Chapter 4

Characteristic Form Games and Coalition Formation

There is another type of game studied in game theory: the **characteristic form** game or **coalition** game (Osborne and Rubinstein, 1999). In these games the agents decide how to form coalitions among themselves and each coalition receives some utility. For example, a group of people, each with different skills, all want to start new companies. The problem they face is deciding how to divide themselves into subgroups such that each subgroup has the needed set of skills to succeed in the marketplace. Similarly, a group of agents with different skills must decide how to divide itself into subgroups so as handle as many tasks as possible in the most efficient manner. Because the agents must cooperate with each other in order to form coalitions and an agent cannot unilaterally decide that it will form a coalition with a second agent, these games are known as **cooperative games**. Multiagent researchers have also extended the basic characteristic form into to the more general coalition formation, which we also present in this chapter.

CHARACTERISTIC FORM
COALITION

COOPERATIVE GAMES

It is interesting to note that most game theory textbooks focus exclusively on non-cooperative games as these have found many applications in Economics and Business and have been the focus of most of the research. However, when building multiagent systems we find that cooperative games are much more useful since they clearly and immediately model the problem of which agents should perform which tasks.

4.1 Characteristic Form Games

Formally, a game in characteristic form includes a set $A = \{1, \ldots, |A|\}$ of agents. The agents are assumed to deliberate and the final result of the deliberation is an **outcome** $\vec{u} = (u_1, \ldots, u_{|A|}) \in \Re^{|A|}$ which is just a vector of utilities, one for each agent. There is also a rule $V(\cdot)$ that maps every coalition $S \subset A$ to a utility possibility set, that is $V(S) \subset \Re^{|S|}$. Notice that $V(S)$ returns a set of utility vectors, not a single utility vector. As such, $V(\cdot)$ provides us the set of payoffs that players in S can achieve it they form a coalition. For example, for the players $\{1, 2, 3\}$ we might have that $V(\{1, 2\}) = \{(5, 4), (3, 6)\}$ meaning that if agents 1 and 2 formed a coalition they could either get 5 for agent 1 and 4 for agent 2, or they could get 3 for agent 1 and 6 for agent 2. The function V must be defined for all subsets of A.

OUTCOME

A special case of the characteristic form game–the one nearly all multiagent research focuses on—is the **transferable utility** game in characteristic form. This game assumes that the players can exchange utilities among themselves as they see fit. For example, if the utility payments are in the form of money then we only need to specify the total amount of money the coalition will receive and decide later how this money will be distributed among the agents in the coalition. More formally, we define a transferable utility game

TRANSFERABLE UTILITY

Definition 4.1 (Tranferable utility characteristic form game)**.** *These games consist of a set of agents* $A = \{1, \ldots, A\}$ *and a* characteristic function $v(S) \rightarrow \Re$ *defined for every* $S \subseteq A$*.*

The **characteristic function** $v(S)$ is also sometimes simply referred to as the **value function** for the coalitions. Characteristic form games with transferable

CHARACTERISTIC FUNCTION
VALUE FUNCTION

S	$v(S)$
(1)	2
(2)	2
(3)	4
(12)	5
(13)	7
(23)	8
(123)	9

(1)(2)(3)
$2+2+4=8$

(1)(23)
$2+8=10$

(2)(13)
$2+7=9$

(3)(12)
$4+5=9$

(123)
9

Figure 4.1: Sample characteristic form game with transferable utility for three agents: 1, 2 and 3. The table on the left shows the values of each coalition. On the right are the coalition structures. Below each one we calculate its value.

utility can represent many multiagent scenarios. For example, they can represent a task allocation problem where a set of tasks has to be performed by a set of agents, subsets of whom can sometimes improve their performance by joining together to perform a task. They can represent a sensor network problems where the sensors must join together in subgroups to further refine their readings or relay important information, or they can represent workflow scheduling systems where agents must form groups to handle incoming workflows.

4.1.1 Solution Concepts

As is often the case in game theory, there is no clear best solution to all characteristic form games. Instead, various solutions concepts have been proposed each one having its own advantages and disadvantages.

Before defining the solution concepts, we must first notice that the outcome as we have defined it allows for impossible utility values in the transferable utility game. Specifically, there might not be a set of coalitions such that, given v, the agents can all get their utility as promised by $\vec{u}$. In order to rectify this problem we first specify that we are only interested in **feasible** outcomes, that is, those that can be implemented given v.

FEASIBLE

Definition 4.2 (Feasible)**.** *An outcome $\vec{u}$ is* feasible *if there exists a set of coalitions $T = S_1, \ldots, S_k$ where $\bigcup_{S \in T} S = A$ such that $\sum_{S \in T} v(S) \geq \sum_{i \in A} \vec{u}_i$.*

That is, an outcome $\vec{u}$ is feasible if we can find a disjoint set of coalitions whose values are as much as that in $\vec{u}$, so we can payoff $\vec{u}$ with v. The set of disjoint coalitions T defined above is also often referred to as a **coalition structure**, also sometimes represented with the symbol CS.

COALITION STRUCTURE

Notice that, if the characteristic function is super-additive then we can check if an outcome $\vec{u}$ is feasible by simply ensuring that

$$\sum_{i \in A} \vec{u}_i = v(A). \tag{4.1}$$

We define a **super-additive** domain as one where, for all pairs of disjoint coalitions $S, T \subset A$, we have that $v(S \cup T) \geq v(S) + v(T)$. That is, there is nothing to be lost by merging into a bigger coalition. Unfortunately, multiagent systems are rarely supper-additive since agents have a habit of getting into each others' way, so that a team is not always better than letting each agent work on a task separately.

SUPER-ADDITIVE

The problem of finding feasible solutions can best be illustrated with an example. Figure 4.1 shows a sample transferable utility game for three agents along with the definition of the v function and all possible coalition formations. In this game the outcome $\vec{u} = \{5, 5, 5\}$ is not feasible since there is no way to divide the agents into subsets such that they can all get their utility. If we tried the coalition (123) then we only have a value of 9 to distribute and we need a total of 15. On the other hand, the outcome $\vec{u} = \{2, 4, 3\}$, is feasible because the coalition structures (1)(23), (2)(13), and (123) can satisfy it (but not the other ones). However, $\vec{u} = \{2, 4, 3\}$ does have a problem in that in it agent 3 is getting an utility of 3 while we have that $v((3)) = 4$. That is, agent 3 could defect any one of the three coalition structures

S	$v(S)$
(1)	1
(2)	2
(3)	2
(12)	4
(13)	3
(23)	4
(123)	6

(1)(2)(3)
$1+2+2=5$

(1)(23)
$1+4=5$

(2)(13)
$2+3=5$

(3)(12)
$2+4=6$

$\vec{u}$	in Core?
$\{2,2,2\}$	*yes*
$\{2,2,3\}$	*no*
$\{1,2,2\}$	*no*

(123)
6

Figure 4.2: Sample game showing some outcomes that are in the core and some that are not. The characteristic function v is super-additive.

we found, join the coalition (3), and get a higher utility than he currently has. This outcome thus seems unstable.

The Core

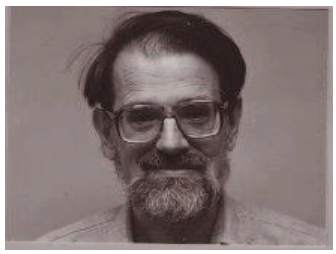
In general, we say that an outcome $\vec{u}$ is stable if no subset of agents gets paid more, as a whole, than what they get paid in $\vec{u}$. Stability is a nice property because it means that the agents do not have an incentive to go off into their own coalition. Our first solution concept, the **core**, refers to all the outcomes that are stable.

CORE

Definition 4.3 (Core)**.** *An outcome $\vec{u}$ is in the* core *if*

1. $$\forall_{S \subset A} : \sum_{i \in S} \vec{u}_i \geq v(S)$$

2. *it is feasible.*



Lloyd F. Shapley. 1923–. Responsible for the core and Shapley value solution concepts.

The first condition in this definition tells us that the utility the agents receive in outcome $\vec{u}$ is bigger than those of any coalition, for the agents in the coalition. In other words, that there is no coalition S whose $v(S)$ is bigger than the sum of payments the agents in S get under $\vec{u}$. The second condition merely checks that the total utility we are giving out is not more than what is coming in via $v(\cdot)$.

Figure 4.2 shows a new game with different payments and a list of outcomes some of which are in the core and some which are not. $\{2,2,2\}$ is in the core because it is feasible and there is no subset of agents S with a $v(S)$ that is bigger than what they could get in this outcome. $\{2,2,3\}$ is not in the core because it is not feasible. This outcome adds up to 7 and there is no coalition structure that adds up to 7. Note that, since the v is super-additive, all we need to check is the grand coalition. Finally, $\{1,2,2\}$ is not in the core because agents 1 and 2 are getting a total of 3 while if they formed the coalition (12) they would get a utility of 4.

We like the core because we know that any solution that is in the core cannot be improved by having any of the 2^A subsets of agents form a coalition of higher value than they are getting now. It is a very stable solution. Unfortunately, there are many games with empty cores. Figure 4.3 shows one such example. Try to find an outcome in the core for this example. You will see that every outcome is blocked by some other outcome.

Another problem with the core solutions is that when it is not empty then there are often many outcomes in the core. For example, in figure 4.2 any outcome $\vec{u} = \{x, y, z\}$ where $x + y + z \leq 6$ and $x + y \geq 4$ and $x + z \geq 3$ and $y + z \geq 4$ is in the core. Thus, we still have a coordination or negotiation problem where we must choose between these coalitions. Finally, there is the fact that the outcome does not tell us which coalitions are formed. For example, if we choose the outcome $\vec{u} = \{2,2,2\}$ then we must still also choose a coalition structure as there are two which would work: (123) and (12)(3).

S	$v(S)$
(1)	0
(2)	0
(3)	0
(12)	10
(13)	10
(23)	10
(123)	10

Figure 4.3: Set of payments for a game with an empty core

S	$v(S)$
()	0
(1)	1
(2)	3
(12)	6

Figure 4.4: Example game. If the agents form coalition (12) then how much utility should each one get?

The Shapley Value

While the core gives us one possible solution, it suffers from the fact that many games don't have any solutions in the core and from its lack of guidance in fairly distributing the payments from a coalition to its members. The Shapley value solves these problems by giving us one specific set of payments for coalition members, which are deemed fair.

The problem with identifying fairness in characteristic form games is best illustrated by an example. Figure 4.4 shows a game for two players. Clearly, we should choose the coalition (12) as it has the highest value. Now we must decide how much each agent should get. The simplest solution is to divide the total of 6 evenly amongst the coalition members, so that each agent gets 3. This seems unfair to agent 2 because agent 2 could have gotten 3 by simply staying on its own coalition (2). It seems like the fair thing to do is to give each agent a payment that is proportional to the value it contributes to the coalition, that is, the amount that value increases by having the agent in the coalition. But, how do we extend this idea to cases with more than 2 agents?

Shapley was able to extend this idea by realizing that each agent should get a payment that corresponds to its marginal contribution to the final value. An agent's marginal contribution to a coalition is the difference between the value before the agent joins the coalition and after he joined. For example, if before you join Initech their annual profits are \$10M but after you are there for a year they increase to \$11M then you can claim that your marginal contribution is to Initech is \$1M assuming, of course, that everything else stays the same during that year.

The one remaining problem is that there are many different orderings in which n agents could have joined the coalition, namely, there are $n!$ orderings of n elements. The **Shapley value** simply averages over all possible orderings. That is, the Shapley value gives each agent a utility proportional to its average marginal contribution to every possible coalition, in every possible order it could have been formed. More formally, we define the Shapley value as:

SHAPLEY VALUE

Definition 4.4 (Shapley Value)**.** *Let $B(\pi, i)$ be the set of agents in the agent ordering π which appear before agent i. The* Shapley value *for agent i given A agents is given by*

$$\phi(A, i) = \frac{1}{A!} \sum_{\pi \in \Pi_A} v(B(\pi, i) \cup i) - v(B(\pi, i)),$$

where Π_A is the set of all possible orderings of the set A. Another way to express the same formula is

$$\phi(A, i) = \sum_{S \subseteq A} \frac{(|A| - |S|)!\,(|S| - i)!}{|A|!} [v(S) - v(S - \{i\})].$$

Notice that the Shapley values are calculated for a particular coalition A in the definition above. They are not meant as a way of determining which is the best coalition structure. They can only be used to distribute the payments of a coalition once it is formed.

Lets calculate the Shapley values for the game in figure 4.4 and the grand coalition (12). Since there are only two agents it means that there are only two possible orderings: (12) and (21). As such we have that

$$\begin{aligned}
\phi(\{1,2\},1) &= \frac{1}{2}\cdot(v(1)-v()+v(21)-v(2)) \\
&= \frac{1}{2}\cdot(1-0+6-3)=2 \\
\phi(\{1,2\},2) &= \frac{1}{2}\cdot(v(12)-v(1)+v(2)-v()) \\
&= \frac{1}{2}\cdot(6-1+3-0)=4
\end{aligned}$$

A somewhat surprising and extremely useful characteristic of the Shapley value is that it is always feasible. In our example the payments of 4 and 2 add up to 6 which is the same value we get in the grand coalition (12). Another nice feature of the Shapley value is that it always exists and is unique. Thus, we do not have to worry about coordination mechanism to choose among different payments. A final interesting result is that the Shapley value might not be in the core, even for cases where the core exists. This is a potential problem as it means that the resulting payments might not be stable and some agents might choose to leave the coalition in order to receive a higher payment on a different coalition.

Unfortunately, while the Shapley value has some very attractive theoretical properties, it does have some serious drawbacks when we try to use it for building multiagent systems. The biggest problem is computational. The Shapley value requires us to calculate at least $2^{|A|}$ orderings, this is only possible for very small sets A. It also requires that we know the value of v for every single subset S. In many real-world applications the calculation of v is complex. For example, it might require simulating how a particular coalition of agents would work together. These complex calculations could dramatically increase the total time. Finally, the Shapley value does not give us the actual coalition structure. Thus, it only solves the second part of the coalition formation problem. We must still determine which coalition the agents will form and how they will do it.

The Nucleolus

Since the core is often empty, researchers started looking for ways of relaxing it and find a new solution concept that would exist for every game. The problem with the core is that it says that there is no subset of agents that could get paid more than what they are currently getting paid in $\vec{u}$, because then they would be tempted to defect and form a new coalition. If it is impossible to find such an $\vec{u}$ then the next best thing would be to find the $\vec{u}$ that minimizes the total temptation felt by the agents. That is what the **nucleolus** aims to do.

NUCLEOLUS

We start by clarifying what we mean by temptation. Specifically, a coalition S is more tempting the higher its value is over what the agents get in $\vec{u}$. This is known as the **excess**.

EXCESS

Definition 4.5 (excess)**.** *The* excess *of coalition S given outcome $\vec{u}$ is given by*

$$e(S,\vec{u}) = v(S) - \vec{u}(S),$$

where

$$\vec{u}(S) = \sum_{i\in S} \vec{u}_i.$$

That is, a coalition S has a positive excess, given $\vec{u}$, if the agents in S can get more from $v(S)$ than they can from $\vec{u}$. The more they can get from S the higher the excess. Note that, by definition, if an outcome $\vec{u}$ is in the core then all coalitions have a excess that is less than or equal to 0 with respect to that outcome. But, since we are now concerned with outcomes that are not in the core we will instead look for those with minimal excess. Since excess is defined for all possible subsets S we first need a way to compare the excesses of two outcomes. We do this by putting them in a sorted list and comparing the list. The one with the higher excess first is declared more excessive. More formally, for each $\vec{u}$ we find its excess for all subsets S and order these in a list; the higher excesses come first, as such

$$\theta(\vec{u}) = \langle e(S_1^{\vec{u}}, \vec{u}), e(S_2^{\vec{u}}, \vec{u}), \ldots, e(S_{2^{|A|}}^{\vec{u}}, \vec{u})\rangle, \tag{4.2}$$

where $e(S_i^{\vec{u}}, \vec{u}) \geq e(S_j^{\vec{u}}, \vec{u})$ for all $i < j$. We then define a lexicographical ordering $\succ$ over these lists where $\theta(\vec{u}) \succ \theta(\vec{v})$ is true when there is some number $q \in 1 \ldots 2^{|A|}$ such for all $p < q$ we have that $e(S_p^{\vec{u}}, \vec{u}) = e(S_p^{\vec{v}}, \vec{v})$ and $e(S_q^{\vec{u}}, \vec{u}) > e(S_q^{\vec{v}}, \vec{v})$ where the S_i have been sorted as per θ. That is, if $\theta(\vec{u}) \succ \theta(\vec{v})$ then that means that when we sort their excesses for all subsets their first, and greatest, excesses are all the same and on the first set for which they have a discrepancy $\vec{u}$ has the highest excess. For example, if we had the lists $\{(2,2,2),(2,1,0),(3,2,2),(2,1,1)\}$ they would be ordered as $\{(3,2,2),(2,2,2),(2,1,1),(2,1,0)\}$.

NUCLEOLUS

We can now define the **nucleolus** as the $\vec{u}$ which is not lexicographically bigger than anyone else.

Definition 4.6 (nucleolus)**.** *The* nucleolus *is the set*

$$\{\vec{u} \,|\, \theta(\vec{u}) \not\succ \theta(\vec{v}) \text{ for all } \vec{v}, \text{ given that } \vec{u} \text{ and } \vec{v} \text{ are feasible.}\}$$

In other words, the outcomes in the nucleolus are those where the excesses for all possible sets are lexicographically not greater than those of any other outcome. A nice feature of the nucleolus is that it is always unique for each coalition structure. That is, given a coalition structure there is only one nucleolus.

The nucleolus captures, to some degree, the idea of an outcome that minimizes the temptation the agents face. However, notice that the lexicographic order it defines only cares about the first coalition that has a higher excess, it does not care about the ones after that. This could lead to a situation where the sum of the excesses from the nucleolus is actually larger than that of some other outcome. For example, $(5.0, 0, 0)$ comes before $(4, 3, 3, 3)$. As such, the nucleolus does not seem to minimize the sum of temptations.

Equal Excess

EQUAL EXCESS

Another technique for calculating the agents' payoff, besides the Shapley value, is called **equal excess**. It is an iterative algorithm where we adjust the payments that the agents expected they will receive from each coalition that includes them. At each time step t we let $E^t(i, S)$ be agent i's expected payoff for each coalition S which includes him. Initially these are set to 0. We thus let

$$A^t(i, S) = \max_{T \neq S} E^t(i, T) \tag{4.3}$$

be agent i's expected payment from not choosing S and instead choosing the best alternative coalition. Then, at each time step we update the players' expected payments using

$$E^{t+1}(i, S) = A^t(i, S) + \frac{v(S) - \sum_{j \in S} A^t(j, S)}{|S|}. \tag{4.4}$$

For example, for the value function given in figure 4.4 we start with $E^0(1, *) = E^0(2, *) = 0$. Then, at time 1 agent 1 can update $E^1(1, (1)) = 1$ and $E^1(1, (12)) = 3$

(1)(2)(3)(4)

(12)(3)(4) (13)(2)(4) (14)(2)(3) (23)(1)(4) (24)(1)(3) (34)(1)(2)

(1)(234) (2)(134) (3)(124) (4)(123) (12)(34) (14)(23) (13)(24)

(1234)

Figure 4.5: Coalition structure formation possibilities for four agents, organized by the number of coalitions.

Level	Bound
A	$A/2$
$A-1$	$A/2$
$A-2$	$A/3$
$A-3$	$A/3$
$A-4$	$A/4$
$A-5$	$A/4$
:	:
2	A
1	none

Figure 4.6: Bounds on optimality after searching various levels.

and then $A^1(1,(1)) = 3$ and $A^1(1,(12)) = 1$, while agent 2 updates $E^1(2,(2)) = 3$ and $E^1(2,(12)) = 3$ and then $A^1(1,(1)) = 3$ and $A^1(1,(12)) = 3$.

It has been shown that this basic algorithm does not always converge to a fixed point, however, variations of it have been proposed which do converge, such as PACT (Goradia and Vidal, 2007). In PACT the agents calculate their own E values and exchange them with others under the assumption that agents will not lie about these values. The algorithm ensures that the process will stop and a solution will be found.

Notice that equal excess is a procedural solution to the problem so we do not know which specific outcome the agents will converge to, other than to say that they will converge to the outcome that is found when we use equal excess.

4.1.2 Finding the Optimal Coalition Structure

As multiagent system designers we often simply want to find the outcome that maximizes the sum of values. That is, we want to find the **utilitarian** solution. When the characteristic function is super-additive then the grand coalition will have the highest value and thus finding the utilitarian solution is trivial. However, if the characteristic is not super-additive—as is often the case in multiagent systems—then we will want an algorithm for finding it. Notice that under this formulation we are no longer interested in the specific outcome (that is, individual payments to agents) we are now only interested in finding best coalition structure, ignoring the problem of dividing up the value of each coalition among its participants.

UTILITARIAN

Centralized Algorithm

One proposed approach is to a perform a complete search of the complete set of possible coalition structures, but in a specified order. Figure 4.5 shows all the possible coalition structures for four agents. Notice that the bottom two rows contain all possible coalitions. This means that after searching those two rows we have seen all possible coalitions. If we let S^* be the value of the highest valued coalition (*not* coalition structure) found after searching those two rows then we know that the best coalition structure cannot be more than $A \cdot S^*$. As such, after searching the bottom two levels we can say that the optimal solution is no more than A times better than the best solution we have found thus far.

FIND-COALITION(i)
1 $L_i \leftarrow$ set of all coalitions that include i.
2 $S_i^* \leftarrow \arg\max_{S \in L_i} v_i(S)$
3 Broadcast S_i^* and wait for all other broadcasts, put these into S^* set.
4 $S_{max} \leftarrow \arg\max_{s \in S^*} v_i(s)$
5 **if** $i \in S_{\max}$
6 **then** join $S_{\max}$
7 **return**
8 **for** $j \in S_{\max}$
9 **do** Delete all coalitions in L_i that contain j
10 **if** L_i is not empty
11 **then** goto 2
12 **return**

Figure 4.7: Distributed algorithm for coalition formation. Each agent i must execute this function. We let $v_i(S) = \frac{v(S)}{|S|}$

Figure 4.6 shows the bounds that can be calculated after examining each of the levels in the graph. One simple algorithm consists of first searching the bottom two levels then continue searching down from the top level (Sandholm et al., 1999). In this way, the bound from optimal is reduced as indicated in the figure. Note that searching the levels in some other order will not guarantee these bounds. Notice also that the number of coalition structures in the second level is given by its **Stirling** number

STIRLING

$$\text{Stirling}(A, 2) = \frac{1}{2}\sum_{i=0}^{1}(-1)^i\binom{2}{i}(2-i)^A = 2^{A-1} - 1. \tag{4.5}$$

So it takes exponential time just to search the second level. In general, the number of coalition structures for all levels is equal to the Stirling number for that level.

There also exists an algorithm for finding the optimal coalition structure which has slightly better bounds than the ones we just presented, but running time remains exponential and unusable for large number of agents (Dang and Jennings, 2004).

Distributed Algorithm

find-coalition

While the previous algorithm found the optical coalition structure, it did so at the expense of a lot of computation and in a centralized manner. We now look at one possible way of finding a good, but possibly not optimal, coalition structure in a decentralized manner.

Figure 4.7 shows a distributed algorithm for coalition formation (Shehory and Kraus, 1998). The agents order all their possible coalitions based on how much each will get in that coalition, where each agent i gets $v_i(S) = v(S)/|S|$ if it joins coalition S. The agents then broadcast the name of their best coalition. The coalition with maximal $v(S)/|S|$ is chosen by the agents in S who join the coalition and drop out of the algorithm. The remaining agents take note of the missing agents by eliminating from consideration all coalitions that include them. The process is then repeated again with the new set of coalitions.

This is a classic example of a greedy or hill-climbing algorithm. As such, we know that it might get stuck on a local optima, which might or might not be the global optimum. Still, the algorithm should execute very fast as there are at most A steps and each step involves having each agent examine all the possible coalitions that it can be participate in.

A slight modification of the algorithm would be to, instead of broadcasting at each time step we could let the agents meet randomly and form a coalition if $v_i(s)$ is maximal for all agents in s. Imagine the agents moving around in a space and forming sets whenever a group of them happen to be close to each other, then forming a coalition only if that set has a maximal value. This process is effectively the same as broadcasting except that it eliminates the need to broadcast at the expense of taking a longer time to converge (Sarne and Arponen, 2007).

Reduction to Constraint Optimization

We note that the problem of finding the optimal coalition structure can be reduced to a constraint optimization problem. The basic idea is that for n agents there will be at most n coalitions as, at worst, each agent will stay in the individual coalition. Thus, we can imagine the problem as consisting of n agents each one deciding which of n rooms to go into. The agents are the variables and the rooms are the domains. The agents in a room form a coalition and empty rooms are ignored. We set a constraint for each room equal to $-v(s)$ where s is the set of agents that choose that room, or 0 if no agents choose it. We then have a constraint optimization problem where we are trying to the set of values which minimizes the sum of the constraint violations, thus maximizing the sum of the valuations.

Note that this is a degenerate case of the constraint optimization problem in that all the n constraints are over all agents. Most constraint problems exhibit some degree of locality in that constraints are only over a small subset of the variables. Having all constraints be over all variables makes this problem harder than average. Thus this reduction is likely to be only of theoretical interest.

4.2 Coalition Formation

The **coalition formation** problem, as studied in multiagent systems, extends the basic characteristic form game in an effort to make it a better match for real world problems. A coalition formation problem consists of three steps. COALITION FORMATION

1. Agents generate values for the $v(\cdot)$ function.
2. Agents solve the characteristic form game by finding a suitable set of coalitions.
3. Agents distribute the payments from these coalitions to themselves in a suitable manner.

Steps 2 and 3 can be thought of as part of the traditional characteristic form game. The coalition formation definition simply chooses to split the problem of finding a suitable outcome $\vec{u}$ into two parts: finding the coalitions and then dividing the payments. The split mirrors the requirements of many application domains. Step 1 is completely new. It is there because in many domains it is computationally expensive to determine the value of $v(S)$ for a given S. For example, if the agents are trying to form groups that solve particular tasks then calculating $v(S)$ requires them to determine out how well they can perform the task as a group, which requires considering how all their different skills can be brought to bear and, in some common scenarios, requires the development of a full plan—an exponential problem. The approaches at solving step 1 are thus generally dependent on the domain and do not generalize well.

Exercises

4.1 Give an example problem in which agents using equal excess and reporting their own E values will want to lie about their own E values.

4.2 Find the set of core solutions and the Shapley value of the grand coalition for the following game:

S	$v(S)$
(1)	1
(2)	2
(3)	3
(12)	5
(13)	4
(23)	5
(123)	5

4.3 Say you have robots which live in a 2-dimensional grid and each one has a strength given by a number in the set $\{1, 2, 3, 4, 5\}$. There are boxes in this world, each one of which must be moved to a specified destination. The speed with which a set of robots S can move a box is given by $1-1/\sum_{i \in S} i$.strength.

- Formulate this problem as a characteristic form game and provide a $v(S)$ definition.
- Find a good algorithm for calculating the optimal coalition structure.

4.4 Modify the FIND-COALITION algorithm from figure 4.7 so that instead of the agents broadcasting their values they move around randomly in a two-dimensional space. After each time step the set of agents in a tile checks if they form a maximal coalition, that is, there is no other coalitions that gives one of the agents a higher $v_i(S)$ value. If so, they form that coalition and leave the game while the rest keep moving. Implement this algorithm in NetLogo and check how long it takes to find a solution.

4.5 We can extend the problem of coalition formation and make it more realistic by defining the value function over a set of possible agent abilities and then giving the agents sets of abilities (Yokoo et al., 2005). We then face the possibility of an agent pretending to be multiple agents, each with a different ability. Why would an agent do this? Give an example when an agent benefits from this technique.

Another problem might be agents that fail to mention to others that they have certain skills. Why would an agent do this? Give an example when an agent benefits from this technique.