12

Teams of Selfish Agents: An Introduction to Coalitional Game Theory

In Chapters 1 and 2 we looked at how teams of cooperative agents can accomplish more together than they can achieve in isolation. Then, in Chapter 3 and many of the chapters that followed, we looked at how self-interested agents make individual choices. In this chapter we interpolate between these two extremes, asking how self-interested agents can combine to form effective teams. As the title of the chapter suggests, this chapter is essentially a crash course in *coalitional game theory*, also known as *cooperative game theory*. As was mentioned at the beginning of Chapter 3, when we introduced noncooperative game theory, the term "cooperative" can be misleading. It does not mean that, as in Chapters 1 and 2, each agent is agreeable and will follow arbitrary instructions. Rather, it means that the basic modeling unit is the group rather than the individual agent. More precisely, in coalitional game theory we still model the individual preference of agents, but not their possible actions. Instead, we have a coarser model of the capabilities of different groups.

We proceed as follows. First, we define the most widely studied model of coalitional games, give examples of situations that can be modeled in this way, and discuss a series of refinements to the model. Then we consider how such games can be analyzed. The main solution concepts we discuss here are the *Shapley value*, the *core*, and the *nucleolus*. Finally, we consider compact representations of coalitional games and their computational implications. We conclude by surveying further directions that have been explored in the literature.

12.1 Coalitional games with transferable utility

In coalitional game theory our focus is on what groups of agents, rather than individual agents, can achieve. Given a set of agents, a coalitional game defines how well each group (or *coalition*) of agents can do for itself. We are not concerned with how the agents make individual choices within a coalition, how they coordinate, or any other such detail; we simply take the payoff[1] to a coalition as given.

1. Alternatively, one might assign *costs* instead of payoffs to coalitions. Throughout this chapter, we will focus on the case of payoffs; the concepts defined herein can be extended analogously to the case of costs.

12.1.1 *Definition*

For most of this chapter we will make the *transferable utility assumption*—that the payoffs to a coalition may be freely redistributed among its members. This assumption is satisfied whenever there is a universal *currency* that is used for exchange in the system. When this assumption holds, each coalition can be assigned a single value as its payoff.

coalitional game with transferable utility

Definition 12.1.1 (Coalitional game with transferable utility) *A* coalitional game with transferable utility *is a pair* (N, v)*, where:*

- N *is a finite*[2] *set of players, indexed by* i*; and*
- $v : 2^N \mapsto \mathbb{R}$ *associates with each coalition* $S \subseteq N$ *a real-valued payoff* $v(S)$ *that the coalition's members can distribute among themselves. The function* v *is also called the* characteristic function*, and a coalition's payoff is also called its* worth*. We assume that* $v(\emptyset) = 0$.

characteristic function

Most of the time, coalitional game theory is used to answer two fundamental questions:

1. Which coalition will form?
2. How should that coalition divide its payoff among its members?

It turns out that the answer to (1) is often "the grand coalition"—the name given to the coalition of all the agents in N—though this answer can depend on having made the right choice about (2). Before we go any further in answering these questions, however, we provide several examples of coalitional games to help motivate the model.

12.1.2 *Examples*

Coalitional games can be used to describe problems arising in a wide variety of different contexts. To emphasize the relevance of coalitional game theory to other topics covered in this book, we give examples motivated by problems from social choice (Chapter 9), mechanism design (Chapter 10), and auctions (Chapter 11). We will also use these examples to highlight some important classes of coalitional games in Section 12.1.3. We note that here we do *not* describe how payments could or should be divided among the agents; we defer such discussion to Section 12.2. Our first example draws on social choice, in the vein of the discussion in Section 9.3.

Example 12.1.2 (Voting game) *A parliament is made up of four political parties,* A*,* B*,* C*, and* D*, which have 45, 25, 15, and 15 representatives, respectively. They are to vote on whether to pass a \$100 million spending bill and how much of this amount should be controlled by each of the parties. A majority vote, that is, a minimum of 51 votes, is required in order to pass any legislation, and if the bill does not pass then every party gets zero to spend.*

2. Observe that we consider only finite coalitional games. The infinite case is also considered in the literature; many but not all of the results from this chapter also hold in this case.

More generally, in a voting game, there is a set of agents N and a set of coalitions $\mathcal{W} \subseteq 2^N$ that are winning *coalitions, that is, coalitions that are sufficient for the passage of the bill if all its members choose to do so. To each coalition $S \in \mathcal{W}$, we assign $v(S) = 1$, and to the others we assign $v(S) = 0$.*

Many voting games that arise in practice can be represented succinctly as *weighted majority* or *weighted voting* games. We discuss these representations in Section 12.3.1.

Our next example concerns sharing the cost of a public good, along the lines of the road-building referendum example given in Section 10.4.

Example 12.1.3 (Airport game) *A number of cities need airport capacity. If a new regional airport is built the cities will have to share its cost, which will depend on the largest aircraft that the runway can accommodate. Otherwise each city will have to build its own airport.*

This situation can be modeled as a coalitional game (N, v), where N is the set of cities, and $v(S)$ is the sum of the costs of building runways for each city in S minus the cost of the largest runway required by any city in S.

Next, we consider a situation in which agents need to get *connected* to the public good in order to enjoy its benefit. One such setting is the problem of multicast cost sharing that we previously examined in Section 10.6.3.

Example 12.1.4 (Minimum spanning tree game) *A group of customers must be connected to a critical service provided by some central facility, such as a power plant or an emergency switchboard. In order to be served, a customer must either be directly connected to the facility or be connected to some other connected customer. Let us model the customers and the facility as nodes on a graph, and the possible connections as edges with associated costs.*

This situation can be modeled as a coalitional game (N, v). N is the set of customers, and $v(S)$ is the cost of connecting all customers in S directly to the facility minus the cost of the minimum spanning tree that spans both the customers in S and the facility.

Finally, we consider a coalitional game in an auction setting.

Example 12.1.5 (Auction game) *Consider an auction mechanism in which the allocation rule is efficient (i.e., social welfare maximizing). Our analysis in Chapter 11 treated the set of participating agents as given. We might instead want to determine whether the seller would prefer to exclude some interested agents to obtain higher payments. (Indeed, it turns out that this can occur; see Section 12.2.2.) To find out, we can model the auction as a coalitional game.*

Let N_B be the set of bidders, and let 0 be the seller. The agents in the coalitional game are $N = N_B \cup \{0\}$. Choosing a coalition means running the auction with the appropriate set of agents. The value of a coalition S is the sum of agents' utilities for the efficient allocation when the set of participating agents is restricted to S.[3] *A coalition that does not include the seller has value 0, because in this case a trade cannot occur.*

3. The value of a coalition can be understood as the sum of the agents' utilities for the auction's outcome (their valuations for bundles received minus payments) plus the seller's utility (the sum of payments

12.1.3 *Classes of coalitional games*

In this section we will define a few important classes of coalitional games, which have interesting applications as well as useful formal properties. We start with the notion of superadditivity, a property often assumed for coalitional games.

superadditive game

Definition 12.1.6 (Superadditive game) *A game $G = (N, v)$ is* superadditive *if for all $S, T \subset N$, if $S \cap T = \emptyset$, then $v(S \cup T) \geq v(S) + v(T)$.*

Superadditivity is justified when coalitions can always work without interfering with one another; hence, the value of two coalitions will be no less than the sum of their individual values. Note that superadditivity implies that the value of the entire set of players (the "grand coalition") is no less than the sum of the value of any nonoverlapping set of coalitions. In other words, the grand coalition has the highest payoff among all coalitional structures. All of the examples we gave earlier describe superadditive games.

Taking noninterference across coalitions to the extreme, when coalitions can never affect one another, either positively or negatively, then we have *additive* (or *inessential*) games.

additive game

Definition 12.1.7 (Additive game) *A game $G = (N, v)$ is* additive *(or* inessential*) if for all $S, T \subset N$, if $S \cap T = \emptyset$, then $v(S \cup T) = v(S) + v(T)$.*

A related class of games is that of constant-sum games.

constant-sum game

Definition 12.1.8 (Constant-sum game) *A game $G = (N, v)$ is* constant sum *if for all $S \subset N$, $v(S) + v(N \setminus S) = v(N)$.*

Note that every additive game is necessarily constant sum, but not vice versa. As in noncooperative game theory, the most commonly studied constant-sum games are *zero-sum games*.

zero-sum game

An important subclass of superadditive games are convex games.

convex game

Definition 12.1.9 (Convex game) *A game $G = (N, v)$ is* convex *if for all $S, T \subset N$, $v(S \cup T) \geq v(S) + v(T) - v(S \cap T)$.*

Clearly, convexity is a stronger condition than superadditivity. While convex games may therefore appear to be a very specialized class of coalitional games, these games are actually not so rare in practice. For example, the Airport game as described in Example 12.1.3 is convex. Convex games have a number of useful properties, as we will discuss in the next section.

Finally, we present a class of coalitional games with restrictions on the values that payoffs are allowed to take.

simple game

Definition 12.1.10 (Simple game) *A game $G = (N, v)$ is* simple *if for all $S \subset N$, $v(S) \in \{0, 1\}$.*

received). Note that because payments are transfers between members of the coalition they cancel out and do not affect the coalition's value.

Superadditive $\supset$ Convex $\supset$ Additive

Constant sum $\supset$ Additive

Constant sum $\supset$ Proper simple

Simple $\supset$ Proper simple

Figure 12.1 A hierarchy of coalitional game classes; $X \supset Y$ means that class X is a superclass of class Y.

Simple games are useful for modeling voting situations, such as those described in Example 12.1.2. In simple games we often add the requirement that if a coalition wins, then all larger sets are also winning coalitions (i.e., if $v(S) = 1$, then for all $T \supset S$, $v(T) = 1$). This condition might seem to imply superadditivity, but it does not quite. For example, the condition is met by a voting game in which only 50% of the votes are required to pass a bill, but such a game is not superadditive. Consider two disjoint winning coalitions S and T; when they join to form the coalition $S \cup T$ they do not achieve at least the sum of the values that they achieve separately as superadditivity requires.

proper simple game

When simple games are also constant sum, they are called *proper simple games*. In this case, if S is a winning coalition, then $N \setminus S$ is a losing coalition.

Figure 12.1 graphically depicts the relationship between the different classes of games that we have discussed in this section.

12.2 Analyzing coalitional games

The central question in coalitional game theory is the division of the payoff to the grand coalition among the agents. This focus on the grand coalition is justified in two ways. First, since many of the most widely studied games are superadditive, the grand coalition will be the coalition that achieves the highest payoff over all coalitional structures, and hence we can expect it to form. Second, there may be no choice for the agents but to form the grand coalition; for example, public projects are often legally bound to include all participants.

If it is easy to decide to concentrate on the grand coalition, however, it is less easy to decide how this coalition should divide its payoffs. In this section we explore a variety of solution concepts that propose different ways of performing this division.

Before presenting the solution concepts, it is helpful to introduce some terminology. First, let $\psi : \mathbb{N} \times \mathbb{R}^{2^{|N|}} \mapsto \mathbb{R}^{|N|}$ denote a mapping from a coalitional game (that is, a set of agents N and a value function v) to a vector of $|N|$ real values, and let $\psi_i(N, v)$ denote the i^{th} such real value. Denote such a vector of $|N|$ real values as $x \in \mathbb{R}^{|N|}$. Each x_i denotes the share of the grand coalition's payoff that agent $i \in N$ receives. When the coalitional game (N, v) is understood from context, we write x as a shorthand for $\psi(N, v)$.

Now we can give some basic definitions about payoff division. Each has an analogue in the properties we required of quasilinear mechanisms in Section 10.3.2, which we name as we come to each definition.

feasible payoff

Definition 12.2.1 (Feasible payoff) *Given a coalitional game (N, v), the* feasible payoff set *is defined as* $\{x \in \mathbb{R}^N \mid \sum_{i \in N} x_i \leq v(N)\}$.

In other words, the feasible payoff set contains all payoff vectors that do not distribute more than the worth of the grand coalition. We can view this as requiring the payoffs to be *weakly budget balanced*.

pre-imputation

Definition 12.2.2 (Pre-imputation) *Given a coalitional game (N, v), the* pre-imputation set, *denoted $\mathcal{P}$, is defined as* $\{x \in \mathbb{R}^N \mid \sum_{i \in N} x_i = v(N)\}$.

We can view the pre-imputation set as the set of feasible payoffs that are *efficient*, that is, they distribute the entire worth of the grand coalition. Looked at another way, the pre-imputation set is the set of feasible payoffs that are *strictly budget balanced*. (In this setting these two concepts are equivalent; do you see why?)

imputation

Definition 12.2.3 (Imputation) *Given a coalitional game (N, v), the* imputation set, *$\mathcal{I}$, is defined as* $\{x \in \mathcal{P} \mid \forall i \in N, x_i \geq v(i)\}$.

Imputations are payoff vectors that are not only efficient but *individually rational*. That is, each agent is guaranteed a payoff of at least the amount that he could achieve by forming a singleton coalition.

Now we are ready to delve into different solution concepts for coalitional games.

12.2.1 *The Shapley value*

Perhaps the most straightforward answer to the question of how payoffs should be divided is that the division should be *fair*. As we did in Section 9.4.1, let us begin by laying down axioms that describe what fairness means in our context.

interchangeable agents

First, say that agents i and j are *interchangeable* if they always contribute the same amount to every coalition of the other agents. That is, for all S that contains neither i nor j, $v(S \cup \{i\}) = v(S \cup \{j\})$. The *symmetry* axiom states that such agents should receive the same payments.

Axiom 12.2.4 (Symmetry) *For any v, if i and j are interchangeable then $\psi_i(N, v) = \psi_j(N, v)$.*

dummy player

Second, say that an agent i is a *dummy player* if the amount that i contributes to any coalition is exactly the amount that i is able to achieve alone. That is, for all S such that $i \notin S$, $v(S \cup \{i\}) - v(S) = v(\{i\})$. The *dummy player* axiom states that dummy players should receive a payment equal to exactly the amount that they achieve on their own.

Axiom 12.2.5 (Dummy player) *For any v, if i is a dummy player then $\psi_i(N, v) = v(\{i\})$.*

Finally, consider two different coalitional game theory problems, defined by two different characteristic functions v_1 and v_2, involving the same set of agents. The *additivity* axiom states that if we re-model the setting as a single game in which each coalition S achieves a payoff of $v_1(S) + v_2(S)$, the agents' payments in each coalition should be the sum of the payments they would have achieved for that coalition under the two separate games.

Axiom 12.2.6 (Additivity) *For any two v_1 and v_2, we have for any player i that $\psi_i(N, v_1 + v_2) = \psi_i(N, v_1) + \psi_i(N, v_2)$, where the game $(N, v_1 + v_2)$ is defined by $(v_1 + v_2)(S) = v_1(S) + v_2(S)$ for every coalition S.*

If we accept these three axioms, we are led to a strong result: there is always exactly one pre-imputation that satisfies them.

Theorem 12.2.7 *Given a coalitional game (N, v), there is a unique pre-imputation $\phi(N, v) = \phi(N, v)$ that satisfies the Symmetry, Dummy player, Additivity axioms.*

Note that our requirement that $\phi(N, v)$ be a pre-imputation implies that the payoff division be feasible and efficient (or strictly budget balanced). Because we do not insist on an imputation, individual rationality is not required to hold, though of course it still may.

What is this unique payoff division $\phi(N, v)$? It is called the *Shapley value*, and it is defined as follows.

Shapley value

Definition 12.2.8 (Shapley value) *Given a coalitional game (N, v), the* Shapley value *of player i is given by*

$$\phi_i(N, v) = \frac{1}{N!} \sum_{S \subseteq N \setminus \{i\}} |S|!(|N| - |S| - 1)!\Big[v(S \cup \{i\}) - v(S)\Big].$$

This expression can be viewed as capturing the "average marginal contribution" of agent i, where we average over all the different sequences according to which the grand coalition could be built up from the empty coalition. More specifically, imagine that the coalition is assembled by starting with the empty set and adding one agent at a time, with the agent to be added chosen uniformly at random. Within any such sequence of additions, look at agent i's marginal contribution at the time he is added. If he is added to the set S, his contribution is $[v(S \cup \{i\}) - v(S)]$. Now multiply this quantity by the $|S|!$ different ways the set S could have been formed prior to agent i's addition and by the $(|N| - |S| - 1)!$ different ways the remaining agents could be added afterward. Finally, sum over all possible sets S and obtain an average by dividing by $|N|!$, the number of possible orderings of all the agents.

For a concrete example of the Shapley value in action, consider the voting game given in Example 12.1.2. Recall that the four political parties A, B, C, and D have 45, 25, 15, and 15 representatives, respectively, and a simple majority (51 votes) is required to pass the \$100 million spending bill. If we want to analyze how much money it is fair for each party to demand, we can calculate the Shapley values of the game. Note that every coalition with 51 or more members has a value of \$100 million,[4] and others have \$0. In this game, therefore, the parties

4. Notice that for these calculations we scale the value function to 100 for winning coalitions and 0 for losing coalitions in order to make it align more tightly with our example.

B, C, and D are interchangeable, since they add the same value to any coalition. (They add \$100 million to the coalitions $\{B, C\}$, $\{C, D\}$, $\{B, D\}$ that do not include them already and to $\{A\}$; they add \$0 to all other coalitions.) The Shapley value of A is given by:

$$\begin{aligned}\phi_A &= \frac{1}{4!}[3!0!(100-100) + 3 \cdot 2!1!(100-0) + 3 \cdot 1!2!(100-0) \\ &\quad + 0!3!(0-0)] \\ &= \frac{1}{24}[0 + 600 + 600 + 0] = \$50 \text{ million.}\end{aligned}$$

The Shapley value for B (and, by symmetry, also for C and D) is given by:

$$\begin{aligned}\phi_B &= \frac{1}{4!}[3!0!(100-100) + 2 \cdot 2!1!(100-100) + 2!1!(100-0) \\ &\quad + 1!2!(100-0) + 2 \cdot 1!2!(0-0) + 0!3!(0-0)] \\ &= \frac{1}{24}[0 + 0 + 200 + 200 + 0 + 0] = \$16.67 \text{ million.}\end{aligned}$$

Thus the Shapley values are (50, 16.67, 16.67, 16.67), which add up to the entire \$100 million.

To continue with an example mentioned earlier, in Section 10.6.3 we discussed the *Shapley mechanism* for sharing the cost of multicast transmissions. Now that we have learned about the Shapley value—what is the connection? It turns out to depend on a probabilistic interpretation of the Shapley value. Suppose that the agents to be served arrive in a random order in the fixed multicast tree, and that each agent is responsible for the cost of the remaining edges needed to be built for him to get connected. The Shapley mechanism charges the agents their expected connection costs in such a model, averaging over all orders chosen uniformly at random.

12.2.2 *The core*

The Shapley value defined a fair way of dividing the grand coalition's payment among its members. However, this analysis ignored questions of stability. We can also ask: would the agents be *willing* to form the grand coalition given the way it will divide payments, or would some of them prefer to form smaller coalitions? Unfortunately, sometimes smaller coalitions can be more attractive for subsets of the agents, even if they lead to lower value overall. Considering the majority voting example, while A does not have a unilateral motivation to vote for a different split, A and B have incentive to defect and divide the \$100 million between themselves (e.g., dividing it (75, 25)).

This leads to the question of what payment divisions would make the agents want to form the grand coalition. The answer is that they would want to do so if and only if the payment profile is drawn from a set called the *core*, defined as follows.

core **Definition 12.2.9 (Core)** *A payoff vector x is in the* core *of a coalitional game (N, v) if and only if*

$$\forall S \subseteq N, \ \sum_{i \in S} x_i \geq v(S).$$

Thus, a payoff is in the core if and only if no sub-coalition has an incentive to break away from the grand coalition and share the payoff it is able to obtain independently. That is, it requires that the sum of payoffs to any group of agents $S \subseteq N$ must be at least as large as the amount that these agents could share among themselves if they formed a coalition on their own. Notice that Definition 12.2.9 implies that payoff vectors in the core must always be imputations: that is, they must always be strictly budget balanced and individually rational.

Since the core provides a concept of stability for coalitional games we can see it as an analog of the Nash equilibrium from noncooperative games. However, it is actually a stronger notion: Nash equilibrium describes stability only with respect to deviation by a single agent. Instead, the core is an analog of the concept of *strong equilibrium* (discussed in Section 10.7.3), which requires stability with respect to deviations by arbitrary coalitions of agents.

How can the core be computed? The answer is conceptually straightforward and is given by the linear feasibility problem[5] that follows.

$$\sum_{i \in S} x_i \geq v(S) \qquad \forall S \subseteq N \tag{12.1}$$

As a notion of stability for coalitional games, the core is appealing. However, the alert reader might have two lingering doubts, arising due to its implicit definition through inequalities:

1. Is the core always nonempty?
2. Is the core always unique?

Unfortunately, the answer to both questions is no. Let us consider again the Parliament example with the four political parties. The set of minimal coalitions that meet the required 51 votes is $\{A, B\}$, $\{A, C\}$, $\{A, D\}$, and $\{B, C, D\}$. We can see that if the sum of the payoffs to parties B, C, and D is less than \$100 million, then this set of agents has incentive to deviate. On the other hand, if B, C, and D get the entire payoff of \$100 million, then A will receive \$0 and will have incentive to form a coalition with whichever of B, C, and D obtained the smallest payoff. Thus, the core is empty for this game.

On the other hand, when the core is nonempty it may not define a unique payoff vector either. Consider changing our example so that instead of a simple majority, an 80% majority is required for the bill to pass. The minimal winning coalitions are now $\{A, B, C\}$ and $\{A, B, D\}$. Any complete distribution of the \$100 million among parties A and B now belongs to the core, since all winning coalitions must have both the support of these two parties.

5. Linear feasibility problems are linear programs with only constraints but no objective function. Linear programs are defined in Appendix B.

These examples call into question the universality of the core as a solution concept for coalitional games. We already saw in the context of noncooperative game theory that solution concepts—notably, the Nash equilibrium—do not yield unique solutions in general. Here we are in an arguably worse situation, in that the solution concept may yield no solution at all.

Can we characterize when a coalitional game has a nonempty core? Fortunately, that at least is possible. To do so, we first need to define a concept known as *balancedness*.

balanced set

Definition 12.2.10 (Balanced weights) *A set of nonnegative weights (over 2^N), λ, is* balanced *if*

$$\forall i \in N, \sum_{S: i \in S} \lambda(S) = 1.$$

Intuitively, the weights on the coalitions involving any given agent i can be interpreted as the conditional probabilities that these coalitions will form, given that i will belong to a coalition.

Theorem 12.2.11 (Bondereva–Shapley) *A coalitional game (N, v) has a nonempty core if and only if for all balanced sets of weights λ,*

$$v(N) \geq \sum_{S \subseteq N} \lambda(S) v(S). \tag{12.2}$$

Proof. Consider the linear feasibility problem used to compute the core, given in Equation (12.1). We can construct the following linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in N} x_i \\ \text{subject to} \quad & \sum_{i \in S} x_i \geq v(S) \qquad && \forall S \subseteq N \end{aligned}$$

Note that when the value of this program is no bigger than $v(N)$, then the payoff vector x is feasible, and belongs to the core. Indeed, the value of the program is equal to $v(N)$ if and only if the core is nonempty. Now consider this linear program's dual.

$$\begin{aligned} \text{maximize} \quad & \sum_{S \subseteq N} \lambda(S) v(S) \\ \text{subject to} \quad & \sum_{S \subseteq N} \lambda(S) = 1 \qquad && \forall i \in N \\ & \lambda(S) \geq 0 && \forall S \subseteq N \end{aligned}$$

Note that the linear constraints in the dual ensure that λ is balanced. By weak duality, the optimal value of the dual is at most the optimal value of the primal, and hence the core is nonempty if and only if the optimal value of the dual is no greater than $v(N)$. ■

While the Bondereva–Shapley theorem completely characterizes when a coalitional game has a nonempty core, it is not always easy or feasible to check that a game satisfies Equation (12.2) for all balanced sets of weights. Luckily, there exist several results that allow us to predict the emptiness or nonemptiness of the core based on a coalitional game's membership in one of the classes we defined in Section 12.1.3.

Theorem 12.2.12 *Every constant-sum game that is not additive has an empty core.*

veto player

We say that a player i is a *veto player* if $v(N \setminus \{i\}) = 0$.

Theorem 12.2.13 *In a simple game the core is empty iff there is no veto player. If there are veto players, the core consists of all payoff vectors in which the nonveto players get zero.*

Theorem 12.2.14 *Every convex game has a nonempty core.*

A final question we consider regards the relationship between the core and the Shapley value. We know that the core may be empty, but if it is not, is the Shapley value guaranteed to lie in the core? The answer in general is no, but the following theorem gives us a sufficient condition for this property to hold. We already know from Theorem 12.2.14 that the core of convex games is nonempty. The following theorem further tells us that for such games the Shapley value belongs to that set.

Theorem 12.2.15 *In every convex game, the Shapley value is in the core.*

We now consider an application of the core to our Auction game (Example 12.1.5). Earlier we asked whether any coalition (consisting of bidders and the seller) could do better than the payoffs they receive when everyone participates in the mechanism. Now that we have defined the core, we can see that the question can be rephrased as asking whether the seller's and the agents' payoffs from the auction are in the core.

First, let us consider the case of single-item, second-price auctions. If the bidders follow their weakly dominant strategy of truthful reporting, it turns out that the payoffs are always in the core. This is because the seller receives revenue equal to or greater than the valuations of all the losing bidders (specifically, equal to the second-highest valuation), and hence cannot entice any of the losing bidders to pay him more.

Now let us consider the case of the VCG mechanism applied to combinatorial auctions. Interestingly, though this mechanism generalizes the second-price auction discussed earlier, it does *not* guarantee payoffs from the core. For example, consider an auction with three bidders and two goods x and y, with the following valuations.

Bidder 1	**Bidder 2**	**Bidder 3**
$v_1(x, y) = 90$	$v_2(x) = v_2(x, y) = 100$	$v_3(y) = v_3(x, y) = 100$
$v_1(x) = v_1(y) = 0$	$v_2(y) = 0$	$v_3(x) = 0$

The efficient allocation awards x to bidder 2 and y to bidder 3. Neither bidder is pivotal, so both pay 0. However, both bidder 1 and the seller would benefit from forming a coalition in which bidder 1 wins the bundle x, y and pays any amount $0 < p_1 < 90$. Thus in a combinatorial auction the VCG payoffs are not guaranteed to belong to the core.

12.2.3 *Refining the core: ϵ-core, least core, and nucleolus*

We now consider some refinements of the core that address its possible nonexistence and nonuniqueness.

We first define a concept analogous to ϵ-equilibrium (defined in Section 3.4.7).

ϵ-core

Definition 12.2.16 (ϵ-core) *A payoff vector x is in the* ϵ-core *of a coalitional game (N, v) if and only if*

$$\forall S \subset N, \ \sum_{i \in S} x_i \geq v(S) - \epsilon. \tag{12.3}$$

One interpretation of the ϵ in Equation (12.3) is that there is an ϵ cost for deviating from the grand coalition. As a result, even if the payoffs to agents in coalition S are less than their worth, $v(S)$, as long as the difference is less than ϵ, the payoff vector is still stable.

Mathematically speaking, there is no requirement that ϵ has to be nonnegative; when ϵ is negative, $-\epsilon$ can be seen as a "bonus" for forming a new coalition. Thus, when ϵ is negative, a payoff vector that is in the ϵ-core is *more* stable than a vector that is only in the core. Note that in Equation (12.3), constraints are quantified only over coalitions that are strict subsets of N. Since payoff vectors are efficient, $\sum_{i \in N} x_i = v(N)$ is always true. Thus, adding a constraint for N is unnecessary: it would be trivially satisfied when ϵ is nonnegative and violated when ϵ is negative, and would shed no light on whether the grand coalition would form.

Note that just like the core, for a given ϵ, the ϵ-core of a game may be empty. On the other hand, it is easy to see that given a game, there always exists some ϵ that is sufficiently large to ensure that the ϵ-core of the game is nonempty. A natural problem, therefore, is to find the smallest ϵ for which the ϵ-core of a game is nonempty. This leads to a solution concept called the *least core*.

least core

Definition 12.2.17 (Least core) *A payoff vector x is in the* least core *of a coalitional game (N, v) if and only if x is the solution to the following linear program.*

$$\begin{aligned} \text{minimize} \quad & \epsilon \\ \text{subject to} \quad & \sum_{i \in S} x_i \geq v(S) - \epsilon \qquad \forall S \subset N \end{aligned}$$

The objective function in the linear program given in Definition 12.2.17 is nonpositive if and only if the core of the game is nonempty. As explained, for sufficiently large ϵ, the constraints in the linear program can always be satisfied; hence, the least core of a game is never empty. Thus the least core can be considered a generalization of the core. On the other hand, the least core does not contain every payoff vector in the core when the core is nonempty; rather, it consists only of payoff vectors that will give all coalitions as little incentive to deviate as possible. In this sense, the least core is also a refinement of the core.

Although it refines the core, the least core does not uniquely determine a payoff vector to a game: it can still return a set of payoff vectors. The intuition is that beyond those coalitions for which the constraints in the linear program are tight (i.e., are realized as equality), there are extra degrees of freedom available for distributing the payoffs to agents in the other coalitions. Based on this intuition, it is easy to construct counterexamples to the uniqueness of the least core. (Can you find one?)

It seems that we could further strengthen the least core by requiring that coalitions whose constraints are slack (i.e., not tight) in the linear program must be as stable as possible. To formalize this idea, let ϵ_1 be the objective value of the linear program, and let $\mathcal{S}_1$ be the set of coalitions corresponding to the set of constraints that are tight in the optimal solution. We now optimize.

$$\begin{array}{lll} \text{minimize} & \epsilon & \\ \text{subject to} & \displaystyle\sum_{i \in S} x_i = v(S) - \epsilon_1 & \forall S \in \mathcal{S}_1 \\ & \displaystyle\sum_{i \in S} x_i \geq v(S) - \epsilon & \forall S \in 2^N \setminus \mathcal{S}_1 \end{array}$$

But what if some constraints remain slack even in the solution to the new optimization problem? We can simply solve yet another optimization problem to make the remaining coalitions as stable as possible. Repeating this procedure, the payoff vector gets progressively tightened. Since at each step, at least one more constraint will be made tight, this process must terminate. Indeed, a careful argument that counts the number of dimensions shows that in fact the process must terminate after at most $|N|$ steps. At the end of this process, we reach a unique payoff vector, known as the *nucleolus*.

nucleolus **Definition 12.2.18 (Nucleolus)** *A payoff vector x is in the* nucleolus *of a coalitional game (N, v) if it is the solution to the series of optimization programs $O_1, O_2, \ldots, O_{|N|}$, where these programs are defined recursively as follows.*

$$(O_1) \qquad \left\{ \begin{array}{lll} \text{minimize} & \epsilon & \\ \text{subject to} & \displaystyle\sum_{i \in S} x_i \geq v(S) - \epsilon & \forall S \subset N \end{array} \right.$$

$$(O_i) \qquad \left\{ \begin{array}{lll} \text{minimize} & \epsilon & \\ \text{subject to} & \displaystyle\sum_{i \in S} x_i = v(S) - \epsilon_0 & \forall S \in \mathcal{S}_1 \\ & \vdots & \\ & \displaystyle\sum_{i \in S} x_i = v(S) - \epsilon_{i-1} & \forall S \in \mathcal{S}_{i-1} \setminus \mathcal{S}_{i-2} \\ & \displaystyle\sum_{i \in S} x_i \geq v(S) - \epsilon & \forall S \in 2^N \setminus \mathcal{S}_{i-1} \end{array} \right.$$

ϵ_{i-1} is the optimal objective value to program O_{i-1} and $\mathcal{S}_{i-1}$ is the set of coalitions for which in the optimal solution to O_{i-1}, the constraints are realized as equalities.[6]

Unlike the core, the ϵ-core, and the least core, the nucleolus possesses the desirable property that it is unique, regardless of the game.

Theorem 12.2.19 *For any coalitional game (N, v), the nucleolus of the game always exists and is unique.*

> **Proof Sketch.** For existence of the nucleolus, one can solve the series of optimization programs as defined in Definition 12.2.18 and end up with some assignment of values to the variables x that corresponds to some payoff vector. These programs are linear programs with finitely many constraints; hence, they can always be solved.
>
> For uniqueness, first observe that the earlier optimization problems can only influence the later ones through the values. Therefore, the set of programs will always yield the same set of solutions $\{\epsilon_1, \ldots, \epsilon_{|N|}\}$. After $|N|$ optimization, we are left with a system of $2^{|N|}$ equations over $|N|$ variables. This system is of rank at most $|N|$, and therefore if a solution exists, it must be unique. ■

The nucleolus has an alternate definition in terms of *excess*, which we define next. This definition is worth understanding because it provides additional intuition about the meaning of the nucleolus.

excess of a coalition

Definition 12.2.20 (Excess of a coalition) *The* excess of a coalition *S in game (N, v) with respect to a payoff vector x, $e(S, x, v)$, is defined as $v(S) - \sum_{i \in S} x_i$, that is, the amount a coalition gain by deviating, as compared to that coalition's payoff as part of the grand coalition.*

Given a coalitional game (N, v) and a payoff vector x, compute the excesses of all coalitions except coalition N and $\emptyset$; we call this $(2^{|N|} - 2)$-dimensional vector the *raw excess vector*. When this vector is sorted in decreasing order of excess, we call it the *sorted excess vector* and denote it as $ev(x, v)$.

Given two payoff vectors x and y, we say the excesses due to x are lexicographically smaller than those due to y, written $x \prec_{e(v)} y$, if for the smallest index such that $ev(x, v)$ and $ev(y, v)$ differ, $ev(x, v) < ev(y, v)$. The nucleolus can then be defined as the payoff vector that is smallest according to the $\prec_{e(v)}$ relation.

Definition 12.2.21 (Nucleolus, alternate definition) *Given a coalitional game (N, v), the* nucleolus *is the payoff vector x such that for all other payoff vectors y, $y \succ_{e(v)} x$, that is, x lexicographically minimizes the excesses of all coalitions except N and $\emptyset$.*

6. We have to be careful and terminate before the $|N|$-th program if $\mathcal{S}_i = 2^N$ for some $i < |N|$.

12.3 Compact representations of coalitional games

Our focus so far has been on analyzing coalitional games. Now that we have some solution concepts under our belts, a natural question is whether and how we can efficiently compute these solution concepts. However, we immediately run into the problem that a straightforward representation of coalitional games by enumeration requires space exponential in the number of agents. This has the odd side-effect that simple brute-force approaches appear to have "good" (i.e., low-order polynomial) complexity, since complexity measures express the amount of time it will take to solve a problem as a function of the input size. Nevertheless, applying these algorithms to the straightforward representation will only allow us to compute solution concepts for problems with very few agents.

In order to ask more interesting questions we must first find a game representation that is more compact. In general one cannot compress the representation, but many natural settings contain inherent symmetries and independencies that do give rise to more compact representations. Conceptually, this section mirrors the discussion in Section 6.5, which considered compact representations of noncooperative games.

In the following, we discuss a number of compactly-represented coalitional games. These games can be roughly categorized into two types. First, we look at games that arise from specific applications, such as weighted voting games and weighted graph games. We then look at "languages" designed to represent coalitional games. This includes the synergy representation for superadditive games and the multi-issue and marginal contribution nets representations for general games.

12.3.1 *Weighted majority games and weighted voting games*

The weighted majority game representation is a compact way of encoding voting situations. Its definition is straightforward.

weighted majority game

Definition 12.3.1 (Weighted majority game) *A* weighted majority game *is defined by weights w_i assigned to each player $i \in N$. Let W be $\sum_{i \in N} w(i)$. The value of a coalition is* 1 *if $\sum_{i \in S} w(i) > \frac{W}{2}$ and* 0 *otherwise.*

Since this game is simple (in the sense of Definition 12.1.10), testing the nonemptiness of the core is equivalent to testing the existence of a veto player, which can be done quickly. However, it is not so easy to compute the Shapley value.

Theorem 12.3.2 *Computing the Shapley value in weighted majority games is #P-complete.*[7]

This can be proved by a reduction from the counting version of KNAPSACK.

weighted voting games

Weighted voting games are natural generalization of weighted majority games.

7. Recall that #P consists of the counting versions of the deterministic polynomial-time decision problems; for example, not simply deciding whether a Boolean formula is satisfiable, but rather counting the number of truth assignments that satisfy the formula.

Instead of stipulating that all coalitions with more than half the votes win, an explicit minimum number of votes, known as the *threshold*, is specified. This representation can be used to represent voting situations in which the number of votes required for the selection of a candidate is not a simple majority.

12.3.2 *Weighted graph games*

A weighted graph game (WGG) is a coalitional game defined by an undirected weighted graph (i.e., a set of nodes and a real-valued weight associated with every unordered pair of nodes). Intuitively, the nodes in the graph represent the players, and the value of a coalition is obtained by summing the weights of the edges that connect pairs of vertices corresponding to members of the coalition. WGGs thus explicitly model pairwise synergies among players and assume that all such synergies increase the coalition's value additively. In exchange for this reduced expressiveness we gain a much more compact representation: a game with n agents is represented by only $\frac{n(n-1)}{2}$ weights.

weighted graph game

Definition 12.3.3 (Weighted graph game) *Let (V, W) denote an undirected weighted graph, where V is the set of vertices and $W \in \mathbb{R}^{V \times V}$ is the set of edge weights; denote the weight of the edge between the vertices i and j as $w(i, j)$. This graph defines a* weighted graph game *(WGG), where the coalitional game is constructed as follows:*

- $N = V$;
- $v(S) = \sum_{i,j \in S} w(i, j)$.

An example that WGGs model well is the Revenue Sharing game.

Example 12.3.4 (Revenue Sharing game) *Consider the problem of dividing the revenues from toll highways between the cities that the highways connect. The pair of cities connected by a highway get to share in the revenues only when they form an agreement on revenue splitting; otherwise, the tolls go to the state. This problem can be represented as a weighted graph game (V, W), where the nodes V represent the cities, each edge represents a highway between a pair of cities, and the weight $w(i, j)$ of a given edge indicates that highway's toll revenues.*

The following is a direct consequence of the definitions.

Proposition 12.3.5 *If all the weights are nonnegative then the game is convex.*

Thus we know that in this case WGGs have a nonempty core and, furthermore, that the core contains the Shapley value. But the core may contain additional payoff vectors, and it is natural to ask whether testing membership is easy or hard. The answer is given by the following proposition.

Proposition 12.3.6 *If all the weights are nonnegative then membership of a payoff vector in the core can be tested in polynomial time.*

The proof is achieved by providing a maxflow-type algorithm.

The Shapley value is also easy to compute, even when we lift our restriction that the weights must be nonnegative.

Theorem 12.3.7 *The Shapley value of the coalitional game (N, v) induced by a weighted graph game (V, W) is*

$$\phi_i(N, v) = \frac{1}{2} \sum_{j \neq i} w(i, j).$$

Proof. Consider the contribution of the edge (i, j) to $\phi_i(N, v)$. For every subset S containing it, it contributes $\frac{(n-|S|)!(|S|-1)!}{n!} w(i, j)$. There are $\binom{n-2}{k-2}$ subsets of size k that contain both i and j. So all subsets of size k contribute $\binom{n-2}{k-2} \frac{(n-|S|)!(|S|-1)!}{n!} w(i, j) = \frac{(k-1)}{n(n-1)} w(i, j)$. And by summing over $k = 2, \ldots, n$ we obtain the result. ■

It follows that we can compute the Shapley value in $O(n^2)$ time.

Answering questions regarding the core of WGGs is more complex. Recall that a cut in a graph is a set of edges that divide the nodes into two disjoint sets, the weight of a cut is the sum of its weights, and a negative cut is a cut whose weight is negative. We begin by noting the following proposition.

Proposition 12.3.8 *The Shapley value is in the core of a weighted graph game if and only if there is no negative cut in the weighted graph.*

Proof. Note that while the value of a coalition S is the sum of the weights within S, the Shapley values in the same coalition is the same sum plus half the total weights of edges between S and $N \setminus S$. But the edges between S and $N \setminus S$ form a cut. Clearly, if that cut is negative, the Shapley value cannot be in the core, and since this holds for all sets S, the converse is also true. ■

And so we get as a consequence that if the weighted graph contains no negative cuts, the core cannot be empty. The next theorem turns this into a necessary and sufficient condition.

Theorem 12.3.9 *The core of a weighted graph game is nonempty if and only if there is no negative cut in the weighted graph.*

Proof. The if part follows from the preceding proposition.

For the only-if part, suppose we have a negative cut in the graph between S and $N \setminus S$. By virtue of being a cut, we have

$$\sum_{i \in S} \phi_i(N, v) - v(S) = \sum_{i \in (N \setminus S)} \phi_i(N, v) - v(N \setminus S) = \frac{\sum_{i \in S, j \in (N \setminus S)} w(i, j)}{2} < 0.$$

For any payoff vector x, we have

$$\begin{aligned} v(N) &= \sum_{i \in N} x_i = \sum_{i \in S} x_i + \sum_{i \in (N \setminus S)} x_i \\ &= \sum_{i \in N} \phi_i(N, v) = \sum_{i \in S} \phi_i(N, v) + \sum_{i \in (N \setminus S)} \phi_i(N, v). \end{aligned}$$

Combining the two, and summing up,

$$\left(\sum_{i \in S} x_i - v(S)\right) + \left(\sum_{i \in (N \setminus S)} x_i - v(N \setminus S)\right) < 0.$$

Hence, either the first or the second term (possibly both) has to be negative. The payoff vecor x is not in the core. Since x is an arbitrary payoff vector, the core is empty. ■

These theorems suggest that one test for nonemptiness of the core is to check whether the Shapley solution lies in the core. However, despite all these promising indications, testing membership in WGGs remains elusive in general.

Theorem 12.3.10 *Testing the nonemptiness of the core of a general WGG is NP-complete.*

The proof is based on a reduction from MAXCUT, a well-known NP-complete problem.

12.3.3 *Capturing synergies: a representation for superadditive games*

So far, we have looked at compact representations of coalitional games that can express only very restricted classes of games, but that are extremely compact for those classes. We now switch gears to consider compact representations that are designed with the intent to represent more diverse families of coalitional games.

The first one we will examine can be used to represent any superadditive game. As mentioned earlier in the chapter, superadditivity is sometimes assumed for coalitional games and is justified when coalitions do not exert negative externalities on each other.

synergy representation

Definition 12.3.11 (Synergy representation) *The* synergy representation *of superadditive games is given by the pair* (N, s), *where* N *is the set of agents and* s *is a set function that maps each coalition to a value interpreted as the* synergy *the coalition generates when the agents of the coalition work together. Only coalitions with strictly positive synergies will be included in the specification of the game.*

The underlying coalitional game (N, v) *under representation* (N, s) *is given by*

$$v(S) = \left(\max_{\{S_1, S_2, \ldots, S_k\} \in \pi(S)} \sum_{i=1}^{k} v(S_i)\right) + s(S),$$

where $\pi(S)$ *denotes the set of all partitions of* S.

Note that for some superadditive games, the representation may still require space exponential in the number of agents. This is unavoidable as the space of coalitional games of n agents, when treated as a vector space, is of $2^n - 1$ dimensions. However, for many games, the space required is much less.

We can evaluate the usefulness of a representation based on a number of criteria. One criterion is whether the representation exposes the structure of the underlying game to facilitate efficient computation. For example, is it easy to find out the value of a given coalition? Unfortunately, the answer is negative for the synergy representation.

Proposition 12.3.12 *It is NP-complete to determine the value of some coalitions for a coalitional game specified with the synergy representation. In particular, it is NP-complete to determine the value of the grand coalition.*

Intuitively, the reason why it is hard to determine the value of a coalition under the synergy representation is that we need to pick the best partitioning for a coalition, for which there is a number of choices exponential in the size of the coalition. As a result, it is (co)NP-hard to determine whether a given payoff vector is in the core, and it is NP-hard to compute the Shapley value of the game, as solution to either problem can be used to compute the value of the grand coalition. It is also (co)NP-hard to determine whether the core is empty or not, which follows from a reduction from Exact Cover by 3 Sets.

Interestingly, if the value of the grand coalition is given as part of the input of the problem, then the emptiness of the core can be determined efficiently.

Theorem 12.3.13 *Given a superadditive coalitional game specified with the synergy representation and the value of the grand coalition, we can determine in polynomial time whether the core of the game is empty or not.*

The proof is achieved by showing that a payoff in the core can be found by solving a linear program.

12.3.4 *A decomposition approach: multi-issue representation*

The central idea behind the multi-issue representation, a representation based on game decomposition, is that of *addition* of games (in the sense of Axiom 12.2.6 (Additivity) in the axiomatization of the Shapley value). Formally, the multi-issue representation is given as follows.

multi-issue representation

Definition 12.3.14 (Multi-issue representation) *A* multi-issue representation *is composed of a collection of coalitional games, each known as an issue,* (N_1, v_1), $(N_2, v_2), \ldots, (N_k, v_k)$, *which together constitute the coalitional game* (N, v), *where:*

- $N = N_1 \cup N_2 \cup \cdots \cup N_k$*; and*
- *for each coalition* $S \subseteq N$, $v(S) = \sum_{i=1}^{k} v_i(S \cap N_i)$.

Intuitively, each issue of the game involves some set of agents, which may be partially or completely overlapping with the set of agents for another issue. The value of a coalition in the game is the sum of the values achieved by the coalition in each issue. For example, consider a robotics domain where there are certain tasks to be performed, each of which can be performed by a certain subset of a group of robots. We can then treat each of these tasks as an issue and model the system as a coalitional game where the value of any group of robots is the sum of the values of the tasks the group can perform.

Clearly, the multi-issue representation can be used to represent any coalitional game, as we can always choose to treat the coalitional game as a single big issue.

From the computational standpoint, due to its close relationship to the additivity axiom, it is perhaps not surprising that the Shapley value of a coalitional game specified in the multi-issue representation can be found efficiently.

Proposition 12.3.15 *The Shapley value of a coalitional game specified with the multi-issue representation can be computed in time linear in the size of the input.*

This is not hard to see. First, note that the Shapley value of a game can be computed in linear time when the input is given by the enumeration of the value function. This is because the direct approach of computing the Shapley value requires summing over each coalition once, and so the total number of operations is linear in the size of the enumeration. Observe that the factorials can be computed quickly to any desired accuracy using the Stirling approximation. Then, to prove the proposition, we must simply use the fact that the Shapley value satisfies the additivity axiom.

On the other hand, the multi-issue representation does not help with computational questions about the core. For example, it is coNP-hard to determine if a given payoff vector belongs to the core when the game is specified with the multi-issue representation.

12.3.5 *A logical approach: marginal contribution nets*

Marginal contribution nets (MC-nets) contstitute a representation scheme for coalitional games that attempts to harness the power of boolean logic to reduce the space required for specifying a coalitional game. The basic idea is to treat each agent in a game as a boolean variable and to treat the (binary) characteristic vector of a coalition as a truth assignment. This truth assignment can be used to evaluate whether a boolean formula is satisfied, which can in turn be used to determine the value of a coalition.

marginal contribution net (MC-net)

Definition 12.3.16 (MC-net representation) *An* MC-net *consists of a set of rules. Each rule has the syntactic form* (Pattern, weight)*, where the pattern is given by a boolean formula and the weight is a real value.*

The MC-net $(p_1, w_1), (p_2, w_2), \dots, (p_k, w_k)$ *specifies a coalitional game* (N, v)*, where* N *is the set of propositions that appear in the patterns and the value function is given by*

$$v(S) = \sum_{i=1}^{k} p_i(e^S) w_i,$$

where $p_i(e^S)$ *evaluates to* 1 *if the boolean formula* p_i *evaluates to true for the truth assignment* e^S *and* 0 *otherwise.*

As an example, consider an MC-net with two rules: $(a \wedge b, 5), (b, 2)$. The coalitional game represented has two agents, a and b, and the following value function.

$$v(\emptyset) = 0 \qquad v(\{a\}) = 0$$
$$v(\{b\}) = 2 \qquad v(\{a, b\}) = 5 + 2 = 7$$

An alternative interpretation of MC-nets is a graphical representation. We can treat the agents as nodes on a graph, and for each pattern, a clique is drawn on the graph for the agents that appear in the same pattern. The weight of a rule is the weight associated with the corresponding clique.

A natural question for a representation language is whether there is a limit on the class of objects it can represent.

Proposition 12.3.17 *MC-nets can represent any game when negative literals are allowed in the patterns or when the weights can be negative. When the patterns are limited to conjunctive formula over positive literals and the weights are nonnegative, MC-nets can represent all and only convex games.*

Intuitively, when negative literals are allowed, we can specify the value of each coalition S directly by having a boolean formula that can be satisfied if and only if the truth assignment corresponds to the characteristic vector of S, and hence arbitrary games can be represented.

Another question is how a language relates to other representation languages. We can show that MC-nets generalize two of the previously discussed representations: WGGs and the multi-issue representation. First, MC-nets can be viewed as a generalization of WGGs that assigns weights to hyper-edges rather than to simple edges. A pattern in this case specifies the agents that share the same edge. However, since MC-nets can represent any coalitional games, MC-nets constitute a strict generalization of WGGs. Second, MC-nets generalize the multi-issue representation. Each issue is represented by patterns that only involve agents relevant to the issue. Comparing the two representations, MC-nets require at most $O(n)$ more space than the multi-issue representation; however, there exist coalitional games for which MC-nets are exponentially more succinct (in the number of agents) than the multi-issue representation.

From a computational standpoint, when only limited to conjunctions in the Boolean formula, MC-nets still make the Shapley value just as easy to compute as did the multi-issue representation.

Theorem 12.3.18 *Given a coalitional game specified with an MC-net limited to conjunctive patterns, the Shapley value can be computed in time linear in the size of the input.*

However, when other logical connectives are allowed, there is no known algorithm for finding the Shapley value efficiently. Essentially, finding the Shapley value involves summing factors of hypergeometric distributed variables, a problem for which there is no known closed-form solution.

Since MC-nets generalize WGGs, the problems of determining whether the core is empty and whether a payoff vector belongs to the core are both coNP-hard. However, there exists an algorithm that can solve both problems in time exponential only in the tree-width of the graphical representation of the MC-net.

12.4 Further directions

Before we conclude the chapter, we briefly survey some more advanced topics in coalitional game theory.

12.4.1 *Alternative coalitional game models*

Let us first revisit the transferable utility assumption that we made at the beginning of the chapter. In some situations, this assumption is not reasonable, for example, due to legal reasons (agents cannot engage in side payments), or because the agents do not have access to a common currency. Such settings are described as nontransferable utility (NTU) games.

coalitional game with nontransferable utility

Definition 12.4.1 (Coalitional game with nontransferable utility) *A* coalitional game (with nontransferable utility) *is a pair* (N, v), *where:*

- N *is a finite set of players, indexed by* i; *and*
- $v : 2^N \mapsto 2^{\mathbb{R}^{|S|}}$ *associates each coalition* $S \subseteq N$ *with a set of value vectors* $v(S) \subseteq \mathbb{R}^{|S|}$, *which can be interpreted as the different sets of payoffs that* S *is able to achieve for each of its members.*

Note that the function v returns *sets* of value vectors rather than single real numbers, as in the case of coalitional games with transferable utility. Thus, rather than giving the total amount of utility and allowing agents to divide it arbitrarily among themselves, coalitional games with nontransferable utility explicitly list all the divisions that are possible and prohibit the rest.

It might seem that there is no problem left to be solved in the case of NTU games—after all, earlier we largely focused on payoff division, and in these games payoffs cannot be divided at all. However, in these games it is interesting to study which coalitions form.

For example, consider the matching problem we introduced in Section 10.6.4, pairing graduate students with advisors. This setting can be modeled as a coalitional game with nontransferable utility as follows. Let $N = A \cup S$ be the set of players in the game. Let Λ be the set of all possible matchings, and let $\mu(i)$ denote the agent matched with agent i, where $\mu \in \Lambda$. For each coalition T, the payoffs achievable by its members, $A' \cup S'$, are the preferences induced by all possible matchings among the group's members. As a concrete example, suppose each student in S achieves a payoff of $|A|$ by being matched to his most preferred advisor, $|A| - 1$ for being matched to his second most preferred advisor, and so on, with a payoff of 0 for not being matched. In general, write $u_s(\mu)$ to denote student s's payoff for matching μ. Similarly, let there be such a payoff function over students for each advisor. For each coalition $T \subseteq N$, each different matching within the group gives rise to different payoff vectors for its members.

A matching μ is in the core if there is no coalition of advisors and students where all members of the coalition are weakly better off matching only among themselves, and at least one member is strictly better off. Mathematically, μ is in the core if and only if there is no matching μ' and set $S \subseteq N$ such that

1. $\forall i \in S, \mu'(i) \in S$;
2. $\forall i \in S, u_i(\mu') \geq u_i(\mu)$; and
3. $\exists i \in S, u_i(\mu') > u_i(\mu)$.

As it turns out, the core of the matching game is always nonempty.

Another commonly-made assumption is that the value (or in the case of NTU games, the set of achievable payoff vectors) of a coalition is independent of the other coalitions. This is at best an approximation for many situations. If the value of a coalition can depend meaningfully on what other coalitions form, one has to take into account the whole *coalitional structure* when assigning payoffs to the coalitions.

coalitional game in partition form

Definition 12.4.2 (Coalitional game in partition form) *A* coalitional game in partition form *is a pair* (N, p)*, where:*

coalitional structure

- *N is a finite set of players, indexed by i; and*
- *p associates each partition π of N (also known as a* coalitional structure*) and a coalition $S \in \pi$ with $p(\pi, S) \subseteq \mathbb{R}^S$, to be interpreted as the set of payoffs that S can achieve for its members under partition π.*

Yet another important direction is the incorporation of uncertainties into coalitional game models. Unlike incomplete-information games in noncooperative game theory, where a well-developed theory exists, a universally-accepted theory of coalitional games with uncertainty has yet to emerge. Such a theory would be useful. In many situations, it is natural to suppose that the values to coalitions are not known with certainty. It is also reasonable to assume that the payoffs to a coalition could depend on private information held by the agents, about which other agents have only probabilistic information. Some efforts toward modeling coalitional games under uncertainty have been made in the work cited at the end of the chapter; yet more work is still needed to create a fully complete theory.

12.4.2 *Advanced solution concepts*

There are a number of other solution concepts that we have not discussed in this chapter. One interesting phenomenon in the analysis of coalitional games is that researchers seem to have quite diverse opinions as to which solution concepts make the most sense. This is perhaps natural since different conflicts call for different notions of stability and fairness. Some other important solution concepts in the literature include *stable sets*, the *bargaining set*, and the *kernel*. Some of these, for example, attempt to capture the intuition that should a coalition deviate due to the violation of some stability property, that it should deviate to a stable coalition itself.

However, interestingly, most solution concepts to date focus on dividing the payoff of the grand coalition. While this appears quite reasonable when the core of the game is nonempty, when the core is empty, one might expect coalitions other than the grand coalition to form. This problem is worse when the game is not superadditive; in this case, it is possible that some partitioning of the coalitions could achieve strictly higher total payoffs than the grand coalition. It is therefore important to consider solution concepts that allow payoffs to depend on the coalitional structure.

Finally, computing the agents' payoffs is often only part of the problem. It can also be important to find out what coalitions would and should form, and how agents should coordinate their actions. This has been an area traditionally ignored in the literature, perhaps due to a focus on the abstract properties of coalitional games. For certain applications, the coalitional formation process cannot be ignored. Indeed, much work in artificial intelligence has been devoted to analyzing the process of coalition formation. By applying coalitional game theory to analyze such process, it may be possible to learn more about the strategic properties of different coordination mechanisms in the presence of selfish agents.

12.5 History and references

In the early days of game theory research, coalitional game theory was a major focus, particularly of economists. This is partly because the theory is closely related to equilibrium analysis and seemingly bridges a gap between game theory and economics. Von Neumann and Morgenstern, for example, devoted more than half of their classic text, *Theory of Games and Economic Behavior* [von Neumann and Morgenstern, 1944], to an analysis of coalitional games. A large body of theoretical work on coalitional game theory has focused on the development of solution concepts, possibly in an attempt to explain the behavior of large systems such as markets. Solid explanations of the many solution concepts and their properties are given by Osborne and Rubinstein [1994] and Peleg and Sudhölter [2003].

Some examples used in this chapter have appeared in other contexts. The connection between matching and coalitional game theory has been explored by a number of economists and surveyed in the works of Al Roth (see e.g., Roth and Sotomayor [1990]). The airport game and the minimum spanning tree game appeared in Peleg and Sudhölter [2003]. The connection between auctions and core was explored, for example, by Day and Milgrom [2008].

The first systematic investigation of the computational complexity of solution concepts in coalitional game theory were carried out in Deng and Papadimitriou [1994]. This paper defined weighted graph games and studied the complexity of computing the core and the Shapley value, as well as a few of the other solution concepts we mentioned. For weighted voting games, a systematic study of the computational complexity of the various solution concepts have appeared in Elkind et al. [2007]. Languages for succinct representation of coalitional games have been developed mostly in the AI community. The superadditive representation was developed by Conitzer and Sandholm [2003a]; it also naturally extends to representing superadditive games with nontransferable utilities. The multi-issue representation was also developed by Conitzer and Sandholm [2004]. The marginal contribution nets representation was first proposed in Ieong and Shoham [2005], and later generalized in Ieong and Shoham [2006].

Work on coalitional games under uncertainty includes Suijs et al. [1999], Chalkiadakis and Boutilier [2004], Myerson [2007], and Ieong and Shoham [2008]; as mentioned earlier, many open problems remain.