of nonnegative integers such that $\sum_{i=1}^{n} a_i = 2K$, decide whether there is a sublist A' of A such that $\sum_{a_i \in A'} a_i = K$.

We first show that CoS-WVG is coNP-hard. Given an instance $A = (a_1, \dots, a_n)$ of PARTITION, we construct a weighted voting game G by setting $I = \{1, \dots, n\}$, $w_i = a_i$ for each i, $1 \le i \le n$, and $q = K$. Set $\Delta = \frac{K-1}{K+1}$. We claim that (G, Δ) is a "yes"-instance of CoS-WVG if and only if A is a "no"-instance of PARTITION.

Indeed, suppose that A is a "yes"-instance of PARTITION, and let A' be the corresponding sublist. Set $I' = \{i \mid a_i \in A'\}$ and $I'' = I \setminus I'$. Suppose for the sake of contradiction that $G(\Delta)$ has a nonempty core, and let $\mathbf{p}$ be an imputation in the core of $G(\Delta)$. We have $p(I) = \frac{2K}{K+1} < 2$, and hence either $p(I') < 1$ or $p(I'') < 1$ (or both). On the other hand, since $\sum_{i \in I'} a_i = K$, we have $w(I') = w(I'') = K = q$, i.e., at least one of the coalitions I' and I'' has a rational incentive to deviate, a contradiction.

On the other hand, suppose that A is a "no"-instance of PARTITION, and consider a vector $\mathbf{p}^* = (p^*_1, \dots, p^*_n)$, where $p^*_i = \frac{w_i}{K+1}$. We have $p^*(I) = \frac{2K}{K+1}$, and hence $p^*(I) - v(I) = \frac{K-1}{K+1}$. That is, $\mathbf{p}^*$ is an imputation for $G(\Delta)$. We will now show that $\mathbf{p}^*$ is in the core of $G(\Delta)$, and therefore $G(\Delta)$ has a nonempty core. Indeed, consider any coalition $C \subset I$ such that $v(C) = 1$. We have $w(C) \ge q$. Moreover, as A is a "no"-instance of PARTITION, there is no coalition $C \subset I$ whose weight is exactly q, so we have $w(C) \ge q + 1 = K + 1$. Thus we have $p^*(C) = \frac{w(C)}{K+1} \ge 1$. Hence, the agents in C have no rational incentive to deviate from $\mathbf{p}^*$ and therefore $\mathbf{p}^* \in \mathrm{core}(G(\Delta))$.

We can use the same construction to show that SUPER-IMPUTATION-STABILITY-WVG is coNP-hard. Indeed, consider G, $\Delta = \frac{K-1}{K+1}$, and $\mathbf{p}^*$ defined above. It follows from our proof that $\mathbf{p}^*$ is in the core of $G(\Delta)$ if and only if A is a "no"-instance of PARTITION. Moreover, SUPER-IMPUTATION-STABILITY-WVG is clearly in coNP: to verify that a given super-imputation $\mathbf{p}$ is unstable, it suffices to guess a coalition C and verify that it is winning, i.e., $w(C) \ge q$, but is paid less than one under $\mathbf{p}$. □

Theorem 8 SUPER-IMPUTATION-STABILITY-WVG and CoS-WVG are in P when the agents' weights are polynomially bounded (or given in unary).

Proof. As argued in Section 3.2, it suffices to show that given a WVG $G = [\mathbf{w}; q]$ and a super-imputation $\mathbf{p}$ for G, we can compute $d(G, \mathbf{p})$ in time $\mathrm{poly}(n, w_{\max}, |\mathbf{p}|)$, where $|\mathbf{p}|$ denotes the number of bits in the binary representation of $\mathbf{p}$.

For any i, $1 \le i \le n$, and any w, $1 \le w \le w(I)$, let

$$X_{i,w} = \min\{p(C) \mid C \subseteq \{1, \dots, i\}, w(C) = w\}.$$

We can compute the quantities $X_{i,w}$ inductively as follows. For $i = 1$, we have $X_{i,w} = p_1$ if $w = w_1$, and $X_{i,w} = +\infty$ otherwise. Now, suppose that we have computed $X_{i',w}$ for each i', $1 \le i' \le i$. We can then compute $X_{i+1,w}$ as $X_{i+1,w} = \min\{X_{i,w}, p_i + X_{i,w-w_i}\}$. Observe that $p^* = \min\{X_{n,w} \mid w \ge q\}$ is the minimal payment that a winning coalition in G can receive under $\mathbf{p}$. As $p_i \ge 0$ for all i, $1 \le i \le n$, we have $d(G, \mathbf{p}) = 1 - p^*$.

Clearly, the running time of this algorithm is polynomial in n, $w_{\max}$ and $|\mathbf{p}|$. Observe that one can construct a similar algorithm that runs in polynomial time even if the weights are large, as long as all entries of $\mathbf{p}$ can take polynomially many values. □

C Proofs for Section 4.2

Theorem 9 There exists an algorithm $\mathcal{A}(G,\varepsilon)$ that, given a WVG $G = [\mathbf{w}; q]$ in which the weights of all players are nonnegative integers given in binary and a parameter $\varepsilon > 0$, outputs a value Δ satisfying $CoS(G) \le \Delta \le (1+\varepsilon)CoS(G)$ and runs in time $\mathrm{poly}(n, \log w_{\max}, 1/\varepsilon)$. That is, there exists a fully polynomial-time approximation scheme (FPTAS) for $CoS(G)$.

Proof. We start by proving a simple lemma that will be useful for the analysis of our algorithm.

Lemma 1. *For any WVG G such that $CoS(G) \neq 0$, we have $CoS(G) \geq 1/n$.*

Proof. Consider a weighted voting game G that does not have a veto player and hence $CoS(G) \neq 0$. Suppose for the sake of contradiction that $CoS(G) = \Delta < 1/n$, that is, the game $G(\Delta)$ has a nonempty core. Let $\mathbf{p} = (p_1, \dots, p_n)$ be an imputation in the core of $G(\Delta)$. As we have $v'(I) = \Delta + 1 > 1$, there must be at least one player i such that $p_i > 1/n$. Hence, $p(I \setminus \{i\}) < 1 + \Delta - 1/n < 1$. Therefore the coalition $I \setminus \{i\}$ satisfies $v(I \setminus \{i\}) = 1$ (since i is not a veto player), $p(I \setminus \{i\}) < 1$, and hence $\mathbf{p}$ is not stable, a contradiction. □

Our proof is inspired by the FPTAS for the value of the least core of WVGs [8].

We will first describe an additive fully polynomial-time approximation scheme for $CoS(G)$, i.e., an algorithm $\mathcal{A}'(G,\varepsilon)$ that, given a WVG $G = [w_1, \dots, w_n; q]$ and $\varepsilon > 0$, can compute a value Δ satisfying $CoS(G) \le \Delta \le CoS(G) + \varepsilon$ and runs in time $\mathrm{poly}(n, \log w_{\max}, 1/\varepsilon)$. We will then show how to convert it into an FPTAS using Lemma 1.

Set $X = 2\lceil 1/\varepsilon \rceil$, and let $\varepsilon' = 1/X$. We have $\varepsilon/4 \le \varepsilon' \le \varepsilon/2$.

Consider the linear program $\mathcal{LP}^*$ given in Section 3. Instead of solving $\mathcal{LP}^*$ directly, we consider a family of linear feasibility programs (LFP) $(\mathcal{L}_i)_{i=1,\dots,nX}$, where the kth LFP $\mathcal{L}_k$ is given by

$$\begin{aligned} p_i &\geq 0 \text{ for } i = 1, \dots, n, \\ p_1 + \dots + p_n &\le 1 + \varepsilon' k, \\ \sum_{i \in C} p_i &\ge 1 \text{ for all } C \subseteq N \text{ such that } \sum_{i \in C} w_i \ge q. \end{aligned}$$

As $\varepsilon' n X = n$, it follows that at least one of these LFPs has a feasible solution. Now, let k^* be the smallest value of k for which $\mathcal{L}_k$ has a feasible solution. We have $\varepsilon'(k^* - 1) < CoS(G) \le \varepsilon' k^*$, or, equivalently, $CoS(G) \le \varepsilon' k^* \le CoS(G) + \varepsilon'$. Hence, by computing k^* we can obtain an additive ε'-approximation to $CoS(G)$. Now, while it is not clear if we could find k^* in polynomial time, we will now show how to find a value k that is guaranteed to be in the set $\{k^*, k^*+1\}$.

It is natural to approach this problem by trying to successively solve $\mathcal{L}_1, \dots, \mathcal{L}_{nX}$. However, just as the linear program $\mathcal{LP}^*$, the LFP $\mathcal{L}_k$ has exponentially many constraints (one for each winning coalition of G). Moreover, an implementation of the separation oracle for $\mathcal{L}_k$ would involve solving KNAPSACK, which is an NP-hard problem when weights are given in binary. Hence, we will now take a somewhat different approach. Namely, we will show how to design an algorithm $\mathcal{S}$ that, given a candidate solution $(p_1, \dots, p_n)$ for $\mathcal{L}_k$, either outputs a constraint that is violated by this solution or finds a feasible solution for $\mathcal{L}_{k+1}$. The running time of $\mathcal{S}(p_1, \dots, p_n)$ is $\mathrm{poly}(n, \log w_{\max}, 1/\varepsilon)$.

The algorithm $\mathcal{S}$ first checks if the candidate solution $(p_1, \dots, p_n)$ satisfies the first $n+1$ constraints of the LFP. If no violated constraint is discovered at this step, it rounds up the payoffs by setting $p'_i = \min\{\frac{\varepsilon' t}{n} \mid t \in \mathbb{N}, \frac{\varepsilon' t}{n} \geq p_i\}$ for each i, $1 \leq i \leq n$. Note that for each i, $1 \leq i \leq n$, we have $p_i \leq p'_i \leq p_i + \frac{\varepsilon'}{n}$, and the rounded payoff p'_i can be represented as $p'_i = \frac{\varepsilon'}{n} t_i$, where $t_i \in \{0, \dots, nX\}$. We can now use a variant of the dynamic programming algorithm used in the proof of Theorem 8 to decide whether there is a subset of agents C that satisfies $\sum_{i \in C} w_i \geq q$, $\sum_{i \in C} p'_i < 1$ (see the remark in the end of that proof). If there is such a subset, the rounded vector $(p'_1, \dots, p'_n)$ violates the constraint that corresponds to C, and hence the original vector $(p_1, \dots, p_n)$, which satisfies $p_i \leq p'_i$ for all $i \in I$, violates it, too. Hence, $\mathcal{S}$ outputs the corresponding constraint and stops. Otherwise, it follows that $(p'_1, \dots, p'_n)$ satisfies all constraints of $\mathcal{L}_k$ that correspond to the winning coalitions of G. Moreover, we have

$$\sum_{i=1}^{n} p'_i \leq \sum_{i=1}^{n} p_i + n\frac{\varepsilon'}{n} \leq 1 + \varepsilon' k + \varepsilon'.$$

Hence, $(p'_1, \dots, p'_n)$ is a feasible solution for $\mathcal{L}_{k+1}$, so $\mathcal{S}$ outputs it and stops.

We are now ready to describe our algorithm $\mathcal{A}'$. It tries to solve $\mathcal{L}_1, \mathcal{L}_2, \dots$ (in this order). To solve $\mathcal{L}_k$, it runs the ellipsoid algorithm on its input. Whenever the ellipsoid algorithm makes a call to the separation oracle, $\mathcal{A}'$ passes this request to $\mathcal{S}$, which either identifies a violated constraint, in which case $\mathcal{A}'$ continues simulating the ellipsoid algorithm, or outputs a feasible solution for $\mathcal{L}_{k+1}$, in which case $\mathcal{A}'$ stops and outputs $\varepsilon'(k+1)$. If the ellipsoid algorithm terminates and decides that the current LFP does not have a feasible solution, $\mathcal{A}'$ proceeds to the next LFP in its list. If the ellipsoid algorithm outputs a feasible solution for $\mathcal{L}_k$, $\mathcal{A}$ outputs $\varepsilon' k$.

Recall that we denote by k^* the smallest value of k for which $\mathcal{L}_k$ has a feasible solution. Clearly, $\mathcal{A}$ will correctly report that neither of $\mathcal{L}_1, \dots, \mathcal{L}_{k^*-2}$ has a feasible solution. When solving $\mathcal{L}_{k^*-1}$, it will either solve it correctly (i.e., report that it has no feasible solutions) and move on to $\mathcal{L}_{k^*}$, or discover a feasible solution for $\mathcal{L}_{k^*}$. In the former case, $\mathcal{A}'$ will either solve $\mathcal{L}_{k^*}$ correctly, i.e., find a feasible solution, or discover a feasible solution to $\mathcal{L}_{k^*+1}$. In either case, the output $\varepsilon' k$ of our algorithm satisfies $k \in \{k^*, k^*+1\}$.

We have shown that $CoS(G) \leq \varepsilon' k^* \leq CoS(G) + \varepsilon'$. Consequently, we have $CoS(G) \leq \varepsilon' k \leq \varepsilon'(k^*+1) \leq CoS(G) + 2\varepsilon' \leq CoS(G) + \varepsilon$. This proves that $\mathcal{A}'$ is an additive fully polynomial-time approximation scheme for the cost of stability.

We will now show how to convert $\mathcal{A}'$ into an FPTAS $\mathcal{A}$. Our algorithm $\mathcal{A}$ is given a game $G = [\mathbf{w}; q]$ and a parameter ε. It first tests if $CoS(G) = 0$ (equivalently, if G has a nonempty core). By Theorem 1, this can be done by checking if G has a veto player, i.e., whether $w(I \setminus \{i\}) < q$ for some i, $1 \leq i \leq n$.

If $CoS(G) \neq 0$, $\mathcal{A}$ runs $\mathcal{A}'$ on input $(G, \varepsilon/n)$. Let Δ be the output of $\mathcal{A}'(G, \varepsilon/n)$; we have $CoS(G) \leq \Delta \leq CoS(G) + \varepsilon/n$. On the other hand, by Lemma 1 we have

$CoS(G) \geq 1/n$, and therefore

$$CoS(G) + \varepsilon/n \leq CoS(G) + \varepsilon CoS(G) = (1 + \varepsilon)CoS(G).$$

Hence Δ satisfies $CoS(G) \leq \Delta \leq (1 + \varepsilon)CoS(G)$, as required. □

Theorem 10 For any weighted voting game $G = (I, v)$ with $CoS(G) = \Delta$ and any $\mathbf{p} \in \mathrm{core}(G(\Delta))$, we have $p^*(I) \leq 2\mathbf{p}(I)$.

Proof. Set $\Delta = CoS(G)$ and fix a super-imputation $\mathbf{p}$ in the core of $G(\Delta)$. Let $I' = \{i \mid w_i \geq q\}$ and set $k = |I'|$. Clearly, if $i \in I'$, for any stable super-imputation $\mathbf{p}'$ we have $p'_i \geq 1 = p^*_i$. On the other hand, it is clear that paying any agent more than 1 is suboptimal, so $p_i = 1$ for any $i \in I'$.

Sort all agents in $I \backslash I'$ by decreasing weights, and partition them into sets $C_1, \dots, C_m$ in the following way:

– Set $j = 0$;
– While there are unallocated agents:
  – Set $j = j + 1$;
  – Add agents to C_j until $w(C_j) \geq q$
    or until there are no more agents;
– Set $m = j$;
– If $w(C_j) \geq q$, set $m = j + 1$ and $C_m = \emptyset$.

Note that this procedure guarantees that $w(C_m) < q$, i.e., the last coalition C_m loses. In particular, if $m = 1$ then $w(C_1) < q$. Since $w(I) \geq q$, this means that $k \geq 1$ and $C_1 = I \setminus I'$. In this case, we have

$$p(I) \geq k, \quad p^*(I) = k + \sum_{i \in C_1} \frac{w_i}{q} < k + \frac{q}{q} = k + 1,$$

and hence $p^*(I)/p(I) < (k+1)/k \leq 2$. Therefore, throughout the rest of the proof we can assume $m > 1$.

Set $j' = \arg\max_{j \leq m} w(C_j)$, that is, j' is the index of a maximum-weight coalition among $C_1, \dots, C_m$. Observe that since $w(C_1) \geq q$ and $w(C_m) < q$, we have $j' \neq m$. To finish the proof, we consider two cases and show that in each of them $p^*(I) \leq 2p(I)$.

Case 1: $w(C_{j'}) + w(C_m) \leq 2q$. For each $j \leq m - 1$, we have $w(C_j) \geq q$, and therefore $p(C_j) \geq 1$. Thus, we have

$$p(I) \geq k + \sum_{j \neq m} p(C_j) = k + m - 1.$$

On the other hand, we have $w(C_j) \leq 2q$ for all j, $1 \leq j \leq m$, so

$$\begin{aligned} p^*(I) &= p^*(I') + \sum_{j \neq j', m} p^*(C_j) + p^*(C_{j'}) + p^*(C_m) \\ &\leq k + \sum_{j \neq j', m} \frac{w(C_j)}{q} + \frac{w(C_{j'}) + w(C_m)}{q} \\ &\leq k + 2(m-2) + 2 \leq 2(k + m - 1) \leq 2p(I). \end{aligned}$$

Case 2: $w(C_{j'}) + w(C_m) > 2q$. We begin by computing $p^*(I)$, as it may be slightly larger in this case:

$$p^*(I) = k + \sum_{j \neq m} \frac{w(C_j)}{q} + \frac{w(C_m)}{q}$$
$$\leq k + \frac{(m-1)2q + q}{q} = k + 2m - 1.$$

Fortunately, we can provide a better lower bound for $p(I)$. Let A_1 be the set that contains the last player in $C_{j'}$ only, and set $A_2 = C_{j'} \setminus A_1$ and $A_3 = C_m$. We have $w(A_1) < q$, since A_1 has just one agent, and we have already removed all agents whose weight is at least q. Furthermore, we have $w(A_2) < q$, since we move on to the next set as soon as a total weight of at least q is reached in the current set. On the other hand, we have $A = A_1 \cup A_2 \cup A_3 = C_{j'} \cup C_m$. As $w(C_{j'}) + w(C_m) > 2q$, we have $w(A_1) + w(A_3) = w(A) - w(A_2) \geq 2q - q = q$ and $w(A_2) + w(A_3) = w(A) - w(A_1) \geq 2q - q = q$.
Therefore, we have $p(A_1 \cup A_2) \geq 1$, $p(A_1 \cup A_3) \geq 1$, $p(A_2 \cup A_3) \geq 1$, and hence $p(A_1 \cup A_2 \cup A_3) \geq 3/2$. Thus, we have

$$p(I) = \sum_{i \in I'} p_i + \sum_{j \neq j', m} p(C_j) + p(C_{j'}) + p(C_m)$$
$$\geq k + (m-2) + p(C_{j'} \cup C_m)$$
$$= k + m - 2 + p(A_1 \cup A_2 \cup A_3)$$
$$\geq k + m - 2 + \frac{3}{2} = \frac{1}{2}(2k + 2m - 1) \geq \frac{1}{2}p^*(I).$$

□

To see that the analysis presented above is tight, consider the game $[1 - \frac{\epsilon}{3}, 1 - \frac{\epsilon}{3}; 1]$ for any fixed $\varepsilon > 0$. We have $p^*(I) = 2 - \frac{2\varepsilon}{3}$. On the other hand, this game has a nonempty core, so we have $p(I) = 1$, and hence $p^*(I) > (2 - \epsilon)p(I)$.

D Examples for Section 5.3

Example 2. Consider the game $G = [1, 1, 1; 2]$ and the coalition $C = \{1, 2\}$. If we were to stabilize C by paying its members only, we would have to ensure that each of them receives a payment of 1, resulting in an external payment of 1: if, e.g., player 1 receives $p_1 < 1$, player 3 could offer him to form the coalition $\{1, 3\}$ and distribute the payoffs as $p'_1 = p_1 + \frac{1-p_1}{2} > p_1$, $p'_3 = \frac{1-p_1}{2} > 0 = p_3$. On the other hand, it is not hard to see that the payoff vector $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ ensures that no group of players wants to deviate from $(\{1, 2\}, \{3\})$, i.e., the central authority can stabilize C by spending $\frac{1}{2}$ only as long as it is willing to pay the players outside of C. Thus, the cheapest way to stabilize a particular coalition may involve paying agents who do not belong to that coalition.

Example 3. Consider the weighted voting game $G = [8, 8, 9, 9, 1; 10]$ and a coalition $C = \{1, 2\}$. It is not hard to check that G has an empty CS-core and therefore

$CoS_{CS}(G) > 0$. However, no player in C has an incentive to deviate from the coalition structure $CS = (\{1,2\},\{3,4\},\{5\})$ with the payoff vector $\mathbf{p} = (.5, .5, .5, .5, 0)$. That is, if the central authority is only interested in stabilizing C, it can achieve this goal without spending any money. However, from a long-term perspective this approach may be dangerous. Indeed, consider the coalition $\{4,5\}$ that has an incentive to deviate from $(CS, \mathbf{p})$. If this deviation happens, player 3 is left on her own, and will be happy to form a coalition with player 1 in which 1 gets, e.g., $.9$ and 3 gets $.1$. Clearly, this proposition would be attractive to player 1 as well, which would cause the coalition C to fall apart. Thus, stabilizing a given coalition may be strictly cheaper than stabilizing *any* of the coalition structures that contain it.