Game Theory

Chapter 5
Cooperative Games

Instructor: Chih-Wen Chang

Contents

5.1 Coalitions and Characteristic Functions Problems

Coalitions and Characteristic Functions

- We consider a game in which the players may choose to cooperate by forming coalitions.
  - There are $n > 1$ players numbered 1, 2,…, n.
  - The set of all the players: N={1,2,..., n}
  - A **coalition** is any subset $S \subset N$, or numbered collection of the players.
  - Since there are 2^n possible subsets of N, there are 2^n possible coalitions.

- Coalitions form in order to benefit every member of the coalition so that all members might receive more than they could individually on their own.

Coalitions and Characteristic Functions

- In this section we try to determine a **fair allocation** of the benefits of cooperation among the players to each member of a coalition.
  – A major problem in cooperative game theory is to precisely define what **fair** means.

- First we need to quantify the benefits of a coalition through the use of a real-valued function, called the **characteristic function.**
  – The characteristic function of a coalition $S \subset N$ is the **largest** guaranteed payoff to the coalition.

Definition 5.1.1

- **Definition 5.1.1**
  - *Let 2^N denote the set of all possible coalitions for the players N. If S =* {i} *is a coalition containing the single member i, we simply denote S by i.*

    Any function $v : 2^N \to \mathbb{R}$ *satisfying*

$$v(\emptyset) = 0 \ \ and \ \ v(N) \geq \sum_{i=1}^{n} v(i)$$

is a characteristic function (of an n-person cooperative game).

Definition 5.1.1 (cont'd)

- In other words, the only condition placed on a characteristic function is that the benefit of the empty coalition be zero and the benefit of the **grand coalition** *N,* consisting of all the players, be at least the sum of the benefits of the individual players if no coalitions form.
  - This means that every one pulling together should do better than each player on his or her own.
  - With that much flexibility, games may have more than one characteristic function.

EXAMPLE 5.1

- Let's start with some simple characteristic function examples.

1. Suppose that there is a factory with n workers each doing the same task. If each worker earns the same amount b dollars, then we can take the characteristic function to be $v(S) = b|S|$, where $|S|$ is the number of workers in S. Clearly, $v(\emptyset) = b|\emptyset| = 0$, and $v(N) = b|N| = bn = b\sum_{i=1}^{n} v(i)$.

EXAMPLE 5.1 (cont'd)

2. Suppose that the owner of a car, labeled player 1, offers it for sale for \$M. There are two customers interested in the car. Customer C, labeled player 2, values the car at c and customer D, labeled player 3, values it at d. Assume that the price is nonnegotiable. This means that if $M > c$ and $M > d$, then no deal will be made. We will assume then that $M < \min\{c, d\}$, and, for definiteness we may assume $M < c \leq d$. The set of possible coalitions are $2^N \equiv \{123,\ 12,\ 13,\ 23,\ 1,\ 2,\ 3,\ \emptyset\}$. For simplicity we are dropping the braces in the notation for any individual coalition.

It requires a seller and a buyer to reach a deal. Therefore, we may define the characteristic function as follows:

$$v(123) = d, \quad v(1) = M, \quad v(\emptyset) = 0$$
$$v(13) = d, \quad v(12) = c, \quad v(23) = 0,$$
$$v(2) = v(3) = 0.$$

EXAMPLE 5.1 (cont'd)

Why? Well, $v(123) = d$ because the car will be sold for d, $v(1) = M$ because the car is worth M to player 1, $v(13) = d$ because player 1 will sell the car to player 3 for $d > M$, $v(12) = c$ because the car will be sold to player 2 for $c > M$, and so on. The reader can easily check that v is a characteristic function.

3. A customer wants to buy a bolt and a nut for the bolt. There are three players but player 1 owns the bolt and players 2 and 3 each own a nut. A bolt together with a nut is worth 5. We could define a characteristic function for this game as

$$v(123) = 5, \; v(12) = v(13) = 5, \quad v(1) = v(2) = v(3) = 0, \quad \text{and} \quad v(\emptyset) = 0.$$

In contrast to the car problem $v(1) = 0$ because a bolt without a nut is worthless to player 1.

EXAMPLE 5.1 (cont'd)

4. A small research drug company, labeled 1, has developed a drug. It does not have the resources to get FDA (Food and Drug Administration) approval or to market the drug, so it considers selling the rights to the drug to a big drug company. Drug companies 2 and 3 are interested in buying the rights but only if both companies are involved in order to spread the risks. Suppose that the research drug company wants \$1 billion, but will take \$1 million if only one of the two big drug companies are involved. The profit to a participating drug company 2 or 3 is \$5 billion, which they split. Here is a possible characteristic function with units in billions:

$$v(1) = v(2) = v(3) = 0, v(12) = 0.1, v(13) = 0.1, v(23) = 0, v(123) = 5,$$

because any coalition which doesn't include player 1 will be worth nothing.

EXAMPLE 5.1 (cont'd)

5. A **simple game** is one in which $v(S) = 1$ or $v(S) = 0$ for all coalitions S. A coalition with $v(S) = 1$ is called a **winning coalition** and one with $v(S) = 0$ is a **losing coalition**. For example, if we take $v(S) = 1$ if $|S| > n/2$ and $v(S) = 0$ otherwise, we have a simple game that is a model of majority voting. If a coalition contains more than half of the players, it has the majority of votes and is a winning coalition.

6. In any bimatrix (A, B) nonzero sum game we may obtain a characteristic function by taking $v(1) = value(A)$, $v(2) = value(B^T)$, and $v(12) =$sum of largest payoff pair in (A, B). Checking that this is a characteristic function is skipped. The next example works one out.

EXAMPLE 5.2

- In this example we will construct a characteristic function for a version of the prisoner's dilemma game in which we assumed that there was no cooperation. Now we will assume that the players may cooperate and negotiate.

- Prisoner's dilemma bimatrix

$$\begin{bmatrix} (8,8) & (0,10) \\ (10,0) & (2,2) \end{bmatrix}$$

  – Here N = {1, 2} and the possible coalitions are 2^N = $\{\emptyset, 1, 2, 12\}$

EXAMPLE 5.2 (cont'd)

– If the players do not form a coalition, they are playing the nonzero sum noncooperative game. Each player can guarantee only that they receive their **safety level.**

For player I $A = \begin{bmatrix} 8 & 0 \\ 10 & 2 \end{bmatrix}$ $value(A) = 2$

For player II $B^T = \begin{bmatrix} 8 & 0 \\ 10 & 2 \end{bmatrix}$ $value(B^T) = 2$

– Thus we could define v(l) = v(2) = 2 as the characteristic function for single member coalitions.

EXAMPLE 5.2 (cont'd)

- If the players cooperate and form the coalition $S = \{12\}$, the Figure 5.1, which is generated by Maple, shows what is going on.
  - The parallelogram is the boundary of the set of all possible payoffs to the two players when they use all possible mixed strategies.
  - You can see that without cooperation the profits are each at the lower left vertex point (2, 2).
  - Any point in the parallelogram is attainable with some suitable selection of mixed strategies if the players cooperate. Consequently, the maximum benefit to cooperation for both players results in the payoff pair at vertex point (8,8), and so we set $v(12) = 16$ as the **maximum sum of the** benefits awarded to each player.

EXAMPLE 5.2 (cont'd)

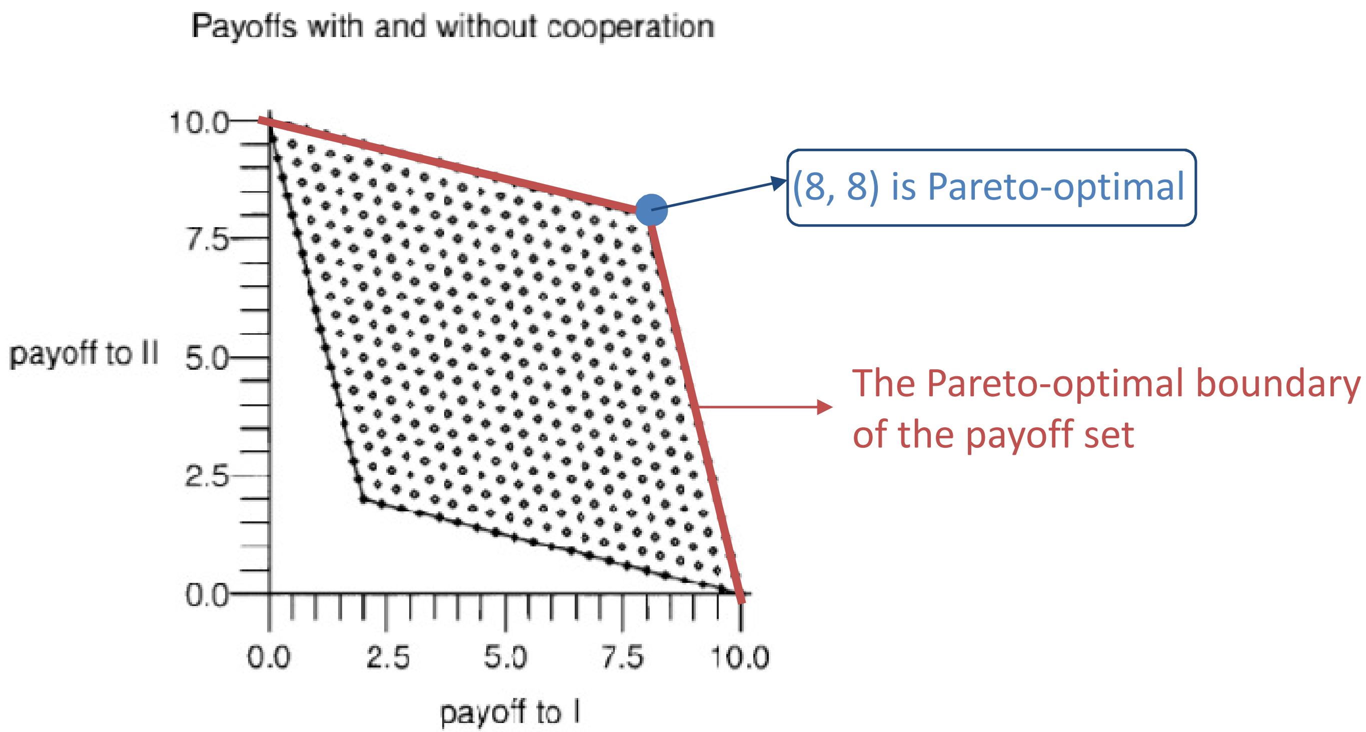


Figure 5.1 Payoff to player I versus payoff to player II.

EXAMPLE 5.3

- Here is a much more complicated but systematic way to create a characteristic function given any n-person, noncooperative, nonzero sum game.
  - The idea is to create a two-person zero sum game in which any given coalition is played against a pure opposing coalition consisting of everybody else. The two players are the coalition *S* versus all the other players, which is also a coalition *N* - *S.*
  - The characteristic function will be the value of the game associated with each coalition *S.*

EXAMPLE 5.3 (cont'd)

- Suppose that we have a three-player nonzero sum game with the following matrices:

3 plays A		player 2	
		A	B
player 1	A	$(1, 1, 0)$	$(4, -2, 2)$
	B	$(1, 2, -1)$	$(3, 1, -1)$

3 plays B		player 2	
		A	B
player 1	A	$(-3, 1, 2)$	$(0, 1, 1)$
	B	$(2, 0, -1)$	$(2, 1, -1)$

EXAMPLE 5.3 (cont'd)

- We need to consider all of the zero sum games which would consist of the two-player coalitions versus each player, and the converse, which will switch the roles from maximizer to minimizer and vice versa.
  - For example, one such possible game is $S = \{12\}$ versus $N — S = 3$, in which player $S = \{12\}$ is the row player and player 3 is the column player. We also have to consider the game 3 versus {12}, in which player 3 is the row player and coalition {12} is the column player.

EXAMPLE 5.3 (cont'd)

- **1. Play** *S* = {12} **versus** {3}.
  - players 1 and 2 team up against player 3.
  - In the game {12} versus {3}

12 versus 3		player 3	
		A	B
player 12	AA	2	−2
	AB	2	1
	BA	3	2
	BB	4	3

  - For example, if 1 plays A and 2 plays A and 3 plays B, the payoffs in the nonzero sum game are (-3,1, 2) and so the payoff to player 12 is -3 + 1 = -2, the **sum of the payoff to player 1 and player 2,** which is our coalition**.**

EXAMPLE 5.3 (cont'd)

- Now we calculate the value of the zero sum two-person game with this matrix to get the *value*(12 vs. 3) = 3 and we write *v*(12) = 3.

– In the game {3} versus {12}

3 versus 12		12			
		AA	AB	BA	BB
3	A	0	−2	−1	−1
	B	2	1	−1	−1

- We now want to know the maximum possible payoff to player 3 assuming that the coalition {12} is actively working against player 3.
- The value of this game is -1. Consequently, in the game {3} versus {12} we would get *v(3) =* - 1.
- Observe that the game matrix for 3 versus 12 is **not** the transpose of the game matrix for 12 versus 3.

EXAMPLE 5.3 (cont'd)

- 2. **Play** *S* = {13} **versus** {2}.
  - In the game {13} versus {2}

13 versus 2		player 2 A	B
player 13	AA	1	6
	AB	−1	1
	BA	0	2
	BB	1	1

- We see that the value of this game is 1 so that $v(13) = 1$.

EXAMPLE 5.3 (cont'd)

– In the game {2} versus {13}

2 versus 13		13			
		AA	AB	BA	BB
2	A	1	1	2	0
	B	−2	1	1	1

- The value of this game is 1/4, and so *v*(2) = 1/4.

- we summarize that the characteristic function for this three-person game is

$$v(1) = 1,\ v(2) = \frac{1}{4},\ v(3) = -1,$$
$$v(12) = 3,\ v(13) = 1,\ v(23) = 1,$$
$$v(123) = 4,\ v(\emptyset) = 0.$$

EXAMPLE 5.3 (cont'd)

- The value is obtained from the pure strategies: 3 plays A, 1 plays A, and 2 plays B with payoffs (4, -2,2). Summing these payoffs for all the players gives $v(123) = 4$.
  - This is the most the players can get if they form a grand coalition, and they can get this only if all the players cooperate.

EXAMPLE 5.3 (cont'd)

- The central question in cooperative game theory is how to allocate the reward of 4 to the three players. In this example, player 2 contributes a payoff of -2 to the grand coalition, so should player 2 get an equal share of the 4? On the other hand, the 4 can only be obtained if player 2 agrees to play strategy B, so player 2 does have to be induced to do this. What would be a **fair allocation?**

The General Formula for The Characteristic Function

Remark.

– There is a general formula for the characteristic function obtained by converting an n-person nonzero sum game to a cooperative game. Given any coalition $S \subset N$, the characteristic function is

$$v(S) = \max_{X \in X_S} \min_{Y \in Y_{N-S}} \sum_{i \in S} E_i(X, Y) = \min_{Y \in Y_{N-S}} \max_{X \in X_S} \sum_{i \in S} E_i(X, Y),$$

- X_S is the set of mixed strategies for the coalition *S*
- Y_{N-S} is the set of mixed strategies for the coalition *N* - *S*
- $E_i(X, Y)$ is the expected payoff to player $i \in S$
- $\sum_{i \in S} E_i(X, Y)$ is the total payoff for each player in $i \in S$ and represents the payoff to the coalition *S.*

Remarks on Characteristic Functions

- 1. A very desirable property of a characteristic function is that it satisfy

$$v(S \cup T) \geq v(S) + v(T) \quad \text{for all } S, T \subset N, S \cap T = \emptyset$$

This is called **superadditivity.**

  - It says that the benefits of the larger consolidated coalition *S* U *T* of the two separate coalitions *S*. *T* must be at least the total benefits of the individual coalitions *S* and *T*.
  - Many results on cooperative games do not need superadditivity, but we will take it as an **axiom that our characteristic functions in all that follows must be superadditive.**
  - With the assumption of superadditivity, the players have the incentive to form and join the grand coalition *N.*

Remarks on Characteristic Functions (cont'd)

- 2. A game is **inessential** if and only if $v(N) = \sum_{i=1}^{n} v(i)$. An **essential** game therefore is one with $v(N) > \sum_{i=1}^{n} v(i)$.

- 3. Any game with $v(S \cup T) = v(S) + v(T)$ for all $S, T \subset N, S \cap T = \emptyset$, is called an **additive** game. A game is inessential if and only if it is additive.
  - The word **inessential** implies that these games are not important.

Remarks on Characteristic Functions (cont'd)

- To see why a characteristic function for an inessential game must be additive, we simply write down the definitions. In fact let $S, T \subset N, S \cap T = \emptyset$.
  - Then

$$\begin{aligned} v(N) &= \sum_{i=1}^{n} v(i) && \text{(inessential game)} \\ &= \sum_{i \in S} v(i) + \sum_{i \in T} v(i) + \sum_{i \in N-(S \cup T)} v(i) \\ &\leq v(S) + v(T) + v(N - (S \cup T)) && \text{(superadditivity)} \\ &\leq v(S \cup T) + v(N - (S \cup T)) && \text{(superadditivity)} \\ &\leq v(N) && \text{(superadditivity again).} \end{aligned}$$

  - Since we now have equality throughout

$$v(S) + v(T) + v(N - (S \cup T)) = v(S \cup T) + v(N - (S \cup T)),$$

and so $v(S) + v(T) = v(S \cup T)$.

Definition 5.1.2

- We need a basic definition regarding the allocation of rewards to each player. Recall that $v(N)$ represents the reward available if all players cooperate.

- **Definition 5.1.2**

Let x_i be the share of the value of $v(N)$ received by player $i = 1, 2, \ldots, n$. A vector $\vec{x} = (x_1, \ldots, x_n)$ is an **imputation** *if*

- $x_i \geq v(i)$ **(individual rationality)**
- $\sum_{i=1}^{n} x_i = v(N)$ **(group rationality)**

The imputation $\vec{x}$ is also called a **payoff vector** *or an* **allocation***, and we will use these words interchangeably.*

Definition 5.1.2 (cont'd)

- **Remarks**
  1. It is possible for x_i to be a negative number! That allows us to model coalition members that do not benefit and may be a detriment to a coalition.
  2. Individual rationality means that the share received by player *i* should be at least what he could get on his own. Each player must be individually rational.
  3. Group rationality means that the total rewards allocated to each individual in the grand coalition should **equal** the total rewards available by cooperation.

Definition 5.1.2 (cont'd)

4. Any inessential game, $v(N) = \sum_{i=1}^{n} v(i)$, has one and only one imputation and it is $\vec{x} = (v(1), \ldots, v(n))$.

   These games are uninteresting because there is no incentive for any of the players to form any sort of coalition and there is no wiggle room in finding a better allocation.

(0,1) Normalization

- We begin by presenting a way to transform a given characteristic function for a cooperative game to one which is frequently easier to work with. It is called the **(0,1) normalization of the original game.**
  - The normalized game will result in a characteristic function with $v(i) = 0$, $v(N) = 1$.
  - In addition, any two games may be compared by comparing their normalized characteristic functions. If they are the same, the two games are said to be **strategically equivalent.**

Lemma 5.1.3

- **Lemma 5.1.3**

Any essential game with characteristic function v has a $(0,1)$ normalization with characteristic function v; that is, given the characteristic function $v(\cdot)$ there is a unique characteristic function $v'(\cdot)$ that satisfies $v'(N) = 1, v'(i) = 0, 1 \le i \le n$, and $v'(S) = cv(S) + \sum_{i \in S} a_i$ for some constants $c > 0, a_1, \ldots, a_n$. The constants are given by

$$c \equiv \frac{1}{v(N) - \sum_{i=1}^{n} v(i)} \quad \textit{and} \quad a_i \equiv -c\, v(i), \quad i = 1, 2, \ldots, n.$$

Proof. Consider the $n+1$ system of equations for constants $c, a_i, 1 \le i \le n$, given by

$$cv(i) + a_i = 0, \quad 1 \le i \le n,$$

$$cv(N) + \sum_{i=1}^{n} a_i = 1.$$

Lemma 5.1.3 (cont'd)

If we add up the first n equations, we get $c\sum v(i) + \sum a_i = 0$. Subtracting this from the second equation results in

$$c[v(N) - \sum v(i)] = 1,$$

and we can solve for $c > 0$ because the game is essential (so $v(N) > \sum v(i)$). Now that we have c, we set $a_i = -c\, v(i)$. Solving this system, we get

$$c = \frac{1}{v(N) - \sum v(i)} > 0, \qquad a_i = -\frac{v(i)}{v(N) - \sum_i v(i)}.$$

Then, for any coalition $S \subset N$ define the characteristic function

$$v'(S) = cv(S) + \sum_{i \in S} a_i \quad \text{for all } S \subset N.$$

The equations we started with give immediately that $v'(N) = 1$ and $v'(i) = 0, i = 1, 2, \ldots, n.$ □

How Does Normalizing Affect Imputations?

- If we have an imputation for an unnormalized game, what does it become for the normalized game? Conversely, if we have an imputation for the normalized game, how do we get the imputation for the original game?
  - The set of imputations for the original game is

$$X = \{\vec{x} = (x_1, \ldots, x_n) \mid x_i \geq v(i), \sum_{i=1}^{n} x_i = v(N)\}.$$

  - For the normalized game, indeed, for any game with *v*(*i*) = 0, *v*(*N*) = 1, the set of all possible imputations is given by

$$X' = \{\vec{x}' = (x'_1, \ldots, x'_n) \mid x'_i \geq 0, \sum_{i=1}^{n} x'_i = 1\}.$$

How Does Normalizing Affect Imputations? (cont'd)

- If $\vec{x}' = (x'_1, \ldots, x'_n) \in X'$ is an imputation for *v'* then the imputation for *v* becomes

$$\vec{x} = (x_1, \ldots, x_n) \in X \text{ where } x_i = \frac{(x'_i - a_i)}{c},$$

$$c = \frac{1}{v(N) - \sum_{i=1}^{n} v(i)} \quad \text{and} \quad a_i = -c\, v(i), \quad i = 1, 2, \ldots, n.$$

- Conversely, if $\vec{x} = (x_1, \ldots, x_n) \in X$ is an imputation for the original game, then $\vec{x}' = (x'_1, \ldots, x'_n)$ is the imputation for the normalized game, where $x'_i = cx_i + a_i, i = 1, 2, \ldots, n$.

EXAMPLE 5.4

In the three–person nonzero sum game considered above we found the (unnormalized) characteristic function to be

$$v(1) = 1,\ v(2) = \frac{1}{4},\ v(3) = -1$$
$$v(12) = 3,\ v(13) = 1,\ v(23) = 1$$
$$v(123) = 4.$$

To normalize this game we compute

$$c = \frac{1}{v(N) - \sum_{i=1}^{3} v(i)} = \frac{1}{4 - \frac{1}{4}} = \frac{4}{15}, \text{ and } a_i = -\frac{4}{15} v(i).$$

EXAMPLE 5.4 (cont'd)

So

$$a_1 = \frac{4}{15}, a_2 = -\frac{1}{15}, \text{ and } a_3 = \frac{4}{15}.$$

Then the normalized characteristic function by v' is calculated as

$$\begin{aligned}
v'(i) &= \frac{4}{15}\, v(i) + a_i = 0 \\
v'(12) &= \frac{4}{15}\, v(12) + a_1 + a_2 = \frac{7}{15} \\
v'(13) &= \frac{4}{15}\, v(13) + a_1 + a_3 = \frac{4}{15} \\
v'(23) &= \frac{4}{15}\, v(23) + a_2 + a_3 = \frac{7}{15} \\
v'(123) &= \frac{4}{15}\, v(123) + a_1 + a_2 + a_3 = 1.
\end{aligned}$$

Definition 5.1.4

- Let *X* denote the set of imputations $\vec{x}$. We look for an allocation $\vec{x} \in X$ as a solution to the game.

- **Definition 5.1.4**

  The **reasonable allocation set** *of a cooperative game is a set of imputations $R \subset X$ given by*

$$R \equiv \{\vec{x} \in X \mid x_i \leq \max_{T \in \Pi^i} \{v(T) - v(T - i)\}, i = 1, 2, \ldots, n\},$$

where Π^i is the set of all coalitions for which player i is a member. So, if $T \in \Pi^i$, then $i \in T \subset N$, and $T - i$ denotes the coalition T without the player i.

Definition 5.1.4 (cont'd)

- In other words, the reasonable set is the set of imputations so that the amount allocated to each player is no greater than the maximum benefit that the player brings to any coalition of which the player is a member
- The difference $v(T) - v(T - i)$ is the measure of the rewards for coalition T due to player i.
- The reasonable set gives us a first way to reduce the size of X and try to focus in on a solution.

- If the reasonable set has only one element, which is extremely unlikely for most games, then that is our solution.

Definition 5.1.5

- If there are many elements in *R,* we need to cut it down further. In fact, we need to cut it down to the **core** imputations, or even further. Here is the definition.

- **Definition 5.1.5**

Let $S \subset N$ be a coalition and let $\vec{x} \in X$. The **excess** *of coalition $S \subset N$ for imputation $\vec{x} \in R$ is defined by*

$$e(S, x) = v(S) - \sum_{i \in S} x_i.$$

The **core of the game** *is*

$$C(0) = \{\vec{x} \in X \mid e(S, \vec{x}) \leq 0, \ \forall S \subset N\} = \{\vec{x} \in X \mid v(S) \leq \sum_{i \in S} x_i, \ \forall S \subset N\}.$$

Definition 5.1.5 (cont'd)

The ε**-core**, *for* $-\infty < \varepsilon < +\infty$, *is*

$$C(\varepsilon) = \{\vec{x} \in X \mid e(S, \vec{x}) \leq \varepsilon,\ \forall S \subset N, S \neq N, S \neq \emptyset\}$$

Let $\varepsilon^1 \in (-\infty, \infty)$ *be the smallest* ε *for which* $C(\varepsilon) \neq \emptyset$. *The* **least core** , *labeled* X^1, *is* $C(\varepsilon^1)$. *It is possible for* ε^1 *to be positive, negative, or zero.*

- The grand coalition is excluded in the requirements for $C(\varepsilon)$ because if *N* were an eligible coalition, then $e(N, \vec{x}) = 0 \leq \varepsilon$, and it would force ε to be nonnegative. That would put too strict a requirement on ε in order for $C(\varepsilon)$ to be nonempty.

Definition 5.1.5 (cont'd)

– We will use the notation that for a given imputation $\vec{x} = (x_1, \ldots, x_n)$ and a given coalition $S \subset N$

$$\vec{x}(S) = \sum_{i \in S} x_i,$$

the total amount allocated to coalition *S.*

Definition 5.1.5 (cont'd)

- **Remark**
  1. The excess function $e(S, \vec{x})$ is a **measure of dissatisfaction** of a particular coalition S with the allocation $\vec{x}$. Consequently, $\vec{x}$ is in the core if all coalitions are satisfied with $\vec{x}$. If the core has only one allocation, that is our solution.
     - If $e(S, \vec{x}) > 0$, this would say that the maximum possible benefits of joining the coalition S are greater than the total allocation to the members of S using the imputation $\vec{x}$. But then the members of S would not be very happy with $\vec{x}$ and would want to change to a better allocation.
     - In that sense, if $\vec{x} \in C(0)$, then $e(S, \vec{x}) \leq 0$ for every coalition S, and there would be no incentive for any coalition to try to use a different imputation. An imputation is in the core of a game if it is acceptable to all coalitions.

Definition 5.1.5 (cont'd)

2. Likewise, if $\vec{x} \in C(\varepsilon)$, then the measure of dissatisfaction of a coalition with $\vec{x}$ is limited to ε. The size of *£* determines the measure of dissatisfaction because $e(S, \vec{x}) \leq \varepsilon$.

3. It is possible for the core of the game $C(0)$ to be empty, but there will always be some $\varepsilon \in (-\infty, \infty)$ so that $C(\varepsilon) \neq \emptyset$. The least core uses the smallest such ε . If the smallest $C(0) = \emptyset$.

4. It should be clear, since $C(\varepsilon)$ is just a set of inequalities, that as ε increases, $C(\varepsilon)$ gets bigger, and as ε decreases, $C(\varepsilon)$ gets smaller.
   - The idea is that we should shrink (or expand if necessary) $C(\varepsilon)$ by adjusting ε until we get one and only one imputation in it, if possible.
   - $\varepsilon < \varepsilon' \implies C(\varepsilon) \subset C(\varepsilon')$

Definition 5.1.5 (cont'd)

5. We will see shortly that $C(0) \subset R$ every allocation in the core is always in the reasonable set.

6. The definition of **solution** for a cooperative game we are going to use in this section is that an imputation should be a fair allocation if it is the allocation which minimizes the maximum dissatisfaction for all coalitions.

EXAMPLE 5.5

Let's give an example of a calculation of $C(0)$. Take the three-person game $N = \{1, 2, 3\}$, with characteristic function

$$v(1) = 1,\ v(2) = 2,\ v(3) = 3,$$
$$v(23) = 6, v(13) = 5, v(12) = 4, v(\emptyset) = 0, v(N) = 8.$$

The excess functions for a given imputation $\vec{x} = (x_1, x_2, x_3) \in C(0)$ must satisfy

$$e(1, \vec{x}) = 1 - x_1 \leq 0,\ e(2, \vec{x}) = 2 - x_2 \leq 0,\ e(3, \vec{x}) = 3 - x_3 \leq 0$$
$$e(12, \vec{x}) = 4 - x_1 - x_2 \leq 0,\ e(13, \vec{x}) = 5 - x_1 - x_3 \leq 0,$$
$$e(23, \vec{x}) = 6 - x_2 - x_3 \leq 0,$$

and we must have $x_1 + x_2 + x_3 = 8$. These inequalities imply that $x_1 \geq 1, x_2 \geq 2, x_3 \geq 3$, and

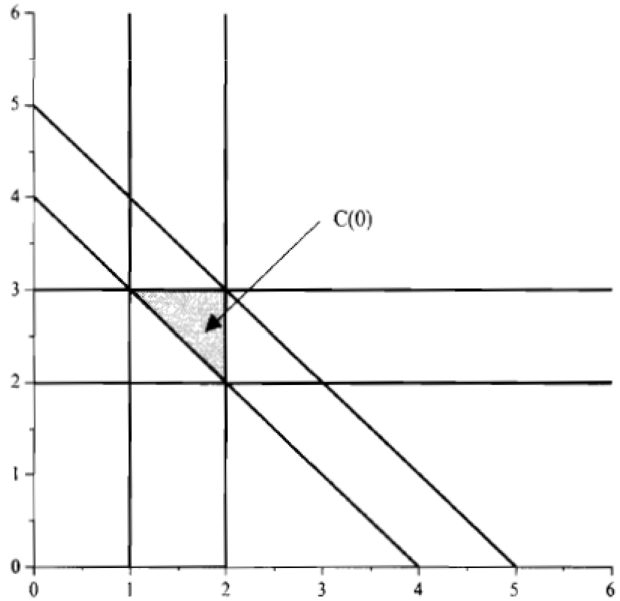
$$x_1 + x_2 \geq 4, x_1 + x_3 \geq 5, x_2 + x_3 \geq 6.$$

EXAMPLE 5.5 (cont'd)

If we use some algebra and the substitution $x_3 = 8 - x_1 - x_2$ to solve these inequalities, we see that

$$C(0) = \{(x_1, x_2, 8 - x_1 - x_2) \mid 1 \leq x_1 \leq 2, 2 \leq x_2 \leq 3, 4 \leq x_1 + x_2 \leq 5\}.$$

If we plot this region in the (x_1, x_2) plane, we get the following Maple–generated diagram



EXAMPLE 5.5 (cont'd)

In a similar way it can be shown that the smallest ε for which $C(\varepsilon) \neq \emptyset$ is $\varepsilon = \varepsilon^1 = -\frac{1}{3}$. In fact, the least core is the single imputation $C\left(-\frac{1}{3}\right) = \left\{\left(\frac{5}{3}, \frac{8}{3}, \frac{11}{3}\right)\right\}$. Indeed, the imputations in $C(\varepsilon)$ must satisfy $e(S, \vec{x}) \leq \varepsilon$ for all coalitions $S \subsetneq N$. Written out, these inequalities become

$$\begin{aligned} &1 - \varepsilon \leq x_1 \leq 2 + \varepsilon,\ 2 - \varepsilon \leq x_2 \leq 3 + \varepsilon, \\ &4 - \varepsilon \leq x_1 + x_2 \leq 5 + \varepsilon, \end{aligned}$$

where we have eliminated $x_3 = 8 - x_1 - x_2$. Adding the inequalities involving only x_1, x_2 we see that $4 - \varepsilon \leq x_1 + x_2 \leq 5 + 2\varepsilon$, which implies that $\varepsilon \geq -\frac{1}{3}$. You can check that this is the smallest ε for which $C(\varepsilon) \neq \emptyset$. With $\varepsilon = -\frac{1}{3}$, it follows that $x_1 + x_2 = \frac{13}{3}$ and $x_1 \leq \frac{5}{3},\ x_2 \leq \frac{8}{3}$. Then,

$$\left(\frac{5}{3} - x_1\right) + \left(\frac{8}{3} - x_2\right) = 0,$$

which implies that $x_1 = \frac{5}{3},\ x_2 = \frac{8}{3}$ because two nonnegative terms adding to zero must each be zero. This is one technique for finding ε^1 and $C(\varepsilon^1)$.

Lemma 5.1.6

- **Lemma 5.1.6**

  The core must be a subset of the reasonable set $C(0) \subset R$.

Proof. We may assume the game is in normalized form because we can always transform it to one that is and then work with that one. So $v(N) = 1, v(i) = 0, i = 1, \ldots, n$. Let $\vec{x} \in C(0)$. If $\vec{x} \notin R$ there is some player j such that

$$x_j > \max_{T \in \Pi^j} v(T) - v(T - j).$$

This means that for every $T \subset N$ with $j \in T$, $x_j > v(T) - v(T - j)$, and so the amount allocated to player j is larger than the amount of her benefit to any coalition. Take $T = N$. Then

$$x_j > v(N) - v(N - j) = 1 - v(N - j).$$

But then, $v(N - j) > 1 - x_j = \sum_{i \neq j} x_i$, and so $e(N - j, \vec{x}) > 0$, which means $\vec{x} \notin C(0)$. □

EXAMPLE 5.6

- In this example we will normalize the given characteristic function, find the reasonable set, and find the core of the game. Finally, we will find the least core and then find the unnormalized imputation.
  - We have the characteristic function in the three–player game from Example 5.3:

    $$\bar{v}(1) = 1, \bar{v}(2) = \frac{1}{4}, \bar{v}(3) = -1, \bar{v}(12) = 3, \bar{v}(13) = 1, \bar{v}(23) = 1, \bar{v}(123) = 4.$$
  - This is an essential game that we normalized in Example 5.4 to obtain the characteristic function that we will use:

    $$v(i) = 0,\ v(123) = 1,\ v(12) = \frac{7}{15},\ v(13) = \frac{4}{15},\ v(23) = \frac{7}{15}.$$

    The normalization constants are $c = \frac{4}{15}$, and $a_1 = -\frac{4}{15}, a_2 = -\frac{1}{15}$, and $a_3 = \frac{4}{15}$.

EXAMPLE 5.6 (cont'd)

- The set of imputations is

$$X = \{\vec{x} = (x_1, x_2, x_3) \mid x_i \geq 0, \sum_{i=1}^{3} x_i = 1\}.$$

  - The reasonable set is easy to find:

$$R = \{\vec{x} = (x_1, x_2, x_3) \in X \mid x_i \leq \max_{T \in \Pi^i} \{v(T) - v(T - i)\}, i = 1, 2, 3\}$$
$$= \{(x_1, x_2, 1 - x_1 - x_2) \mid x_1 \leq \frac{8}{15}, x_2 \leq \frac{11}{15}, \frac{7}{15} \leq x_1 + x_2 \leq 1\}.$$

  - For example, let's consider

$$x_1 \leq \max_{T \in \Pi^1} v(T) - v(T - 1).$$

EXAMPLE 5.6 (cont'd)

- The coalitions containing player 1 are $\{1, 12, 13, 123\}$, so we are calculating the maximum of

$$v(1) - v(\emptyset) = 0, \quad v(12) - v(2) = \frac{7}{15}, \quad v(13) - v(3) = \frac{4}{15},$$
$$v(123) - v(23) = 1 - \frac{7}{15} = \frac{8}{15}.$$

- Hence $0 \le x_1 \le \frac{8}{15}$. Similarly, $0 \le x_2 \le \frac{11}{15}$. We could also show $0 \le x_3 \le \frac{8}{15}$, but this isn't good enough because we can't ignore $x_1 + x_2 + x_3 = 1$. That is where we use

$$0 \le 1 - x_1 - x_2 = x_3 \le \frac{8}{15} \implies \frac{7}{15} \le x_1 + x_2 \le 1.$$

EXAMPLE 5.6 (cont'd)

– Another benefit of replacing x_3 is that now we can draw the reasonable set in (x_1, x_2) space. Figure 5.2 below is a plot of R.

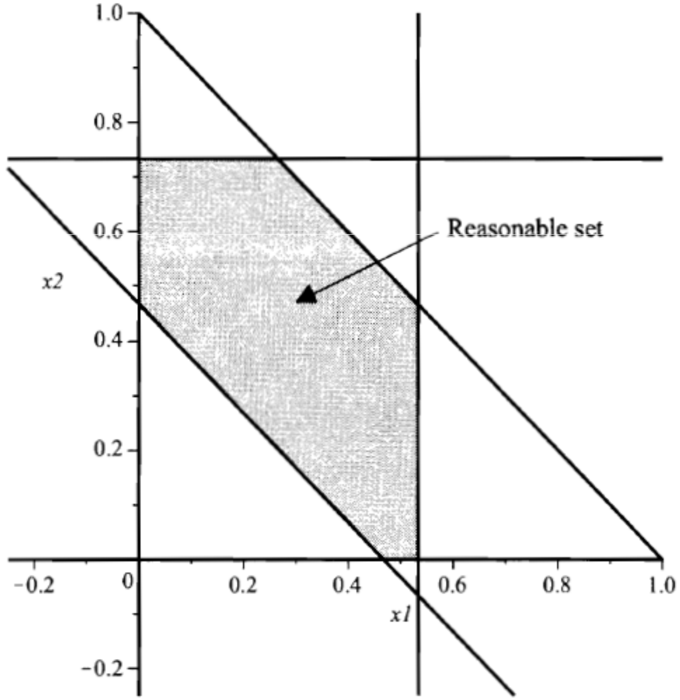


Figure 5.2 The set of reasonable imputations.

EXAMPLE 5.6 (cont'd)

– Figure 5.2 was generated with the simple Maple commands

```
> with(plots):with(plottools):
> inequal( { x<=8/15,y<=11/15,x+y >=7/15, x+y<=1,x>=0,y>=0},
             x=-.25..1, y=-.25..1,
    optionsfeasible=(color=red),
    optionsopen=(color=blue,thickness=2),
    optionsclosed=(color=gray, thickness=2),
    optionsexcluded=(color=white),labels=[`x[1]`,`x[2]`] );
```

EXAMPLE 5.6 (cont'd)

- We would like to find next the point (or points) in the reasonable set which is acceptable to all coalitions.
  - That is the core of the game:

$$
\begin{aligned}
C(0) &= \{\vec{x} \in X \mid e(S,x) \le 0,\ \forall S \subset N, S \ne N, S \ne \emptyset\} \\
&= \{x_1 \ge 0, x_2 \ge 0, \frac{7}{15} - x_1 - x_2 \le 0, -\frac{11}{15} + x_2 \le 0, -\frac{8}{15} + x_1 \le 0, \\
&\qquad x_1 + x_2 \le 1\}.
\end{aligned}
$$

Unfortunately, this gives us exactly the same set as the reasonable set, $C(0) = R$ in this example, and that is too big a set.

EXAMPLE 5.6 (cont'd)

– Now let's calculate the ε-core for any $\varepsilon \in (-\infty, \infty)$. The ε-core is, by definition

$$\begin{aligned} C(\varepsilon) &= \{\vec{x} \in X \mid e(S, \vec{x}) \le \varepsilon,\ \forall S \subset N, S \ne N, S \ne \emptyset\} \\ &= \{x_1 \ge 0, x_2 \ge 0, \tfrac{7}{15} - x_1 - x_2 \le \varepsilon, -\tfrac{11}{15} + x_2 \le \varepsilon, -\tfrac{8}{15} + x_1 \le \varepsilon, \\ &\qquad -x_1 \le \varepsilon, -x_2 \le \varepsilon, -1 + x_1 + x_2 \le \varepsilon\}. \end{aligned}$$

We have used the fact that $x_1 + x_2 + x_3 = 1$ to substitute $x_3 = 1 - x_1 - x_2$.

EXAMPLE 5.6 (cont'd)

– By working with the inequalities in $C(\varepsilon)$, we can find the least core X^1. We verify that the smallest ε so that $C(\varepsilon) \neq \emptyset$ is $\varepsilon^1 = -\frac{4}{15}$. The procedure is to add the inequality for x_2 with the one for x_1 and then use the first inequality:

$$-\frac{11}{15} + x_2 - \frac{8}{15} + x_1 = -\frac{19}{15} + x_1 + x_2 \leq 2\varepsilon,$$

but $x_1 + x_2 \geq \frac{7}{15} - \varepsilon$, so that

$$-\frac{12}{15} - \varepsilon = -\frac{19}{15} + \frac{7}{15} - \varepsilon \leq -\frac{19}{15} + x_1 + x_2 \leq 2\varepsilon$$

which can be satisfied if and only if $\varepsilon \geq -\frac{4}{15} = \varepsilon^1$.

EXAMPLE 5.6 (cont'd)

– If we replace ε by $\varepsilon^1 = -\frac{4}{15}$, the **least core** is the set

$$C(\varepsilon^1) = X^1 = \{(x_1, x_2, 1 - x_1 - x_2) \mid \frac{4}{15} \geq x_1 \geq 0, \frac{7}{15} \geq x_2 \geq 0, x_1 + x_2 = \frac{11}{15}\}$$

$$= \left\{\left(\frac{4}{15}, \frac{7}{15}, \frac{4}{15}\right)\right\}.$$

→ $(x_1 = \frac{4}{15}, x_2 = \frac{7}{15}, x_3 = \frac{4}{15})$

– If we want the imputation of the original unnormalized game, we use $\bar{x}_i = (x_i - a_i)/c$ and obtain

$$\bar{x}_1 = \frac{\frac{4}{15} + \frac{4}{15}}{\frac{4}{15}} = 2, \ \bar{x}_2 = \frac{\frac{7}{15} + \frac{1}{15}}{\frac{4}{15}} = 2, \ \ \bar{x}_3 = \frac{\frac{4}{15} - \frac{4}{15}}{\frac{4}{15}} = 0.$$

EXAMPLE 5.7

- We have already argued that the core $C(0)$ should consist of the good imputations and so would be considered the solution of our game.
  - If in fact $C(0)$ contained exactly one point, then that would be true. Unfortunately, the core may contain many points, as in the last example, or may even be empty.

- Here is an example of a game with an empty core.

EXAMPLE 5.7 (cont'd)

Suppose that the characteristic function of a three-player game is given by

$$v(123) = 1 = v(12) = v(13) = v(23) \quad \text{and} \quad v(1) = v(2) = v(3) = 0.$$

Since this is already in normalized form, the set of imputations is

$$X = \{\vec{x} = (x_1, x_2, x_3) \mid x_i \geq 0, \sum_{i=1}^{3} x_i = 1\}.$$

To calculate the reasonable set R, we need to find

$$x_i \leq \max_{T \in \Pi^i} \{v(T) - v(T - i)\}, i = 1, 2, 3.$$

Starting with $\Pi^1 = \{1, 12, 13, 123\}$, we calculate

$$v(1) - v(\emptyset) = 0, \; v(12) - v(2) = 1, v(13) - v(3) = 1, v(123) - v(23) = 0,$$

so $x_1 \leq \max\{0, 1, 1, 0\} = 1$. This is true for x_2 as well as x_3. So all we get from this is $R = X$, all the imputations are reasonable.

EXAMPLE 5.7 (cont'd)

Next we have

$$C(0) = \{\vec{x} \in X \mid v(S) \le \sum_{i \in S} x_i, \ \forall S \subsetneq N\}.$$

If $\vec{x} \in C(0)$, we calculate

$$e(i, \vec{x}) = v(i) - x_i = -x_i \le 0, \ e(12, \vec{x}) = 1 - (x_1 + x_2) \le 0$$

and, in likewise fashion

$$e(13, \vec{x}) = 1 - (x_1 + x_3) \le 0, \ e(23, \vec{x}) = 1 - (x_2 + x_3) \le 0.$$

The set of inequalities we have to solve are

$$x_1 + x_2 \ge 1, \ x_1 + x_3 \ge 1, \ x_2 + x_3 \ge 1, \ x_1 + x_2 + x_3 = 1, \ x_i \ge 0.$$

But clearly there is no $\vec{x} \in X$ that can satisfy these inequalities, because it is impossible to have three positive numbers, any two of which have sum at least 1, which can add up to 1, so $C(0) = \emptyset$.

EXAMPLE 5.8

- In this example we will determine a necessary and sufficient condition for any cooperative game with three players to have a nonempty core.
  - We take $N = \{1, 2, 3\}$ and a characteristic function in normalized form

$$v(i) = v(\emptyset) = 0, \quad i = 1, 2, 3, \quad v(123) = 1,$$
$$v(12) = a_{12}, \quad v(13) = a_{13}, \quad v(23) = a_{23}.$$

Of course, we have $0 \le a_{ij} \le 1$. We can state the proposition.

Proposition 5.1.7 *For the three-person cooperative game with normalized characteristic function v we have $C(0) \neq \emptyset$ if and only if*

$$a_{12} + a_{13} + a_{23} \le 2.$$

EXAMPLE 5.8 (cont'd)

Proof. We have

$$C(0) = \{(x_1, x_2, 1 - x_1 - x_2) \mid x_i \geq 0, a_{12} \leq x_1 + x_2,$$
$$a_{13} \leq x_1 + (1 - x_1 - x_2) = 1 - x_2, \text{ and } a_{23} \leq 1 - x_1\}.$$

So, $x_1 + x_2 \geq a_{12}$, $x_2 \leq 1 - a_{13}$, and $x_1 \leq 1 - a_{23}$. Adding the last two inequalities says $x_1 + x_2 \leq 2 - a_{23} - a_{13}$ so that with the first inequality

$a_{12} \leq 2 - a_{13} - a_{23}$. Consequently, if $C(0) \neq \emptyset$, it must be true that $a_{12} + a_{13} + a_{23} \leq 2$.

For the other side, if $a_{12} + a_{13} + a_{23} \leq 2$, we define the imputation

$$\vec{x} = (x_1, x_2, x_3)$$
$$= \left(\frac{1 - 2a_{23} + a_{13} + a_{12}}{3}, \frac{1 + a_{23} - 2a_{13} + a_{12}}{3}, \frac{1 + a_{23} + a_{13} - 2a_{12}}{3} \right).$$

EXAMPLE 5.8 (cont'd)

Then $x_1 + x_2 + x_3 = 1 = v(123)$. Furthermore

$$\begin{aligned} v(23) - x_2 - x_3 &= a_{23} - x_2 - x_3 = a_{23} - x_2 - (1 - x_1 - x_2) \\ &= a_{23} - 1 + x_1 \\ &= a_{23} - 1 + \frac{1 - 2a_{23} + a_{13} + a_{12}}{3} \\ &= \frac{a_{23} + a_{13} + a_{12} - 2}{3} \le 0. \end{aligned}$$

Similarly, $v(12) - x_1 - x_2 \le 0$ and $v(13) - x_1 - x_3 \le 0$. Hence $\vec{x} \in C(0)$ and so $C(0) \neq \emptyset$. $\square$

EXAMPLE 5.8 (cont'd)

- **Remark: An Automated Way to Determine Whether** C(0) = 0.
  - Maple can give us a simple way of determining whether the core is empty. Consider the linear program:

$$\text{Minimize } z = x_1 + \cdots + x_n$$
$$\text{subject to } v(S) \le \sum_{i \in S} x_i \text{ for every } S \subsetneq N.$$

It is not hard to check that $C(0)$ is not empty if and only if the linear program has a minimum, say, z^*, and $z^* \le v(N)$. If the game is normalized, then we need $z^* \le 1$. When this condition is not satisfied, $C(0) = \emptyset$.

EXAMPLE 5.8 (cont'd)

– For instance, in the last example the commands would be

```
> with(simplex):
> obj:=x+y+z;
> cnsts:={1-x-z<=0,1-y-z<=0,1-x-y<=0};
> minimize(obj,cnsts,NONNEGATIVE);
> assign(%);
> obj;
```

Maple gives the output $\{x = \frac{1}{2}, y = \frac{1}{2}, z = \frac{1}{2}\}$ as the allocation and $obj = \frac{3}{2}$ as the sum of the allocation components. Since this is a game in which the allocation components must sum to 1, $v(N) = 1$, we see that the core must be empty.

5.1.1 Finding the Least Core Problems

Definition 5.1.8

- One way to describe the fact that one imputation is better than another is the concept of domination.

- **Definition 5.1.8**

  If we have two imputations $\vec{x} \in X, \vec{y} \in X$, and a nonempty coalition $S \subset N$, then $\vec{x}$ **dominates** *$\vec{y}$ (for the coalition S) if $x_i > y_i$ for all members $i \in S$, and $\vec{x}(S) = \sum_{i \in S} x_i \leq v(S)$.*

  – If $\vec{x}$ dominates $\vec{y}$ for the coalition S, then members of S prefer the allocation $\vec{x}$ to the allocation $\vec{y}$, because they get more $x_i > y_i$, for each $i \in S$, and the coalition *S* can actually achieve the allocation because $v(S) \geq \sum_{i \in S} x_i$.

Theorem 5.1.9

- **Theorem 5.1.9**

The core of a game is the set of all undominated imputations for the game; that is,

$$C(0) = \{\vec{x} \in X \mid \textit{there is no } \vec{z} \in X \ \textit{ and } \ S \subset N \textit{ such that}$$
$$z_i > x_i, \forall\, i \in S, \ \textit{ and } \sum_{i \in S} z_i \leq v(S)\}.$$

Proof. Call the right hand side the set B. We have to show $C(0) \subset B$ and $B \subset C(0)$.

We may assume that the game is in $(0, 1)$ normalized form.

Let $\vec{x} \in C(0)$ and suppose $\vec{x} \notin B$. Since $\vec{x} \notin B$ that means that $\vec{x}$ must be dominated by another imputation for at least one nonempty coalition $S \subset N$; that is, there is $\vec{y} \in X$ and $S \subset N$ such that $y_i > x_i$ for all $i \in S$ and $v(S) \geq \sum_{i \in S} y_i$. Summing on $i \in S$ this shows

$$v(S) \geq \sum_{i \in S} y_i > \sum_{i \in S} x_i \Longrightarrow e(S, \vec{x}) > 0,$$

contradicting the fact that $\vec{x} \in C(0)$. Therefore $C(0) \subset B$.

Theorem 5.1.9 (cont'd)

Now let $\vec{x} \in B$. If $\vec{x} \notin C(0)$, there is a nonempty coalition $S \subset N$ so that $e(S, \vec{x}) = v(S) - \sum_{i \in S} x_i > 0$. Let

$$\varepsilon = v(S) - \sum_{i \in S} x_i > 0 \text{ and } \alpha = 1 - v(S) \geq 0.$$

Let $s = |S|$, the number of players in S, and

$$z_i = \begin{cases} x_i + \dfrac{\varepsilon}{s} & \text{if } i \in S; \\ \dfrac{\alpha}{n-s} & \text{if } i \notin S. \end{cases}$$

We will show that $\vec{z} = (z_1, \ldots, z_n)$ is an imputation and $\vec{z}$ dominates $\vec{x}$ for the coalition S; that is, that $\vec{z}$ is a better allocation for the members of S than is $\vec{x}$.

First $z_i \geq 0$ and

$$\sum_{i=1}^{n} z_i = \sum_S x_i + \sum_S \frac{\varepsilon}{s} + \sum_{N-S} \frac{\alpha}{n-s} = \sum_S x_i + \varepsilon + \alpha = v(S) + 1 - v(S) = 1.$$

Therefore $\vec{z}$ is an imputation.

Theorem 5.1.9 (cont'd)

Next we show $\vec{z}$ is a better imputation than is $\vec{x}$ for the coalition S.If $i \in S$ $z_i = x_i + \varepsilon/s > x_i$ and $\sum_{i \in S} z_i = \sum_{i \in S} x_i + \varepsilon = v(S)$. Therefore $\vec{z}$ dominates $\vec{x}$. But this says $\vec{x} \notin B$ and that is a contradiction. Hence $B \subset C(0)$. $\square$

EXAMPLE 5.9

- Suppose that Bill has 150 sinks to give away to the whomever shows up to take them away.
  - Amy(l), Agnes(2), and Agatha(3) simultaneously show up with their trucks to take as many of the sinks as their trucks can haul.
  - Amy can haul 45, Agnes 60, and Agatha 75, for a total of 180, 30 more than the maximum of 150.

- The wrinkle in this problem is that the sinks are too heavy for any one person to load onto the trucks so they must cooperate in loading the sinks. The question is: How many sinks should be allocated to each person?

EXAMPLE 5.9 (cont'd)

– Define the characteristic function $v(S)$ as the number of sinks the coalition $S \subset N = \{1, 2, 3\}$ can load. We have $v(i) = 0, i = 1, 2, 3$, since they must cooperate to receive any sinks at all, and

$$v(12) = 105,\ v(13) = 120,\ v(23) = 135,\ v(123) = 150.$$

– The set of imputations will be $X = \{(x_1, x_2, x_3) | x_i \geq 0, \sum x_i = 150\}$.

– Let's use Maple to see if the core is nonempty:

```
> with(simplex):
> obj:=x1+x2+x3:
> cnsts:={105-x1-x2<=0,120-x1-x3<=0,135-x2-x3<=0};
> minimize(obj,cnsts,NONNEGATIVE);
> assign(%);
> obj;
```

EXAMPLE 5.9 (cont'd)

– Maple gives the output $x_1 = 45, x_2 = 60, x_3 = 75$, and $obj = x_1 + x_2 + x_3 = 180 > v(123) = 150$. So the core of this game is empty. A direct way to get this is to note that the inequalities

$$x_1 + x_2 \geq 105, x_1 + x_3 \geq 120 \text{ and } x_2 + x_3 \geq 135$$

imply that $2(x_1 + x_2 + x_3) = 2(150) = 300 \geq 360$, which is impossible.

– The next step is to calculate the least core. Begin with the definition:

$$\begin{aligned} C(\varepsilon) &= \{x \in X \mid e(S, x) \leq \varepsilon, \forall S \subsetneq N\} \\ &= \{\vec{x} \in X \mid v(S) - \sum_{i \in S} x_i \leq \varepsilon\} \\ &= \{\vec{x} \mid 105 \leq x_1 + x_2 + \varepsilon, 120 \leq x_1 + x_3 + \varepsilon, \\ &\qquad 135 \leq x_2 + x_3 + \varepsilon, -x_i \leq \varepsilon\}. \end{aligned}$$

EXAMPLE 5.9 (cont'd)

- We know that $x_1 + x_2 + x_3 = 150$ so by replacing $x_3 = 150 - x_1 - x_2$ we obtain as conditions on ε that

$$120 \leq 150 - x_2 + \varepsilon,\ 135 \leq 150 - x_1 + \varepsilon,\ 105 \leq x_1 + x_2 + \varepsilon.$$

We see that $45 \geq x_1 + x_2 - 2\varepsilon \geq 105 - 3\varepsilon$, implying that $\varepsilon \geq 20$. This is in fact the smallest $\varepsilon^1 = 20$, making $C(\varepsilon) \neq \emptyset$. Using $\varepsilon^1 = 20$, we calculate

$$C(20) = \{(x_1 = 35, x_2 = 50, x_3 = 65)\}.$$

Hence the fair allocation is to let Amy have 35 sinks, Agnes 50, and Agatha 65 sinks, and they all cooperate.

Lemma 5.1.10

- **Lemma 5.1.10**

Let

$$\varepsilon^1 = \min_{\vec{x} \in X} \max_{S \subsetneq N} e(S, \vec{x}).$$

Then the least core $X^1 = C(\varepsilon^1) \neq \emptyset$ *and if* $\varepsilon > \varepsilon^1$, *then* $C(\varepsilon^1) \subsetneq C(\varepsilon)$.

Proof. Since the set of imputations is compact (=closed and bounded) and $\vec{x} \mapsto \max_S e(S, \vec{x})$ is at least lower semicontinuous, there is an allocation $\vec{x}_0$ so that the minimum in the definition of ε^1 is achieved, namely, $\varepsilon^1 = \max_S e(S, \vec{x}_0) \geq e(S, \vec{x}_0), \forall S \subsetneq N$. This is the very definition of $\vec{x}_0 \in C(\varepsilon^1)$ and so $C(\varepsilon^1) \neq \emptyset$.

On the other hand, if we have a smaller $\varepsilon < \varepsilon^1 = \min_{\vec{x}} \max_{S \subsetneq N}$, then for every allocation $\vec{x} \in X$, we have $\varepsilon < \max_S e(S, \vec{x})$. So, for any allocation there is at least one coalition $S \subsetneq N$ for which $\varepsilon < e(S, \vec{x})$. This means that for this ε, no matter which allocation is given, $\vec{x} \notin C(\varepsilon)$. Thus, $C(\varepsilon) = \emptyset$. As a result, ε^1 is the smallest ε so that $C(\varepsilon) \neq \emptyset$. □

Lemma 5.1.10 (cont'd)

Remarks.
These remarks summarize the ideas behind the use of the least core.

1. For a given grand allocation $\vec{x}$, the coalition S_0 that most objects to $\vec{x}$ is the coalition giving the largest excess and so satisfies

$$e(S_0, \vec{x}) = \max_{S \subsetneq N} e(S, \vec{x}).$$

   For each fixed coalition S, the allocation giving the minimum dissatisfaction is

$$e(S, \vec{x}_0) = \min_{\vec{x} \in X} e(S, \vec{x}).$$

2. The value of ε giving the least ε-core is

$$\varepsilon^1 \equiv \min_{\vec{x} \in X} \max_{S \subsetneq N} e(S, \vec{x}),$$

   and this is the smallest level of dissatisfaction.

Lemma 5.1.10 (cont'd)

3. If $\varepsilon^1 = \min_{\vec{x}} \max_{S \subsetneq N} < 0$, then there is an allocation $\vec{x}^*$ so that $\max_S e(S, \vec{x}^*) < 0$. That means that $e(S, \vec{x}^*) < 0$ for every coalition $S \subsetneq N$. Every coalition is satisfied with $\vec{x}^*$ because $v(S) < \vec{x}^*(S)$, so that every coalition is allocated at least its maximum value.

   If $\varepsilon^1 = \min_{\vec{x}} \max_{S \subsetneq N} > 0$, then for every allocation $\vec{x} \in X$, $\max_S e(S, \vec{x}) > 0$. Consequently, there is at least one coalition S so that $e(S, \vec{x}) = v(S) - \vec{x}(S) > 0$. For any allocation, there is at least one coalition that will not be happy with it.

Lemma 5.1.10 (cont'd)

4. The excess function $e(S, \vec{x})$ is a measure of dissatisfaction of S with the imputation $\vec{x}$. It makes sense that the best imputation would minimize the largest dissatisfaction over all the coalitions. This leads us to one possible definition of a solution for the n-person cooperative game. An allocation $\vec{x}^* \in X$ is a solution to the cooperative game if

$$\varepsilon^1 = \min_{\vec{x} \in X} \max_S e(S, x) = \max_S e(S, x^*),$$

so that $\vec{x}^*$ minimizes the maximum excess for any coalition S. When there is only one such allocation $\vec{x}^*$, it is the fair allocation. The problem is that there may be more than one element in the least core, then we still have a problem as to how to choose among them.

Maple Calculation of the Least Core

Remark: Maple Calculation of the Least Core. The point of calculating the ε-core is that the core is not a sufficient set to ultimately solve the problem in the case when the core $C(0)$ is (1) empty or (2) consists of more than one point. In case (2) the issue, of course, is which point should be chosen as the fair allocation. The ε-core seeks to address this issue by shrinking the core at the same rate from each side of the boundary until we reach a single point. We can use Maple to do this.

The calculation of the least core is equivalent to the linear programming problem

Minimize z

subject to

$$v(S) - \vec{x}(S) = v(S) - \sum_{i \in S} x_i \leq z, \text{ for all } S \subsetneq N.$$

The characteristic function need not be normalized. So all we really need to do is to formulate the game using characteristic functions, write down the constraints, and plug them into Maple. The result will be the smallest $z = \varepsilon^1$ that makes $C(\varepsilon^1) \neq \emptyset$, as well as an imputation which provides the minimum.

Maple Calculation of the Least Core (cont'd)

- The Maple commands used to solve this are very simple:

```
> with(simplex):
> cnsts:={-x1<=z,-x2<=z,-x3<=z,2-x1-x2<=z,1-x2-x3<=z,-x1-x3<=z,
        x1+x2+x3=\frac {5}{2}};
> minimize(z,cnsts);
```

- Maple produces the output

$$x_1 = \frac{5}{4},\ x_2 = 1,\ x_3 = \frac{1}{4},\ z = -\frac{1}{4}.$$

Hence the smallest $\varepsilon^1 = z$ for which the ε-core is nonempty is $\varepsilon^1 = -\frac{1}{4}$. Now, Maple also gives us the allocation $\vec{x} = \left(\frac{5}{4}, 1, \frac{1}{4}\right)$ which will be in $C\left(-\frac{1}{4}\right)$, but we don't know if that is the **only point** in $C\left(-\frac{1}{4}\right)$.

Maple Calculation of the Least Core (cont'd)

- With Maple we can graph the core and the set $C\left(-\frac{1}{4}\right)$ with the following commands:

```
> cnsts:={-x1<=z,-x2<=z,-(5/2-x1-x2)<=z,2-x1-x2<=z,
            1-x2-(5/2-x1-x2)<=z,-x1-(5/2-x1-x2)<=z};
> Core:=subs(z=0,cnsts);
> with(plots):
> inequal(Core,x1=0..2,x2=0..3,optionsfeasible=(color=red),
>     optionsopen=(color=blue,thickness=2),
>     optionsclosed=(color=green, thickness=3),
>     optionsexcluded=(color=yellow));
> ECore:=subs(z=-1/4,cnsts);
> inequal(ECore,x1=0..2,x2=0..3,optionsfeasible=(color=red),
>     optionsopen=(color=blue,thickness=2),
>     optionsclosed=(color=green, thickness=3),
>     optionsexcluded=(color=yellow));
```

Maple Calculation of the Least Core (cont'd)

Figure 5.3 shows the core $C(0)$.

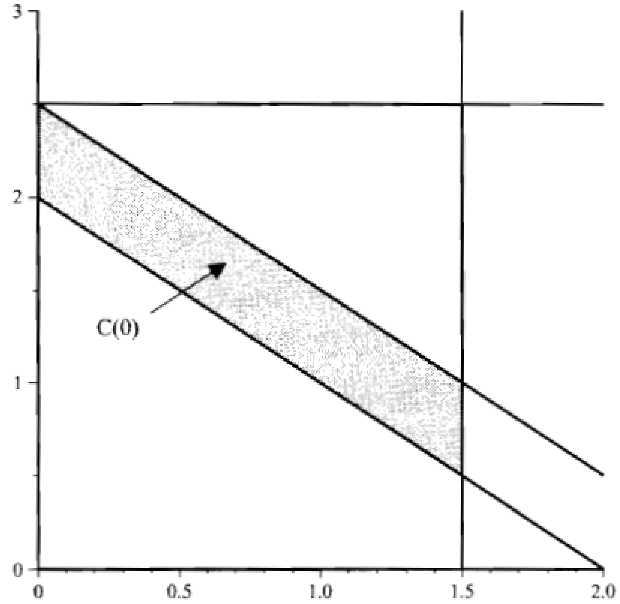


Figure 5.3 Graph of $C(0)$.

Maple Calculation of the Least Core (cont'd)

- You can even see how the core shrinks to the c-core using an animation:

```
> animate(inequal,[cnsts, x1=0..2,x2=0..3,
                        optionsfeasible=(color=red),
                        optionsopen=(color=blue,thickness=2),
                        optionsclosed=(color=green,thickness=3),
                        optionsexcluded=(color=white)],
                        z=-1..0,frames=50);
```

Figure 5.4 results from the animation at $z = -0.18367$ with the dark region constituting the core $C(-0.18367)$. You will see that at $z = -\frac{1}{4}$ the dark region becomes the line segment. Hence $C\left(-\frac{1}{4}\right)$ is certainly not empty, but it is also not just one point.

Maple Calculation of the Least Core (cont’d)

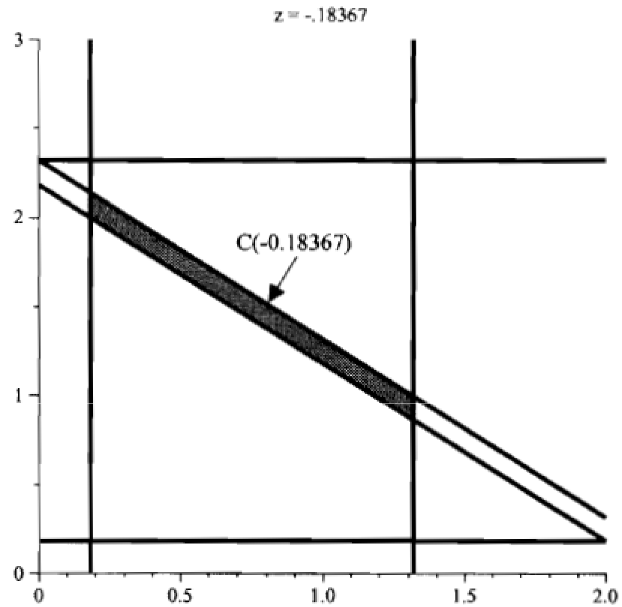


Figure 5.4 The shrinking of the core frozen at $C(-0.18367)$.

So we have solved any cooperative game if the least core contains exactly one point. But when $C(\varepsilon^1) = X^1$ has more than one point, we still have a problem, and that leads us in the next section to the **nucleolus**.

5.2 The Nucleolus Problems

THE NUCLEOLUS

- The core $C(0)$ might be empty, but we can find an ε so that . $C(\varepsilon)$ is not empty. We can fix the **empty** problem. Even if $C(0)$ is not empty, it may contain more than one point and again we can use $C(\varepsilon)$ to **maybe** shrink the core down to one point or, if $C(0) = \emptyset$, to expand the core until we get it nonempty. The problem is what happens when the least core $C(\varepsilon)$ itself has too many points.

- In the previous section we saw that we should shrink $C(0)$ to $C(\varepsilon^1)$, so if $C(\varepsilon^1)$ has more than one allocation, why not shrink that also? No reason at all.

EXAMPLE 5.10

– Let us take the normalized characteristic function for the three–player game

$$v(12) = \frac{4}{5}, v(13) = \frac{2}{5}, v(23) = \frac{1}{5} \text{ and } v(123) = 1, v(i) = 0, \quad i = 1, 2, 3.$$

Step 1: Calculate the least core. We have the ε-core

$$\begin{aligned} C(\varepsilon) &= \{(x_1, x_2, x_3) \in X \mid e(S, x) \leq \varepsilon, \forall S \subsetneq N\} \\ &= \{(x_1, x_2, 1 - x_1 - x_2) \mid -\varepsilon \leq x_1 \leq \frac{4}{5} + \varepsilon, \\ &\qquad -\varepsilon \leq x_2 \leq \frac{3}{5} + \varepsilon, \ \frac{4}{5} - \varepsilon \leq x_1 + x_2 \leq 1 + \varepsilon\}. \end{aligned}$$

EXAMPLE 5.10

- We then calculate that the smallest ε for which $C(\varepsilon) \neq \emptyset$ is $\varepsilon^1 = -\frac{1}{10}$, and then

$$C\left(\varepsilon^1 = -\frac{1}{10}\right) = \{(x_1, x_2, 1 - x_1 - x_2) \mid x_1 \in \left[\frac{2}{5}, \frac{7}{10}\right],$$
$$x_2 \in \left[\frac{1}{5}, \frac{1}{2}\right], \quad x_1 + x_2 = \frac{9}{10}\}.$$

- This is a line segment in the (x_1, x_2) plane as we see in Figure 5.5, which is obtained from the Maple animation shrinking the core down to the line frozen at $z = -0.07$.

EXAMPLE 5.10 (cont'd)

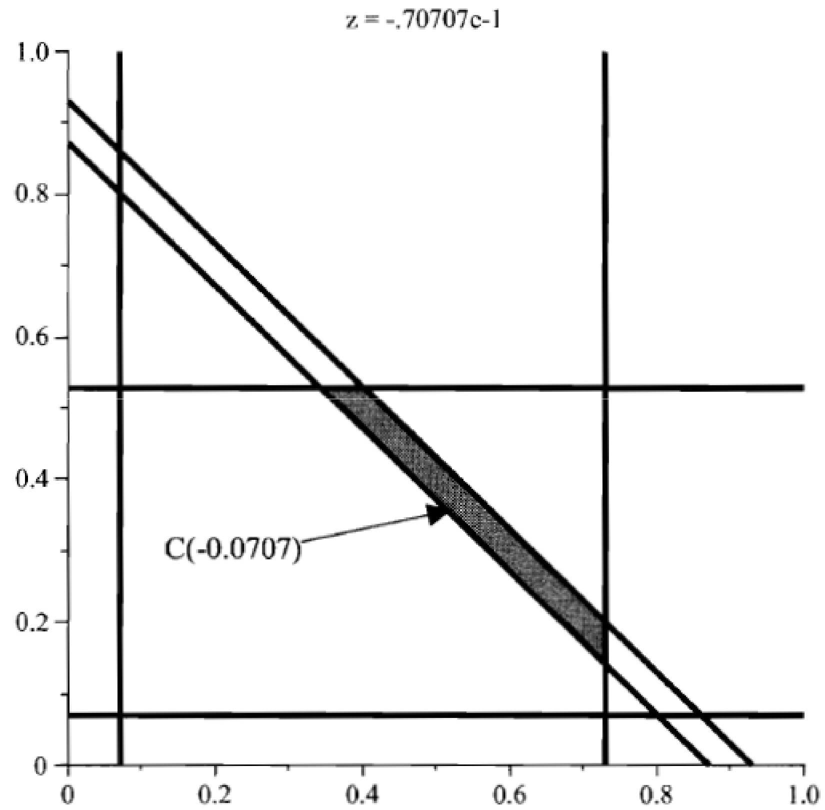


Figure 5.5 $C(-0.07)$: shrinking down to a line segment.

EXAMPLE 5.10 (cont'd)

- So we have the problem that the least core does not have only one imputation that we would be able to call our solution. What is the fair allocation now? We must shrink the line down somehow.
- **Step 2: Calculate the next least core.** The idea is, **restricted to the allocations in the first least core**, minimize the maximum excesses over all the allocations in the least core. So we must take allocations with $\vec{x} = (x_1, x_2, x_3)$, with $\frac{2}{5} \leq x_1 \leq \frac{7}{10}$, $\frac{1}{5} \leq x_2 \leq \frac{1}{2}$, and $x_1 + x_2 = \frac{9}{10}$. This last equality then requires that $x_3 = 1 - x_1 - x_2 = \frac{1}{10}$.
- If we take any allocation $\vec{x} \in C(\varepsilon^1)$, we want to calculate the excesses for each coalition:

$$\begin{array}{ll} e(1, \vec{x}) = -x_1 & e(2, \vec{x}) = -x_2 \\ e(13, \vec{x}) = x_2 - \frac{3}{5} & e(23, \vec{x}) = x_1 - \frac{4}{5} \\ e(12, \vec{x}) = \frac{4}{5} - x_1 - x_2 = -\frac{1}{10} & e(3, \vec{x}) = -x_3 = -\frac{1}{10} \end{array}$$

EXAMPLE 5.10 (cont'd)

– If we take any allocation $\vec{x} \in C(\varepsilon^1)$, we want to calculate the excesses for each coalition:

$$
\begin{array}{ll}
e(1, \vec{x}) = -x_1 & e(2, \vec{x}) = -x_2 \\
e(13, \vec{x}) = x_2 - \frac{3}{5} & e(23, \vec{x}) = x_1 - \frac{4}{5} \\
e(12, \vec{x}) = \frac{4}{5} - x_1 - x_2 = -\frac{1}{10} & e(3, \vec{x}) = -x_3 = -\frac{1}{10}
\end{array}
$$

Since $\vec{x} \in C(\varepsilon^1)$, we know that these are all $\leq -\frac{1}{10}$. Observe that the excesses $e(12, \vec{x}) = e(3, \vec{x}) = -\frac{1}{10}$ do not depend on the allocation $\vec{x}$ as long as it is in $C(\varepsilon^1)$. But then, there is nothing we can do about those coalitions by changing the allocation. Those coalitions will always have an excess of $-\frac{1}{10}$ as long as the imputations are in $C(\varepsilon^1)$, and they cannot be reduced. Therefore, we may eliminate those coalitions from further consideration.

EXAMPLE 5.10 (cont'd)

– Now we set

$$\Sigma^1 \equiv \{S \subsetneq N \mid e(S, \vec{x}) < \varepsilon^1, \text{ for some } \vec{x} \in C(\varepsilon^1)\}.$$

This is a set of coalitions with excesses for some imputation smaller than ε^1.

– These coalitions can use some imputation that gives a **better** allocation for them, as long as the allocations used are also in $C\left(-\frac{1}{10}\right)$. For our example, we get

$$\Sigma^1 = \{1, 2, 13, 23\}.$$

The coalitions $\{12\}$ and $\{3\}$ are out because the excesses of those coalitions cannot be dropped below $-\frac{1}{10}$ no matter what allocation we use in $C(-\frac{1}{10})$. Their level of dissatisfaction cannot be dropped any further.

EXAMPLE 5.10 (cont'd)

– Now pick any allocation in $C(-\frac{1}{10})$ and calculate the smallest level of dissatisfaction for the coalitions in Σ^1 :

$$\varepsilon^2 \equiv \min_{\vec{x} \in X^1} \max_{S \in \Sigma^1} e(S, \vec{x}).$$

The number ε^2 is then the smallest maximum excess over all allocations in $C(-\frac{1}{10})$. It is defined just as is ε^1 except we restrict to the coalitions that can have their dissatisfaction reduced. Finally, set

$$X^2 \equiv \{\vec{x} \in X^1 = C(\varepsilon^1) \mid e(S, \vec{x}) \leq \varepsilon^2, \ \forall S \in \Sigma^1\}.$$

- The set X^2 is the subset of allocations from X^1 that are preferred by the coalitions in Σ^1.

EXAMPLE 5.10 (cont'd)

- In our example we now use allocations $\vec{x} \in X^1$ so that $x_1 + x_2 = \frac{9}{10}$, $\frac{2}{5} \leq x_1 \leq \frac{7}{10}$, and $\frac{1}{5} \leq x_2 \leq \frac{1}{2}$. The next least core is

$$\begin{aligned} C(\varepsilon^2) \equiv X^2 &= \{\vec{x} \in X^1 \mid e(1,\vec{x}) \leq \varepsilon^2, e(2,\vec{x}) \leq \varepsilon^2, \\ &\qquad e(13,\vec{x}) \leq \varepsilon^2, e(23,\vec{x}) \leq \varepsilon^2\} \\ &= \{\vec{x} \in X^1 \mid -x_1 \leq \varepsilon^2, -x_2 \leq \varepsilon^2, \\ &\qquad x_2 - \frac{3}{5} \leq \varepsilon^2, x_1 - \frac{4}{5} \leq \varepsilon^2\}. \end{aligned}$$

- We need to find the smallest ε^2 for which X^2 is nonempty. We do this by hand as follows. Since $x_1 + x_2 = \frac{9}{10}$, we get rid of $x_2 = \frac{9}{10} - x_1$. Then

$$\begin{aligned} &-x_1 \leq \varepsilon^2 \quad -x_2 = x_1 - \frac{9}{10} \leq \varepsilon^2 \implies -\varepsilon^2 - \frac{9}{10} \leq \varepsilon^2 \implies \varepsilon^2 \geq -\frac{9}{20} \\ &\frac{9}{10} - x_1 - \frac{3}{5} \leq \varepsilon^2, \;\; x_1 - \frac{4}{5} \leq \varepsilon^2 \implies -\frac{5}{10} - \varepsilon^2 \leq \varepsilon^2 \implies \varepsilon^2 \geq -\frac{5}{20}, \end{aligned}$$

and so on.

EXAMPLE 5.10 (cont'd)

– The smallest ε^2 satisfying all the requirements is then $\varepsilon^2 = -\frac{5}{20} = -\frac{1}{4}$. Next, we replace ε^2 by $-\frac{1}{4}$ in the definition of X^2 to get

$$C\left(-\frac{1}{4}\right) \equiv X^2 = \{\vec{x} \in X^1 \mid -x_1 \le -\frac{1}{4}, -x_2 \le -\frac{1}{4}, \\ x_2 - \frac{3}{5} \le -\frac{1}{4}, x_1 - \frac{4}{5} \le -\frac{1}{4}\} \\ = \{\vec{x} \in X^1 \mid \frac{1}{4} \le x_1 \le \frac{11}{20}, \ \frac{1}{4} \le x_2 \le \frac{7}{20} \\ x_1 + x_2 = 18/20.\}$$

The last equality gives us

$$0 = \left(\frac{11}{20} - x_1\right) + \left(\frac{7}{20} - x_2\right) \implies x_1 = \frac{11}{20}, \ x_2 = \frac{7}{20},$$

since both terms are nonnegative and cannot add up to zero unless they are each zero. So we have found $x_1 = \frac{11}{20}$, $x_2 = \frac{7}{20}$, and, finally, $x_3 = \frac{2}{20}$. We have our second least core

$$X^2 = \left\{\left(\frac{11}{20}, \frac{7}{20}, \frac{2}{20}\right)\right\},$$

EXAMPLE 5.10 (cont'd)

– X^2 consists of exactly one point. That is our solution to the problem. Notice that for this allocation

$$e(13, \vec{x}) = x_2 - \frac{3}{5} = \frac{7}{20} - 12/20 = -\frac{1}{4}$$
$$e(23, \vec{x}) = x_1 - \frac{4}{5} = 11/20 - \frac{4}{5} = -\frac{1}{4}$$
$$e(1, \vec{x}) = -11/20, \quad \text{and} \quad e(2, x) = -\frac{7}{20},$$

and each of these is a constant smaller than $-\frac{1}{10}$.

EXAMPLE 5.10 (cont'd)

- The most difficult part of this procedure is finding ε^1, ε^2, and so on. This is where Maple is a great help. For instance, we can find $\varepsilon^2 = -\frac{1}{4}$ very easily if we use the commands

```
> with(simplex):
> cnsts:={-x1<=z,-x2<=z,x2-3/5<=z,x1-4/5<=z,x1+x2=\frac {9}{10}};
> minimize(z,cnsts);
```

  Maple informs us that z=-1/4, x1=11/20, x2=7/20.

EXAMPLE 5.10 (cont'd)

In general, we would need to continue this procedure if X^2 also contained more than one point. Here are the sequence of steps to take in general until we get down to one point:

1. **Step 0: Initialize.** We start with the set of all possible imputations X and the coalitions excluding N and $\emptyset$:

$$X^0 \equiv X, \quad \Sigma^0 \equiv \{S \subsetneq N, S \neq \emptyset\}$$

2. **Step $k \geq 1$: Successively calculate**

   (a) The minimum of the maximum dissatisfaction

$$\varepsilon^k \equiv \min_{\vec{x} \in X^{k-1}} \max_{S \in \Sigma^{k-1}} e(S, \vec{x}).$$

   (b) The set of allocations achieving the minimax dissatisfaction

$$X^k \equiv \{\vec{x} \in X^{k-1} \mid \varepsilon^k = \min_{\vec{x} \in X^{k-1}} \max_{S \in \Sigma^{k-1}} e(S, \vec{x}) = \max_{S \in \Sigma^{k-1}} e(S, \vec{x})\}$$
$$= \{\vec{x} \in X^{k-1} \mid e(S, \vec{x}) \leq \varepsilon^k, \ \forall S \subsetneq \Sigma^{k-1}\}.$$

EXAMPLE 5.10 (cont'd)

(c) The set of coalitions achieving the minimax dissatisfaction

$$\Sigma_k = \{S \in \Sigma^{k-1} \mid e(S, \vec{x}) = \varepsilon^k, \forall\, \vec{x} \in X^k\}.$$

(d) Delete these coalitions from the previous set

$$\Sigma^k \equiv \Sigma^{k-1} - \Sigma_k.$$

3. **Step: Test if Done.** If $\Sigma^k = \emptyset$ we are done; otherwise set $k = k + 1$ and go to step (2) with the new k.

When this algorithm stops at, say, $k = m$, then X^m is the **nucleolus** of the core and will satisfy the relationships

$$X^m \subset X^{m-1} \subset \cdots \subset X^1 = C(\varepsilon^1) \subset X^0 = X.$$

Also, $\Sigma^0 \supset \Sigma^1 \supset \Sigma^2 \cdots \Sigma^{m-1} \supset \Sigma^m = \emptyset$. The allocation sets decrease down to a single point, **the nucleolus**, and the unhappiest coalitions decrease down to the empty set. The nucleolus is guaranteed to contain only one allocation $\vec{x}$, and this is the solution of the game. In fact, the following theorem can be proved.[1]

Theorem 5.2.1

- **Theorem 5.2.1**

The nucleolus algorithm stops in a finite number of steps $m < \infty$ *and for each* $k = 1, 2, \ldots, m$ *we have*

1. $-\infty < \varepsilon_k < \infty$.
2. $X^k \neq \emptyset$ *are convex, closed, and bounded.*
3. $\Sigma_k \neq \emptyset$ *for* $k = 1, 2, \ldots, m-1$.
4. $\varepsilon_{k+1} < \varepsilon_k$.

In addition, X^m *is a single point, called the* **nucleolus of the game***:*

$$Nucleolus = X^m = \bigcap_{k=1}^{m} X^k.$$

Theorem 5.2.1 (cont'd)

- The nucleolus algorithm stops when all coalitions have been eliminated, but when working this out by hand you don't have to go that far. When you see that X^k is a single point you may stop.
  - The procedure to find the nucleolus can be formulated as a sequence of linear programs that can be solved using Maple.
    - To begin, set $k = 1$ and calculate the constraint set

$$X^1 = \{\vec{x} \in X \mid e(S, \vec{x}) \le \varepsilon, \forall S \subsetneq N\}.$$

The smallest ε that makes this nonempty is ε^1, given by

$$\varepsilon^1 = \min_{\vec{x} \in X} \max_{S \in \Sigma^0} e(S, \vec{x}), \quad \Sigma^0 = \{S \mid S \subsetneq N, \emptyset\}.$$

Theorem 5.2.1 (cont'd)

- The first linear programming problem that will yield $\varepsilon^1, X^1, \Sigma^1$ is

  Minimize ε
  subject to $v(S) - \vec{x}(S) \le \varepsilon,\ \vec{x} \in X^0 = X.$

  The set of $\vec{x}$ values that provide the minimum in this problem is labeled X^1 (this is the least core). Now we take

  $$\Sigma_1 = \{S \in \Sigma^0 \mid e(S, \vec{x}) = \varepsilon^1, \forall \vec{x} \in X^1\},$$

  which is the set of coalitions that give excess ε^1 for any allocation in X^1. Getting rid of those gives us the next set of coalitions that we have to deal with, $\Sigma^1 = \Sigma^0 - \Sigma_1$.

Theorem 5.2.1 (cont'd)

- The next linear programming problem can now be formulated:

Minimize ε
subject to $v(S) - \vec{x}(S) \leq \varepsilon,\ \vec{x} \in X^1, S \in \Sigma^1 = \Sigma^0 - \Sigma_1$.

The minimum such ε is ε^2, and we set X^2 to be the set of allocations in X^1 at which $\varepsilon^2 = \max_{S \in \Sigma^1} e(S, \vec{x})$. Then

$$\Sigma_2 = \{S \in \Sigma^1 | e(S, \vec{x}) = \varepsilon^2, \forall \vec{x} \in X^2\}.$$

Set $\Sigma^2 = \Sigma^1 - \Sigma_2$ and see if this is empty. If so, we are done; if not, we continue until we get our solution.

EXAMPLE 5.11

- Three hospitals, A,B,C, want to have a proton therapy accelerator (PTA) to provide precise radiological cancer therapy. These are very expensive devices because they are subatomic particle accelarators. The hospitals can choose to build their own or build one, centrally located, PTA to which they may refer their patients.

EXAMPLE 5.11 (cont'd)

- The costs for building their own PTA are estimated at 50, 30, 50, for A,B,C, respectively. The units for these numbers are million-dollars. If A and B cooperate to build a PTA, the total cost will be 60 because of land costs for the location, coordination, and so on. If B and C cooperate, the cost will be 70; if A and C cooperate, the cost will be 110. Because the cost for cooperation between A and C is greater than what it would cost if they built their own, they would decide to build their own, so the cost is still 100 for AC cooperation. Finally, the cost to build one PTA for all three hospitals A,B,C is 105.

EXAMPLE 5.11 (cont'd)

– We reformulate the problem by looking at the amount saved by each player and for each coalition.
– The characteristic function is then

$$v(S) = \text{total cost if each } i \in S \text{ builds its own} - \text{cost if they cooperate.}$$

With A=player 1, B=player 2, C=player 3, we get

$$v(1) = v(2) = v(3) = v(13) = 0, v(12) = 20, v(23) = 10, v(123) = 25.$$

For instance, $v(123) = 50 + 30 + 50 - 105 = 25$. We are looking for the fair allocation of the savings to each hospital that we can then translate back to costs. This game is not in normalized form and need not be.

EXAMPLE 5.11 (cont'd)

– The first linear program finds the least core:

Minimize ε

subject to

$$\{-x_i \leq \varepsilon, i = 1, 2, 3,\ -(x_1 + x_3) \leq \varepsilon,\ 20 - (x_1 + x_2) \leq \varepsilon,$$
$$10 - (x_2 + x_3) \leq \varepsilon,\ x_1 + x_2 + x_3 = 25\}.$$

This gives us $\varepsilon^1 = -\frac{5}{2}$, which you should check without using Maple.Replacing ε by $\varepsilon^1 = \frac{5}{2}$ and simplifying, we see that the least core will be the set

$$X^1 = \left\{(x_1, x_2, 25 - x_1 - x_2) \mid \frac{5}{2} \leq x_1 \leq \frac{25}{2}, x_1 + x_2 = \frac{45}{2}\right\}.$$

EXAMPLE 5.11 (cont'd)

– Notice that $x_3 = 25 - \frac{45}{2} = \frac{5}{2}$. Next we have to calculate

$$\Sigma_1 = \{S \in \Sigma^0 \mid e(S, \vec{x}) = \varepsilon^1, \forall \vec{x} \in X^1.\}$$

Calculate the excesses for all the coalitions except $N, \emptyset$, assuming that the allocations are in X^1 :

$$e(1, \vec{x}) = v(1) - x_1 = -x_1, \ \ e(2, \vec{x}) = -x_2, \ \ e(3, \vec{x}) = -x_3 = -\frac{5}{2}$$

$$e(12, \vec{x}) = v(12) - x_1 - x_2 = 20 - \frac{45}{2} = -\frac{5}{2}$$

$$e(13, \vec{x}) = v(13) - x_1 - x_3 = 0 - x_1 - \frac{5}{2} = -x_1 - \frac{5}{2}$$

$$e(23, \vec{x}) = v(23) - x_2 - x_3 = 10 - x_2 - \frac{5}{2} = \frac{15}{2} - x_2$$

EXAMPLE 5.11 (cont'd)

Thus the coalitions that give $\varepsilon^1 = -\frac{5}{2}$ are $\Sigma_1 = \{12, 3\}$, and so these two coalitions are dropped from consideration in the next step. The ones left are

$$\Sigma^1 = \Sigma^0 - \{12, 3\} = \{1, 2, 13, 23\}.$$

– This is going to lead to the constraint set for the next linear program:

$$\begin{aligned}
&\text{Minimize } \varepsilon,\\
&\text{subject to } \vec{x} \in X^2,\\
&X^2 = \{\vec{x} \in X^1 \mid v(S) - \vec{x}(S) \leq \varepsilon,\ \forall S \in \Sigma^1\}\\
&\quad = \left\{\left(x_1, x_2, \frac{5}{2}\right) \mid x_1 + x_2 = \frac{45}{2},\ -x_1 \leq \varepsilon, -x_2 \leq \varepsilon\right.\\
&\qquad\qquad \left. -\left(x_1 + \frac{5}{2}\right) \leq \varepsilon,\ 10 - \left(x_2 + \frac{5}{2}\right) \leq \varepsilon\right\}.
\end{aligned}$$

EXAMPLE 5.11 (cont'd)

We get the solution of this linear program as

$$\varepsilon^2 = -\frac{15}{2},\ x_1 = \frac{15}{2}, x_2 = 15, \text{ and } x_3 = \frac{5}{2}.$$

Furthermore, it is the one and only solution, so we should be done, but we will continue with the algorithm until we end up with an empty set of coalitions.

– Calculate the excesses for all the coalitions excluding $N, \emptyset$, assuming that the allocations are in X^2 :

$$e(1, \vec{x}) = -x_1 = -\frac{15}{2}, \quad e(2, \vec{x}) = -x_2 = -15, \ e(3, \vec{x}) = -x_3 = -\frac{5}{2},$$

$$e(12, \vec{x}) = v(12) - x_1 - x_2 = 20 - \frac{45}{2} = -\frac{5}{2},$$

$$e(13, \vec{x}) = v(13) - x_1 - x_3 = 0 - x_1 - \frac{5}{2} = -\frac{15}{2} - \frac{5}{2} = -\frac{20}{2},$$

$$e(23, \vec{x}) = v(23) - x_2 - x_3 = 10 - x_2 - \frac{5}{2} = \frac{15}{2} - x_2 = \frac{15}{2} - 15 = -\frac{15}{2},$$

EXAMPLE 5.11 (cont'd)

so that $e(23, \vec{x}) = e(1, \vec{x}) = -\frac{15}{2}$. Now we can get rid of coalitions $\Sigma_2 = \{1, 3, 12, 13, 23\}$ because none of the excesses for those coalitions can be further reduced by changing the allocations. Then

$$\Sigma^2 = \Sigma^1 - \Sigma_2 = \emptyset.$$

– We are now done, having followed the algorithm all the way through. We conclude that

$$\text{Nucleolus} = \left\{ \left(\frac{15}{2}, 15, \frac{5}{2} \right) \right\}.$$

EXAMPLE 5.12

In this example we will give the Maple commands at each stage to find the nucleolus. This entire procedure can be automated but that is a programming problem.

We take the characteristic function

$$v(i) = 0, i = 1, 2, 3,\ v(12) = \frac{1}{3},\ v(13) = \frac{1}{6},\ v(23) = \frac{5}{6}, v(123) = 1,$$

and this is in normalized form. We see that $\frac{1}{3} + \frac{1}{6} + \frac{5}{6} < 2$, and so the core of the game $C(0)$ is not empty by Proposition 5.1.7. We need to find the allocation within the core which solves our problem.[2]

EXAMPLE 5.12 (cont'd)

- In this example we will give the Maple commands at each stage to find the nucleolus. This entire procedure can be automated but that is a programming problem.

- We take the characteristic function

$$v(i) = 0, i = 1,2,3,\ v(12) = \frac{1}{3},\ v(13) = \frac{1}{6},\ v(23) = \frac{5}{6}, v(123) = 1,$$

  and this is in normalized form. We see that $\frac{1}{3} + \frac{1}{6} + \frac{5}{6} < 2$, and so the core of the game C(0) is not empty by Proposition 5.1.7. We need to find the allocation within the core which solves our problem.

EXAMPLE 5.12 (cont'd)

- 1. First linear programming problem. We start with the full set of possible coalitions excluding the grand coalition *N* $\emptyset$ $\Sigma^0 = \{1, 2, 3, 12, 13, 23\}$ . . In addition, with the given characteristic function, we get the excesses

$$\begin{aligned} e(1, \vec{x}) &= -x_1,\ e(2, x) = -x_2,\ e(3, \vec{x}) = -x_3 \\ e(12, \vec{x}) &= \tfrac{1}{3} - x_1 - x_2, e(13,\ \vec{x}) = \tfrac{1}{6} - x_1 - x_3 \\ e(23, \vec{x}) &= \tfrac{5}{6} - x_2 - x_3 \end{aligned}$$

- The Maple commands that give the solution are

```
> with(simplex):v1:=0:v2:=0:v3:=0:v12:=1/3:v13:=1/6:
               v23:=5/6:v123:=1;
> cnsts:={v1-x1<=z,v2-x2<=z,v3-x3<=z,v12-(x1+x2)<=z,
          v13-(x1+x3)<=z,v23-(x2+x3)<=z,x1+x2+x3=v123};
> minimize(z,cnsts);
```

EXAMPLE 5.12 (cont'd)

- Maple gives the solution $\varepsilon^1 = z = -\frac{1}{12}, x_1 = \frac{1}{12}, x_2 = \frac{3}{4}, x_3 = \frac{1}{6}$ .So this gives the allocation $\vec{x} = \left(\frac{1}{12}, \frac{3}{4}, \frac{1}{6}\right)$.
  - But this is not necessarily the unique allo-cation and therefore the solution to our game.
  - To see if there are more allocations in X^1, substitute $z = -\frac{1}{12}$ as well as $x_3 = 1 - x_1 - x_2$ in the constraint set.

- To do that in Maple use the substitute command

```
> fcnsts:=subs(z=-1/12,x1=1-x2-x3,cnsts);
```

  - This will put the new constraint set into the variable f cnsts and gives us the output

```
fcnsts={1=1,x3 <= 7/12,-x2 <= -1/12, -x3 <= -1/12,
        -x2-x3 <= -11/12, x2+x3 <= 11/12, x2 <= 3/4}.
```

EXAMPLE 5.12 (cont'd)

- To get rid of the first equality so that we can continue, use

```
> gcnsts:=fcnsts[2..7];
```

- This puts the second through seventh elements of f cnsts into gcnsts. Now, to see if there are other solutions, we need to solve the system of inequalities in gcnsts for x_1, x_2 . Maple does that as follows:

```
> with(SolveTools:-Inequality):
> glc:=LinearMultivariateSystem(gcnsts,[x2,x3]);
```

EXAMPLE 5.12 (cont'd)

- Maple solves the system of inequalities in the sense that it reduces the inequal-ities to simplest form and gives the following output:

```
{x2 <=3/4,1/3<=x2} {x3<=-x2+11/12, -x2+11/12<=x3}.
```

  – We see that $\frac{1}{3} \leq x_2 \leq \frac{3}{4}$, $x_2 + x_3 = \frac{11}{12}$ and $x_1 = \frac{1}{12}$.

EXAMPLE 5.12 (cont'd)

- 2. To get the second linear program we first have to see which coalitions are dropped. First we assign the variables that are known from the first linear program and recalculate the constraints:

```
> assign(x1=1/12,z=-1/12);
> cnsts1:={v1-x1<=z,v2-x2<=z,v3-x3<=z,v12-(x1+x2)<=z,
           v13-(x1+x3)<=z,v23-(x2+x3)<=z,x1+x2+x3=v123};
```

- Maple gives the output:

```
cnsts1:={-x3 <= -1/12, -x2-x3<=-11/12,-x2 <=-1/12,-x2 <= -1/3,
 -1/12 <= -1/12,-x3 <=-1/6, 1/12+x2+x3=1}.
```

EXAMPLE 5.12 (cont'd)

- Getting rid of the coalitions that have excess= — y^ (indicated by the output without any *x* variables), we have the new constraint set

```
> cnsts2:={v2-x2<=z2,v3-x3<=z2,v12-(x1+x2)<=z2,
           v13-(x1+x3)<=z2,x1+x2+x3=v123};
```

- Now we solve the second linear program

```
> minimize(z2,cnsts2);
```

  – which gives

$$x_2 = \frac{13}{24}, x_3 = \frac{3}{8}, z_2 = -\frac{7}{24}.$$

EXAMPLE 5.12 (cont'd)

- At each stage we need to determine whether there is more than one solution of the linear programming problem. To do that, we have to substitute our solution for z_2 into the constraints and solve the inequalities:

```
> fcnsts2:=subs(z2=-7/24,\frac {11}{12}=x2+x3,cnsts2);
> gcnsts2:=fcnsts2[2..5] union
                    {x2+x3<=\frac {11}{12},x2+x3>=\frac {11}{12}};
> g1c2:=LinearMultivariateSystem(gcnsts2,[x2,x3]);
```

- We get

```
glc2:={x2=13/24,x3 <= x2+x3=11/12},
```

and we know now that $x_1 = \frac{1}{12}, x_2 = \frac{13}{24}$, and $x_3 = \frac{3}{8}$ because $x_2 + x_3 = \frac{11}{12}$.

EXAMPLE 5.13

– Three cities are to be connected to a water tower at a central location. Label the three cities 1, 2, 3 and the water tower as 0. The cost to lay pipe connecting location i with location j is denoted as $c_{ij}, i \neq j$. Figure 5.6 contains the data for our problem.

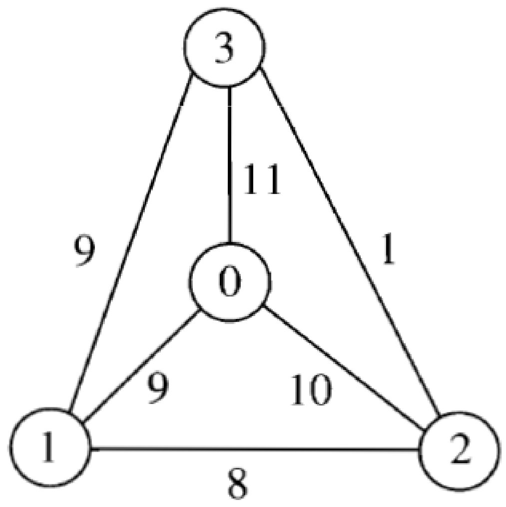


Figure 5.6 Three cities and a water tower.

EXAMPLE 5.13 (cont'd)

– Coalitions among cities can form for pipe to be laid to the water tower. For example, it is possible for city 1 and city 3 to join up so that the cost to the coalition $\{13\}$ would be the sum of the cost of going from 1 to 3 and then 3 to 0. It may be possible to connect from 1 to 3 to 0 but not from 3 to 1 to 0 depending on land conditions. We have the following costs in which we do not treat the water tower as a player:

$$c_1 = 9, c_2 = 10, c_3 = 11, c_{123} = 18,$$
$$c_{12} = 17, c_{13} = 10, c_{23} = 11.$$

The single-player coalitions correspond to hooking up that city directly to location 0. Converting this to a savings game, we let $c(S)$ be the total cost for coalition S and

$$v(S) = \sum_{i \in S} c_i - c(S) = \text{ amount saved by coalition } S.$$

EXAMPLE 5.13 (cont'd)

– This is a three-player game with the set of possible coalitions

$$2^N = \{\emptyset, N, 1, 2, 3, 12, 13, 23, 123\},$$

and we calculate the characteristic function

$$v(i) = 0,\ i = 1, 2, 3,\quad v(123) = 30 - 18 = 12,$$
$$v(12) = 19 - 17 = 2,\ v(13) = 2,\ v(23) = 10.$$

We will find the nucleolus of this game. First, the core of the game is illustrated in Figure 5.7.

EXAMPLE 5.13 (cont'd)

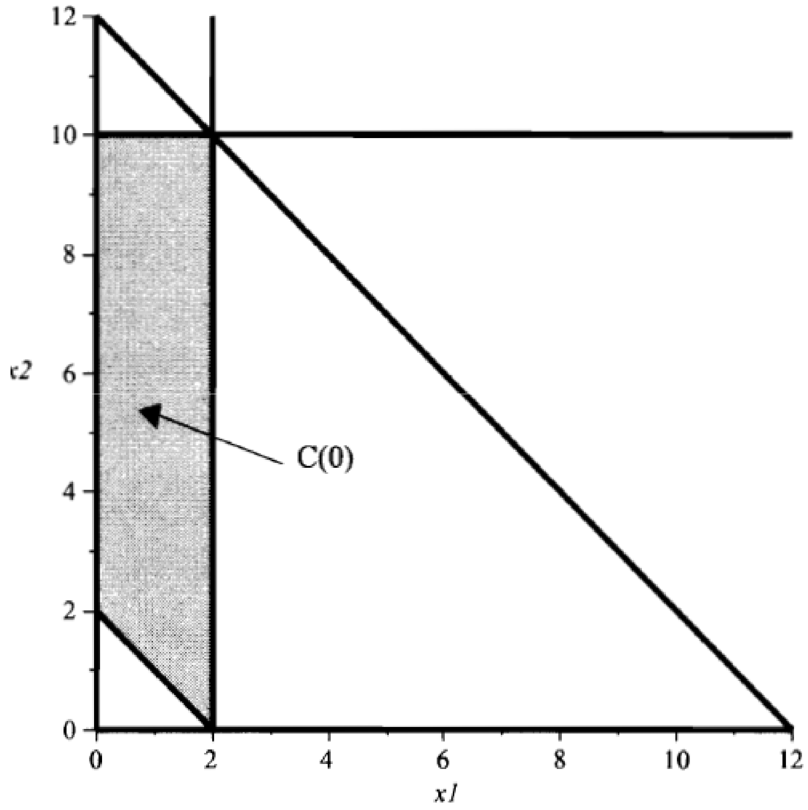


Figure 5.7 Core of three-city problem.

EXAMPLE 5.13 (cont'd)

- This is clearly a nonempty set with many points, so we need to find the nucleolus. This is the complete set of Maple commands needed to do this:

```
> restart:with(simplex):v1:=0:v2:=0:v3:=0:
                          v12:=2:v13:=2:v23:=10:v123:=12;
> with(SolveTools:-Inequality):
> cnsts:={v1-x1<=z,v2-x2<=z,v3-x3<=z,v12-(x1+x2)<=z,
             v13-(x1+x3)<=z,v23-(x2+x3)<=z,x1+x2+x3=v123};
> minimize(z,cnsts);
> fcnsts:=subs(z=-1,x3=12-x1-x2,cnsts);
> gcnsts:=fcnsts[1..7] minus {fcnsts[2]};
> Core1:=subs(z=0,x3=12-x1-x2,cnsts);
> Core:=Core1 minus {Core1[1]};#This is needed to get rid of all
                               #equalities in Core1
```

EXAMPLE 5.13 (cont'd)

```
> with(plots):#The next command plots the core.
> inequal( Core,x1=0..12, x2=0..12,
    optionsfeasible=(color=red),
    optionsopen=(color=blue,thickness=2),
    optionsclosed=(color=green, thickness=3),
    optionsexcluded=(color=yellow),labels=[x1,x2] );
    > # Now we set up for the next least core.
> g1c:=LinearMultivariateSystem(gcnsts,[x1,x2]);
> assign(x1=1,z=-1);
> cnsts1:={v1-x1<=z,v2-x2<=z,v3-x3<=z,v12-(x1+x2)<=z,
           v13-(x1+x3)<=z,v23-(x2+x3)<=z,x1+x2+x3=v123};

> cnsts2:={v2-x2<=z2,v3-x3<=z2,v13-(x1+x3)<=z2,
           v12-(x1+x2)<=z2,x1+x2+x3=v123};
> minimize(z2,cnsts2); #This command results in z2=-9/2
                        for the second least core.
```

EXAMPLE 5.13 (cont'd)

```
> fcnsts2:=subs(z2=-9/2,cnsts2);
> gcnsts2:=fcnsts2[2..4] union {x2+x3<=11,x2+x3>=11};#Needed to
                   #convert equality to inequality
> # We now see if the second least core has more than one point.
> g1c2:=LinearMultivariateSystem(gcnsts2,[x2,x3]);
> assign(x2=11-x3,z2=-9/2);
> cnsts3:={v2-x2<=z2,v3-x3<=z2,v12-(x1+x2)<=z2,
          v13-(x1+x3)<=z2,x1+x2+x3=v123};
```

EXAMPLE 5.13 (cont'd)

- When we get to the last execution group, we have already determined that $x_1 = 1, x_2 + x_3 = 11$, and the last constraint set gives

```
cnst3={12 = 12, -x3 <= -9/2, -x3 <= -11/2, x3 <= 13/2, x3 <= 11/2},
```

which tells us that $x_3 = \frac{11}{2}$ and $x_2 = 11 - \frac{11}{2} = \frac{11}{2}$ .We have found that the nucleolus consists of the single allocation

$$\vec{x} = \left(1, \frac{11}{2}, \frac{11}{2}\right).$$

5.3 The Shapley Value Problems

The Shapley Value

- We change the definition of **fair** from minimizing the maximum dissatisfaction to **allocating an amount proportional to the benefit each coalition derives from having a specific player as a member.**
  - The question is how do we figure out how much benefit each player adds to a coalition. Lloyd Shapley came up with a way.

Definition 5.3.1

- **Definition 5.3.1**

An allocation $\vec{x} = (x_1, \ldots, x_n)$ *is called the* **Shapley value** *if*

$$x_i = \sum_{\{S \in \Pi^i\}} [v(S) - v(S-i)] \frac{(|S|-1)!(|N|-|S|)!}{|N|!}, \quad i = 1, 2, \ldots, n,$$

where Π^i *is the set of all coalitions* $S \subset N$ *containing* i *as a member (i.e.,* $i \in S$*),* $|S|$ *=number of members in* S*, and* $|N| = n$*.*

  - To see where this definition comes from, fix a player, say, *i,* and consider the random variable Z_i, which takes its values in the set of all possible coalitions 2^N.
  - Z_i is the coalition *S* in which *i* is the last player to join *S* and $n - |S|$ players join the grand coalition after player *i*.

Definition 5.3.1 (cont'd)

– Diagrammatically, if *i* joins the coalition *S* on the way to the formation of the grand coalition, we have

$$\underbrace{(1)(2)\cdots(|S|-2)(|S|-1)}_{|S|-1 \text{ arrive}} \quad \underbrace{(i)}_{i \text{ arrives}} \quad \underbrace{(n-|S|)(n-|S|-1)\cdots(2)(1)}_{\text{remaining arrive}}$$

- For a given coalition *S* , by elementary probability, there are (|*S*| — l)!(n - |*S*|)! ways *i* can join the grand coalition N, joining *S* first. With this reasoning, we assume that Z_i has the probability distribution

$$Prob(Z_i = S) = \frac{(|S|-1)!(n-|S|)!}{n!}.$$

Definition 5.3.1 (cont'd)

– The denominator is the total number of ways that the grand coalition can form among *n* players. Any of the *n*! permutations has probability of actually being the way the players join.
  - This distribution assumes that they are **all equally likely.**
– Therefore, for the fixed player i, the benefit player *i* brings to the coalition Z_i is $v(Z_i) - v(Z_i - i)$. *It* seems reasonable that the amount of the total grand coalition benefits that should be allocated to player *i* should be the expected value of $v(Z_i) - v(Z_i - i)$. This gives,

$$x_i \equiv E[v(Z_i) - v(Z_i - i)] = \sum_{\{S \in \Pi_i\}} [v(S) - v(S - i)] Prob(Z_i = S)$$
$$= \sum_{\{S \in \Pi^i\}} [v(S) - v(S - i)] \frac{(|S| - 1)!(n - |S|)!}{n!}.$$

The **Shapley value (or vector)** is then the allocation $\vec{x} = (x_1, \ldots, x_n)$.

EXAMPLE 5.14

- Two players have to divide $M, but they each get zero if they can't reach an agreement as to how to divde it. What is the fair division? Obviously, without regard to the benefit derived from the money the allocation should be $M/2$ to each player. Let's see if Shapley gives that.

  Define $v(1) = v(2) = 0, v(12) = M$. Then

$$x_1 = [v(1) - v(\emptyset)]\frac{0!1!}{1!}2! + [v(12) - v(2)]\frac{1!0!}{2!} = \frac{M}{2}.$$

- Note that if we solve this problem using the least core approach, we get

$$\begin{aligned} C(\varepsilon) &= \{(x_1, x_2) \mid e(S, x) \le \varepsilon, \forall S \subsetneq N\} \\ &= \{(x_1, x_2) \mid -x_1 \le \varepsilon, -x_2 \le \varepsilon,\ x_1 + x_2 = M\} \\ &= \left\{(x_1, x_2) \mid x_1 = x_2 = \frac{M}{2}\right\}. \end{aligned}$$

EXAMPLE 5.14 (cont'd)

- The reason for this is that if $x_1 \geq -\varepsilon$, $x_2 \geq -\varepsilon$, then adding, we have $-2\varepsilon \leq x_1 + x_2 = M$. This implies that $\varepsilon \geq -M/2$, is the restriction on ε. The smallest ε that makes $C(\varepsilon) \neq \emptyset$ is then $\varepsilon^1 = -M/2$. Then $x_1 \geq M/2$, $x_2 \geq M/2$ and, since they add to M, it must be that $x_1 = x_2 = M/2$. So the least core allocation and the Shapley value coincide in the problem.

EXAMPLE 5.15

– Let's go back to the sink allocation (Example 5.9) with Amy, Agnes, and Agatha. Using the core concept, we obtained

player	Truck Capacity	Allocation
Amy	45	35
Agnes	60	50
Agatha	75	65
Total	180	150

– The characteristic function was

$$v(i) = 0, v(13) = 120, v(12) = 105, v(23) = 135, v(123) = 150$$

EXAMPLE 5.15

- In this case $n = 3, n! = 6$ and for player $i = 1$, $\Pi^1 = \{1, 12, 13, 123\}$, so

$$\begin{aligned} x_1 &= \sum_{\{S \in \Pi^1\}} [v(S) - v(S-1)] \frac{(|S|-1)!(3-|S|)!}{3!} \\ &= [v(1) - v(\emptyset)] \frac{2!0!}{3!} + [v(12) - v(2)] \frac{1!1!}{3!} \\ &+ [v(13) - v(3)] \frac{1!1!}{3!} + [v(123) - v(23)] \frac{2!0!}{3!} \\ &= 0 + 105\frac{1}{6} + 120\frac{1}{6} + [150 - 135]\frac{2}{6} \\ &= 42.5 \end{aligned}$$

EXAMPLE 5.15 (cont'd)

– Similarly, with more details,

$$
\begin{aligned}
x_2 &= \sum_{\{S\in\Pi^2\}} [v(S) - v(S-2)] \frac{(|S|-1)!(3-|S|)!}{6} \\
&= [v(2) - v(\emptyset)] Prob(Z_1 = 2) + [v(12) - v(1)] Prob(Z_1 = 12) \\
&+ [v(23) - v(3)] Prob(Z_1 = 23) + [v(123) - v(13)] Prob(Z_1 = 123) \\
&= 0 + 105\frac{1}{6} + 135\frac{1}{6} + [150 - 135]\frac{2}{6} \\
&= 50,
\end{aligned}
$$

EXAMPLE 5.15 (cont'd)

–
$$
\begin{aligned}
x_3 &= \sum_{\{S \in \Pi^3\}} [v(S) - v(S-3)] \frac{(|S|-1)!(3-|S|)!}{6} \\
&= [v(3) - v(\emptyset)] Prob(Z_3 = 3) + [v(13) - v(1)] Prob(Z_3 = 13) \\
&+ [v(23) - v(2)] Prob(Z_3 = 23) + [v(123) - v(12)] Prob(Z_3 = 123) \\
&= 0 + 120\frac{1}{6} + 135\frac{1}{6} + [150 - 105]\frac{2}{6} \\
&= 57.5.
\end{aligned}
$$

Consequently, the Shapley vector is $\vec{x} = (42.5, 50, 57.5)$, or, since we can't split sinks $\vec{x} = (43, 50, 57)$, an allocation quite different from the nucleolus solution of $(35, 50, 65)$.

EXAMPLE 5.16

A typical and interesting fair allocation problem involves a debtor who owes money to more than one creditor. The problem is that the debtor does not have enough money to pay off the entire amount owed to all the creditors. Consequently, the debtor must negotiate with the creditors to reach an agreement about what portion of the assets of the debtor will be paid to each creditor. Usually, but not always, these agreements are imposed by a bankruptcy court.

Let's take a specific problem. Suppose that debtor D has exactly \$100,000 to pay off three creditors A,B,C. Debtor D owes A \$50,000; D owes B \$65,000, and D owes C \$10,000.

Now it is possible for D to split up the \$100K (K=1000) on the basis of percentages; that is, the total owed is \$125,000 and the amount owed to A is 40% of that, to B is 52%, and to C about 8%, so A would get \$40K, B would get \$52K and C would get \$8K. What if the players could form coalitions to try to get more?

EXAMPLE 5.16 (cont'd)

Let's take the characteristic function as follows. The three players are A,B,C and (with amounts in thousands of dollars)

$$v(A) = 25,\ v(B) = 40,\ v(C) = 0,$$
$$v(AB) = 90,\ v(AC) = 35,\ v(BC) = 50,\ v(ABC) = 100.$$

To explain this choice of characteristic function, consider the coalition consisting of just A, If we look at the worst that could happen to A, it would be that B and C get paid off completely and A gets what's left, if anything. If B gets \$65K and C gets \$10K then \$25K is left for A, and so we take $v(A) = 25$. Similarly, if A and B get the entire \$100K, then C gets \$0. If we consider the coalition AC they look at the fact that in the worst case B gets paid \$65K and they have \$35K left as the value of their coalition. This characteristic function is a little pessimistic since it is also possible to consider that AC would be paid \$75K and then $v(AC) = 75$. So other characteristic functions are certainly possible. On the other hand, if two creditors can form a coalition to freeze out the third creditor not in the coalition, then the characteristic function we use here is exactly the result.

EXAMPLE 5.16 (cont'd)

Now we compute the Shapley values. For player A, we have

$$\begin{aligned} x_A &= [v(A) - v(\emptyset)]\frac{1}{3} + [v(AB) - v(B)]\frac{1}{6} \\ &\quad + [v(AC) - v(C)]\frac{1}{6} + [v(ABC) - v(BC)]\frac{1}{3} \\ &= \frac{25}{3} + \frac{50}{6} + \frac{35}{6} + \frac{50}{3} = \frac{235}{6} = 39.17K. \end{aligned}$$

Similarly, for players B and C

$$\begin{aligned} x_B &= [v(B) - v(\emptyset)]\frac{1}{3} + [v(AB) - v(A)]\frac{1}{6} \\ &\quad + [v(BC) - v(C)]\frac{1}{6} + [v(ABC) - v(AC)]\frac{1}{3} \\ &= 40\frac{1}{3} + 65\frac{1}{6} + 50\frac{1}{6} + 65\frac{1}{3} = \frac{325}{6} = 54.17K \end{aligned}$$

EXAMPLE 5.16 (cont'd)

$$
\begin{aligned}
x_C &= [v(C) - v(\emptyset)]\frac{1}{3} + [v(BC) - v(B)]\frac{1}{6} \\
&\qquad + [v(AC) - v(A)]\frac{1}{6} + [v(ABC) - v(AB)]\frac{1}{3} \\
&= 0\frac{1}{3} + 10\frac{1}{6} + 10\frac{1}{6} + 10\frac{1}{3} = \frac{40}{6} = 6.67K,
\end{aligned}
$$

where again K=1000. The Shapley allocation is $\vec{x} = (39.17, 54.17, 6.67)$ compared to the allocation by percentages of $(40, 52, 8)$. Player B will receive more under the Shapley allocation at the expense of players A and C, who are owed the least.

EXAMPLE 5.17

In the beginning of this chapter we presented a typical problem involving small biotech companies. They can discover a new drug but they don't have the resources to manufacture and market it so they have to team up with a large partner. Let's say that A is the biotech firm and B and C are the candidate big pharmaceutical companies. If B or C teams up with A, the big firm will split \$1 billion with A. Here is a possible characteristic function:

$$v(A) = v(B) = v(C) = v(BC) = 0, \quad v(AB) = v(AC) = v(ABC) = 1$$

We will indicate a quicker way to calculate the Shapley allocation when there are a small number of players. We make a table indicating the value brought to a coalition by each player on the way to formation of the grand coalition:

Order of arrival	Player A	Player B	Player C
ABC	0	1	0
ACB	0	0	1
BAC	1	0	0
BCA	1	0	0
CAB	1	0	0
CBA	1	0	0

EXAMPLE 5.17 (cont'd)

The numbers in the table are the amount of value added to a coalition when that player arrives. For example, if A arrives first, no benefit is added; then, if B arrives and joins A, player B has added 1 to the coalition AB; finally, when C arrives (so we have the coalition ABC), C adds no additional value. Since it is assumed in the derivation of the Shapley value that each arrival sequence is **equally likely** we calculate the average benefit brought by each player as the total benefit brought by each player (the sum of each column), divided by the total number of possible orders of arrival. We get

$$x_A = \frac{4}{6},\ x_B = \frac{1}{6},\ \text{and}\ x_C = \frac{1}{6}.$$

So company A, the discoverer of the drug should be allocated two-thirds of the $1 billion and the big companies split the remaining third.

EXAMPLE 5.17 (cont'd)

It is interesting to compare this with the nucleolus. The core, which will be the nucleolus for this example, is

$$\begin{aligned} C(0) = \{\vec{x} = (x_A, x_B, 1 - x_A - x_B) \mid & -x_A \le 0, -x_B \le 0, \\ & -(x_B + 1 - x_A - x_B) \le 0, 1 - x_A - x_B \le 0, \\ & 1 - x_A - (1 - x_A - x_B) \le 0, x_A + x_B \le 1\} \\ = \{(x_A, x_B, x_C) = (1, 0, 0)\}. & \end{aligned}$$

This says that A gets the entire \$1 billion and the other companies get nothing. The Shapley value is definitely more realistic.

Definition 5.3.2

- Shapley vectors can also quickly analyze the winning coalitions in games where winning or losing is all we care about: who do we team up with to win. Here are the definitions.

- **Definition 5.3.2**

Suppose that we are given a normalized characteristic function $v(S)$ that satisfies that for every $S \subset N$, either $v(S) = 0$ or $v(S) = 1$. This is called a **simple game**. *If $v(S) = 1$, the coalition S is said to be a* **winning coalition**. *If $v(S) = 0$, the coalition S is said to be a* **losing coalition**. *Let*

$$W^i = \{S \in \Pi^i \mid v(S) = 1, v(S - i) = 0\},$$

denote the set of coalitions who win with player i and lose without player i.

Definition 5.3.2 (cont'd)

- Simple games are very important in voting systems.
  - For example, a game in which the coalition with a majority of members wins has $v(S) = 1$, if $|S| > n/2$, as the winning coalitions. Losing coalitions have $|S| < n/2$ and $v(S) = 0$. If only unanimous votes win, then $v(N) = 1$ is the only winning coalition. Finally, if there is a certain player who has dictatorial power, say, player 1, then $v(S) = 1$ if $1 \in S$ and $v(S) = 0$ if $1 \in S$.

- In the case of a simple game for player i we need only consider coalitions $S \in \Pi^i$ for which *S* is a winning coalition, but *S* - *i*, that is, *S* without i, is a losing coalition.

Definition 5.3.2 (cont'd)

- We have denoted that set by Wl. We need only consider $S \in W^i$ because $v(S) - v(S - i) = 1$ only when $v(S) = 1$, and $v(S - i) = 0$. In all other cases $v(S) - v(S - i) = 0$. Hence, the Shapley value for a simple game is

$$
\begin{aligned}
x_i &= \sum_{\{S \in \Pi^i\}} [v(S) - v(S-i)] \frac{(|S|-1)!(n-|S|)!}{n!} \\
&= \sum_{\{S \in W^i\}} \frac{(|S|-1)!(n-|S|)!}{n!}
\end{aligned}
$$

- The Shapley allocation for player *i* represents the power that player *i* holds in a game. It is also called the **Shapley-Shubik index.**

EXAMPLE 5.18

A corporation has four stockholders (with 100 total shares) who all vote their own individual shares on any major decision. The majority of shares voted decides an issue. A majority consists of more than 50 shares. Suppose that the holdings of each stockholder are as follows:

player	1	2	3	4
shares	10	20	30	40

The winning coalitions, that is, with $v(S) = 1$ are

$$W = \{24, 34, 123, 124, 234, 1234\}.$$

EXAMPLE 5.18 (cont'd)

We find the Shapley allocation. For x_1, it follows that $W^1 = \{123\}$ because $S = \{123\}$ is winning but $S - 1 = \{23\}$ is losing. Hence

$$x_1 = \frac{(4-3)!(3-1)!}{4!} = \frac{1}{12}.$$

Similarly, $W^2 = \{24, 123, 234\}$, and so

$$x_2 = \frac{1}{12} + \frac{1}{12} + \frac{1}{12} = \frac{1}{4}.$$

Also, $x_3 = \frac{1}{4}$ and $x_4 = \frac{5}{12}$. We conclude that the Shapley allocation for this game is $\vec{x} = (\frac{1}{12}, \frac{3}{12}, \frac{3}{12}, \frac{5}{12})$.Notice that player 1 has the least power, but players 2 and 3 have the same power even though player 3 controls 10 more shares than does player 2. Player 4 has the most power, but a coalition is still necessary to constitute a majority.

EXAMPLE 5.18 (cont'd)

Continue this example, but change the shares as follows

player	1	2	3	4
shares	10	30	30	40

Computing the Shapley value as $x_1 = 0, x_2 = x_3 = x_4 = \frac{1}{3}$, we see that player 1 is completely marginalized as she contributes nothing to any coalition. She has no power. In addition, player 4's additional shares over players 2 and 3 provide no advantage over those players since a coalition is essential to carry a majority in any case.

EXAMPLE 5.19

The United Nations Security Council has 15 members, five of whom are permanent (Russia, Great Britain, France, China, and the United States). These five players have veto power over any resolution. To pass a resolution requires all five permanent member's votes and four of the remaining 10 nonpermanent member's votes. This is a game with fifteen players, and we want to determine the Shapley–Shubik index of their power. We label players 1,2,3,4,5 as the permanent members.

Instead of the natural definition of a winning coalition as one that can pass a resolution, it is easier to use the definition that a winning coalition is one that can **defeat** a resolution. So, for player 1 the winning coalitions are those for which $S \in \Pi^1$, and $v(S) = 1,\ v(S-1) = 0$; that is, player 1, or player 1 and any number up to six nonpermanent members can defeat a resolution, so that the winning coalitions for player 1 is the set

$$W^1 = \{1, 1a, 1ab, 1abc, 1abcd, 1abcde, 1abcdef\},$$

EXAMPLE 5.19 (cont'd)

where the letters denote distinct nonpermanent members. The number of distinct two-player winning coalitions is $10 = \binom{10}{1}$,[5] three-player coalitions is $\binom{10}{2}$, four-player coalitions is $\binom{10}{3}$, and so on, and each of these coalitions will have the same coefficients in the Shapley value. So we get

$$x_1 = \frac{0!14!}{15!} + \binom{10}{1}\frac{1!13!}{15!} + \binom{10}{2}\frac{2!12!}{15!} + \cdots + \binom{10}{6}\frac{6!8!}{15!}$$

We can use Maple to give us the result with this command:

```
> tot:=0;
> for k from 0 to 6 do
    tot:=tot+binomial(10,k)*k!*(14-k)!/15!
    end do:
> print(tot);
```

We get $x_1 = \frac{421}{2145} = 0.1963$. Obviously, it must also be true that $x_2 = x_3 = x_4 = x_5 = 0.19623$. The five permanent members have a total power of $5 \times 0.19623 = 0.9812$ or 98.12% of the power, while the nonpermanent members have $x_6 = \cdots = x_{15} = 0.0019$ or 0.19% each, or a total power of 1.88%.

EXAMPLE 5.20

In this example[6] we show how cooperative game theory can determine a fair allocation of taxes to a community. For simplicity, assume that there are only four households and that the community requires expenditures of \$100,000. The question is how to allocate the cost of the \$100,000 among the four households.

As in most communities, we consider the wealth of the households as represented by the value of their property. Suppose the wealth of household i is w_i. Our four households have specific wealth values

$$w_1 = 75,\ w_2 = 175,\ w_3 = 200,\ w_4 = 300,$$

again with units in thousands of dollars. In addition, suppose that there is a cap on the amount that each household will have to pay (on the basis of age, income, or some other factors) that is independent of the value of their property value. In our case we take the maximum amount each of the four households will be required to pay as

$$u_1 = 25,\ u_2 = 30,\ u_3 = 20,\ u_4 = 80.$$

EXAMPLE 5.20 (cont'd)

What is the fair allocation of expenses to each household?

Let's consider the general problem first. Define the variables

T	Total costs of community
u_i	Maximum amount i will have to pay
w_i	Net worth of player i
z_i	Amount player i will have to pay
$u_i - z_i$	Surplus of the cap over the assessment

The quantity $u_i - z_i$ is the difference between the maximum amount that household i would ever have to pay and the amount household i actually pays. It represents the amount household i does not have to pay.

We will assume that the total wealth of all the players is greater than T, and that the total amount that the players are willing (or are required) to pay is greater than T, but the total actual amount that the players will have to pay is exactly T :

$$\sum_{i=1}^{n} w_i > T, \ \sum_{i=1}^{n} u_i > T, \ \text{ and } \ \sum_{i=1}^{n} z_i = T. \tag{5.3.1}$$

EXAMPLE 5.20 (cont'd)

This makes sense because "you can't squeeze blood out of a turnip." Here is the characteristic function we will use:

$$v(S) = \begin{cases} \max\left(\sum_{i \in S} u_i - T, 0\right) & \text{if } \sum_{i \in S} w_i \geq T; \\ 0 & \text{if } \sum_{i \in S} w_i < T. \end{cases}$$

In other words, $v(S) = 0$ in two cases: (1) if the total wealth of the members of coalition S is less than the total cost, $\sum_{i \in S} w_i < T$, or (2) if the total maximum amount coalition S is required to pay is less than T, $\sum_{i \in S} u_i < T$. If a coalition S cannot afford the expenditure T, then the characteristic function of that coalition is zero.

The Shapley value involves the expression $v(S) - v(S - j)$ in each term. Only the terms with $v(S) - v(S - j) > 0$ need to be considered.

EXAMPLE 5.20 (cont'd)

Suppose first that the coalition S and player $j \in S$ **satisfies** $v(S) > 0$ **and** $v(S-j) > 0$. That means the coalition S and the coalition S without player j can finance the community. We compute

$$v(S) - v(S-j) = \sum_{i \in S} u_i - T - \left(\sum_{i \in S, i \neq j} u_i - T \right) = u_j.$$

Next, suppose that the coalition S can finance the community, but not without $j : v(S) > 0, v(S-j) = 0$. Then

$$v(S) - v(S-j) = \sum_{i \in S} u_i - T.$$

Summarizing the cases, we have

$$v(S) - v(S-j) = \begin{cases} u_j & \text{if } v(S) > 0, v(S-j) > 0; \\ \sum_{i \in S} u_i - T & \text{if } v(S) > 0, v(S-j) = 0; \\ 0 & \text{if } v(S) = v(S-j) = 0. \end{cases}$$

EXAMPLE 5.20 (cont'd)

Notice that if $j \in S$ and $v(S - j) > 0$, then automatically $v(S) > 0$. We are ready to compute the Shapley allocation. For player $j = 1, \ldots, n$, we have,

$$
\begin{aligned}
x_j &= \sum_{\{S \in \Pi^j\}} [v(S) - v(S - j)] \frac{(|S| - 1)!(n - |S|)!}{n!} \\
&= \sum_{\{S | j \in S, v(S-j)>0\}} u_j \frac{(|S| - 1)!(n - |S|)!}{n!} \\
&+ \sum_{\{S | j \in S, v(S)>0, v(S-j)=0\}} \left(\sum_{i \in S} u_i - T \right) \frac{(|S| - 1)!(n - |S|)!}{n!}
\end{aligned}
$$

By our definition of the characteristic function for this problem, the allocation x_j is the portion of the surplus $\sum_i u_i - T > 0$ that will be assessed to household j. Consequently, the amount player j will be billed is actually $z_j = u_j - x_j$.

EXAMPLE 5.20 (cont'd)

For the four-person problem data above we have $T = 100, \sum w_i = 750 > 100, \sum_i u_i = 155 > 100$, so all our assumptions in (5.3.1) are verified. Remember that the units are in thousands of dollars. Then we have

$$v(i) = 0,\ v(12) = v(13) = v(23) = 0,\ v(14) = 5, v(24) = 10, v(34) = 0,$$
$$v(123) = 0, v(134) = 25, v(234) = 30, v(124) = 35, v(1234) = 55.$$

For example, $v(134) = \max(u_1 + u_3 + u_4 - 100, 0) = 125 - 100 = 25$. We compute

$$\begin{aligned} x_1 &= \sum_{\{S \mid 1 \in S, v(S-1)>0\}} u_1 \frac{(|S|-1)!(4-|S|)!}{4!} \\ &+ \sum_{\{S \mid 1 \in S, v(S)>0, v(S-1)=0\}} \left(\sum_{i \in S} u_i - T \right) \frac{(|S|-1)!(n-|S|)!}{n!} \end{aligned}$$

EXAMPLE 5.20 (cont'd)

$$\begin{aligned} &= \frac{2!1!}{4!} \cdot u_1 + \frac{3!0!}{4!} u_1 \\ &\quad + \frac{1!2!}{4!} ([u_1 + u_4 - 100]) + \frac{2!1!}{4!} [u_1 + u_3 + u_4 - 100] \\ &= \frac{65}{6}. \end{aligned}$$

The first term comes from coalition $S = 124$; the second term, from coalition $S = 1234$; the third term comes from coalition $S = 14$; and the last term from coalition $S = 134$.

As a result, the amount player 1 will be billed will be $z_1 = u_1 - x_1 = 25 - \frac{65}{6} = \frac{85}{6}$ thousand dollars. In a similar way we calculate

$$x_2 = \frac{40}{3}, \quad x_3 = \frac{25}{3}, \quad \text{and} \quad x_4 = \frac{45}{2},$$

EXAMPLE 5.20 (cont'd)

so that the actual bill to each player will be

$$\begin{aligned} z_1 &= 25 - \tfrac{65}{6} = 14.167, \\ z_2 &= 30 - \tfrac{40}{3} = 16.667, \\ z_3 &= 20 - \tfrac{25}{3} = 11.667, \\ z_4 &= 80 - \tfrac{45}{2} = 57.5. \end{aligned}$$

EXAMPLE 5.20 (cont'd)

For comparison purposes it is not too difficult to calculate the nucleolus for this game to be $\left(\frac{25}{2}, 15, 10, \frac{35}{2}\right)$, so that the payments using the nucleolus will be

$$\begin{aligned} z_1 &= 25 - \tfrac{25}{2} = \tfrac{25}{2} = 12.5, \\ z_2 &= 30 - 15 = 15, \\ z_3 &= 20 - 10 = 10, \\ z_4 &= 80 - \tfrac{35}{2} = \tfrac{125}{2} = 62.5. \end{aligned}$$

There is yet a third solution, the straightforward solution that assesses the amount to each player in proportion to each household's maximum payment to the total assessment. For example, $u_1/(\sum_i u_i) = 25/155 = 0.1613$ and so player 1 could be assessed the amount $0.1613 \times 100 = 16.13$.

Dummy and Carriers

- A player *i* is a **dummy** if for any coalition *S* in which $i \notin S$, we have

  $$v(S \cup i) = v(S).$$

  So dummy player *i* contributes nothing to any coalition. The players who are not dummies are called the **carriers** of the game. Let's define *C =* set of carriers.

Dummy and Carriers (cont'd)

- Given a characteristic function *v,* we should get an allocation as a function of *v*, $\varphi(v) = (\varphi_1(v), \ldots, \varphi_n(v))$, where $\varphi_i(v)$ will be the allocation or worth or value of player *i* in the game, and this function φ should satisfy the following properties:

1. $v(N) = \sum_{i=1}^{n} \varphi_i(v)$. (Group rationality).
2. If players i and j satisfy $v(S \cup i) = v(S \cup j)$ for any coalition with $i \notin S$, $j \notin S$, then $\varphi_i(v) = \varphi_j(v)$. If i and j provide the same benefit, they should have the same worth.
3. If i is a dummy player, $\varphi_i(v) = 0$. Dummies should be worth nothing.
4. If v_1 and v_2 are two characteristic functions, then $\varphi(v_1 + v_2) = \varphi(v_1) + \varphi(v_2)$.

Dummy and Carriers (cont'd)

  - It essentially says that the allocation to a player using the sum of characteristic functions should be the sum of the allocations corresponding to each characteristic function.

- There is one and only one function φ that satisfies them! It is given by $\varphi(v) = (\varphi_1(v), \ldots, \varphi_n(v))$, where

$$\varphi_i(v) = \sum_{\{S \in \Pi^i\}} [v(S) - v(S - i)] \frac{(|S| - 1)!(|N| - |S|)!}{|N|!}, i = 1, 2, \ldots, n.$$

  This is the **only** function satisfying the properties, and, sure enough, it is the Shapley value.

5.4 Bargaining

Bargaining

- In this section we will introduce a new type of cooperative game in which the players bargain to improve both of their payoffs. Let us start with a simple example to illustrate the benefits of bargaining and cooperation.

- Consider the prisoner's dilemma **two-player nonzero sum game with bimatrix**

	II_1	II_2
I_1	$(2, 1)$	$(-1, -1)$
I_2	$(-1, -1)$	$(1, 2)$

Bargaining (cont'd)

- You can easily check that there are three Nash equilibria given by $X_1 = (0,1) = Y_1, X_2 = (1,0) = Y_2$, and $X_3 = (\frac{3}{5}, \frac{2}{5}), Y_3 = (\frac{2}{5}, \frac{3}{5})$.Now consider Figure 5.8.

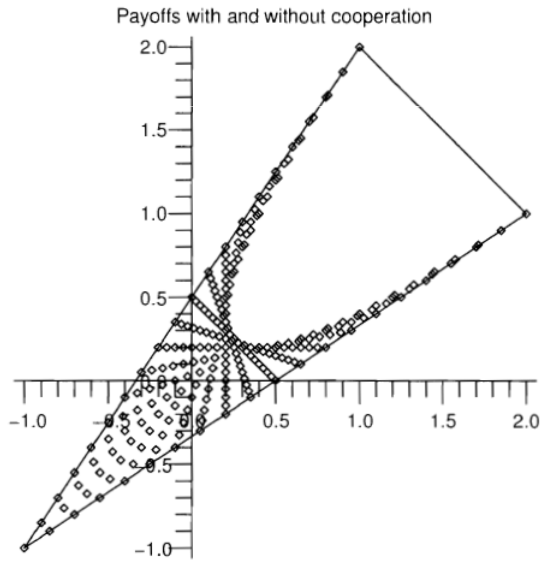


Figure 5.8 Payoff I versus payoff II for the prisoner's dilemma.

Bargaining (cont'd)

- The points represent the possible pairs of pay offs to each $(E_1(x,y), E_2(x,y))$ given by

$$E_1(x,y) = (x, 1-x)A\begin{pmatrix} y \\ 1-y \end{pmatrix}, \qquad E_2(x,y) = (x, 1-x)B\begin{pmatrix} y \\ 1-y \end{pmatrix}.$$

- It was generated with the following Maple commands:

```
> with(plots):with(plottools):with(LinearAlgebra):
> A:=Matrix([[2,-1],[-1,1]]);B:=Matrix([[1,-1],[-1,2]]);
> f:=(x,y)->expand(Transpose(<x,1-x>).A.<y,1-y>);
> g:=(x,y)->expand(Transpose(<x,1-x>).B.<y,1-y>);
> points:={seq(seq([f(x,y),g(x,y)],x=0..1,0.05),y=0..1,0.05)}:
> pure:=([[2,1],[-1,-1],[-1,-1],[1,2]]);
> pp:=pointplot(points);
> pq:=polygon(pure,color=yellow);
> display(pq,pp,title="Payoffs with and without cooperation");
```

Bargaining (cont'd)

- The horizontal axis (abscissa) is the payoff to player I, and the vertical axis (ordinate) is the payoff to player II. Any point in the parabolic region is achievable for a some 0 < *x* < 1,0 < *y* < 1.
- The parabola is given by the implicit equation

$$5(E_1 - E_2)^2 - 2(E_1 + E_2) + 1 = 0$$

  - If the players play pure strategies, the payoff to each player will be at one of the vertices. The pure Nash equilibria yield the payoff pairs $(E_1 = 1, E_2 = 2)$ and $(E_1 = 2, E_2 = 1)$.
  - The mixed Nash point gives the payoff pair $(E_1 = \frac{1}{5}, E_2 = \frac{1}{5})$, which is strictly inside the region of points, called the **noncooperative payoff set.**

Bargaining (cont'd)

- Now, if the players do not cooperate, they will achieve one of two possibilities:
  1. The vertices of the figure, if they play pure strategies.
  2. Any point in the region of points bounded by the two lines and the parabola, if they play mixed strategies.

  – The portion of the triangle outside the parabolic region is **not** achievable simply by the players using mixed strategies. However, if the players agree to cooperate, then any point on the boundary of the triangle, the entire shaded region, including the boundary of the region, are achievable payoffs.

Bargaining (cont'd)

– Player I wants a payoff as large as possible and thus as far to the right on the triangle as possible. Player II wants to go as high on the triangle as possible. So player I wants to get the payoff at (2,1), and player II wants the payoff at (1,2), but this is possible if and only if the opposing player agrees to play the correct strategy. In addition, it seems that nobody wants to play the mixed Nash equilibrium because they can both do better, but they have to cooperate to achieve a higher payoff.

EXAMPLE 5.21

	II_1	II_2	II_3
I_1	$(1,4)$	$(-2,1)$	$(1,2)$
I_2	$(0,-2)$	$(3,1)$	$(\frac{1}{2},\frac{1}{2})$

- We will draw the pure payoff points of the game as the vertices of the graph and connect the pure payoffs with straight lines, as in Figure 5.9.
- The vertices of the polygon are the payoffs from the matrix. The solid lines connect the pure payoffs.
- The dotted lines extend the region of payoffs to those payoffs that **could** be achieved if both players cooperate.

EXAMPLE 5.21

- Suppose that player I always chooses row 2, I_2 , and player II plays the mixed strategy $Y = (y_1, y_2, y_3)$, where $y_i \geq 0, y_1 + y_2 + y_3 = 1$.
- The expected payoff to I is

$$E_1(2, Y) = 0\ y_1 + 3y_2 + \frac{1}{2}\ y_3,$$

- The expected payoff to II is

$$E_2(2, Y) = -2y_1 + 1y_2 + \frac{1}{2}\ y_3.$$

- Hence

$$(E_1, E_2) = y_1(0, -2) + y_2(3, 1) + y_3\left(\frac{1}{2}, \frac{1}{2}\right),$$

  - which, as a linear combination of the three points (0,-2), (3,1), and (1,1), is in the convex hull of these three points.

EXAMPLE 5.21 (cont'd)

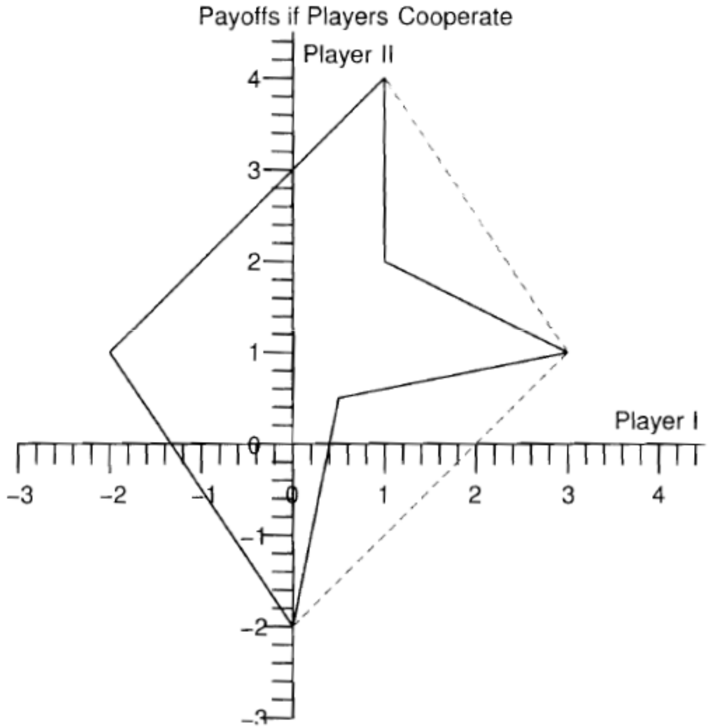


Figure 5.9 Achievable payoffs with cooperation.

EXAMPLE 5.21 (cont'd)

- This means that if players I and II can agree that player I will always play row 2, then player II can choose a $Y = (y_1, y_2, y_3)$ at the payoff pair to each player will be in the triangle bounded by the lower dotted line in Figure 5.9 and the lines connecting (0,-2) with (1,1) with (3,1).
- The conclusion is that any point in the convex hull of all the payoff points is achievable if the players agree to cooperate.

Definition 5.4.1

- The entire triangle in Figure 5.9 is called the **feasible set** for the problem. The precise definition in general is as follows.

- **Definition 5.4.1**

  The **feasible set** *is the convex hull of all the payoff points corresponding to pure strategies of the players.*

  - The objective of player I in Example 5.21 is to obtain a payoff as far to the right as possible in Figure 5.9, and the objective of player II is to obtain a payoff as far up as possible in Figure 5.9. Player I's ideal payoff is at the point (3,1), but that is attainable only if II agrees to play II_2.

Definition 5.4.1 (cont'd)

- Why would he do that? Similarly, II would do best at (1,4), which will happen only if I plays I_1, and why would she do that? There is an incentive for the players to reach a compromise agreement in which they would agree to play in such a way so as to obtain a payoff along the line connecting (1,4) and (3,1).
- That portion of the boundary is known as the **Pareto-optimal boundary** because it is the edge of the set and has the property that if either player tries to do better (say, player I tries to move further right), then the other player will do worse (player II must move down to remain feasible).

Definition 5.4.2

- **Definition 5.4.2**

  The Pareto-optimal boundary of the feasible set is the set of payoff points in which no player can improve his payoff without at least one other player decreasing her payoff.

  – The point of this discussion is that there is an incentive for the players to cooperate and try to reach an agreement that will benefit both players. The result will always be a payoff pair occuring on the Pareto-optimal boundary of the feasible set.

Definition 5.4.3

- In any bargaining problem there is always the possibility that negotiations will fail. Hence, each player must know what the payoff would be if there were no bargaining. This leads us to the next definition.

- **Definition 5.4.3**

  The **status quo payoff point**, *or* **safety point**, *or* **security point** *in a two-person game is the pair of payoffs* (u^*, v^*) *that each player can achieve if there is no cooperation between the players.*

  – Recall that the safety levels we used in previous sections were defined by the pair (*value*(*A*), *value*(B^T)).

Definition 5.4.3 (cont'd)

- In the context of bargaining it is simply a noncooperative payoff to each player if no cooperation takes place.
  - For most problems considered in this section, the status quo point will be taken to be the values of the zero sum games associated with each player, because those values can be guaranteed to be achievable, no matter what the other player does.

EXAMPLE 5.22

- We will determine the security point for each player in Example 5.21, and take it to be the value of the zero sum games for each player
- Consider the payoff matrix for player I:

$$A = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 3 & \frac{1}{2} \end{bmatrix}$$

$v(A) = \frac{1}{2}$

The optimal strategies for player II: $Y = \left(\frac{5}{6}, \frac{1}{6}, 0\right)$

The optimal strategies for player I : $X = \left(\frac{1}{2}, \frac{1}{2}\right)$

EXAMPLE 5.22 (cont'd)

- Next we consider the payoff matrix for player II

$$B^T = \begin{bmatrix} 4 & -2 \\ 1 & 1 \\ 2 & \frac{1}{2} \end{bmatrix}.$$

  For this matrix $v(B^T) = 1$, and we have a saddle point at row 1 column 2.
- We conclude that the status quo point for this game is $(\frac{1}{2}, 1)$ since that is the guaranteed payoff to each player without cooperation or nego-tiation. This means that any bargaining must begin with the guaranteed payoff pair $(\frac{1}{2}, 1)$.

EXAMPLE 5.22 (cont'd)

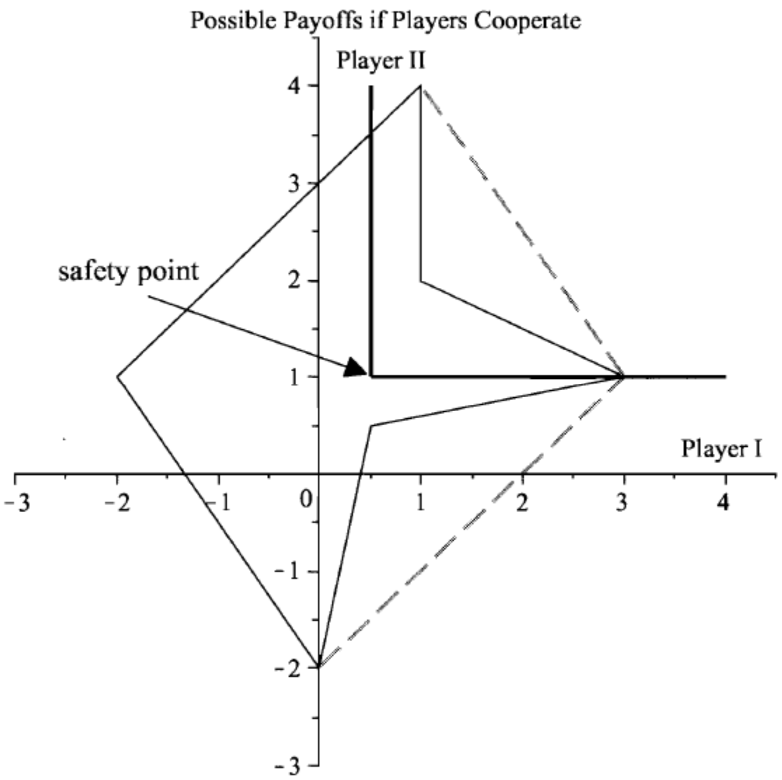


Figure 5.10 The reduced feasible set; safety at $(\frac{1}{2}, 1)$.

EXAMPLE 5.22 (cont'd)

- This cuts off the feasible set as in Figure 5.10. The new feasible of the points in Figure 5.10 above and to the right of the lines emanating from the security point $(\frac{1}{2}, 1)$.
- Notice that in this problem the Pareto-optimal boundary is the line connect-ing (1,4) and (3,1) because no player can get a bigger payoff on this line without forcing the other player to get a smaller payoff.

- The question now is to find the cooperative, negotiated best payoff for each player. How does cooperation help?
  - I will play row I_1 half the time and row I_2 half the time as long as II plays column II_1 half the time and column II_2 half the time.

EXAMPLE 5.22 (cont'd)

- If they agree to play this way, they will get $\frac{1}{2}(1,4) + \frac{1}{2}(3,1) = (2, \frac{5}{2})$.
- So player I gets $2 > \frac{1}{2}$ and player II gets $\frac{5}{2} > 1$, a big improvement for each player over his or her own individual safety level. So, they definitely have an incentive to cooperate.

EXAMPLE 5.23

– The bimatrix is

	II_1	II_2
I_1	$(2, 17)$	$(-10, -22)$
I_2	$(-19, -7)$	$(17, 2)$

the safety level is given by the point

$$(value(A), value(B^T)) = \left(-\frac{13}{4}, -\frac{5}{2}\right),$$

the optimal strategies

$$X_A = (\tfrac{3}{4}, \tfrac{1}{4}), Y_A = (\tfrac{9}{16}, \tfrac{7}{16}), \text{ and } X_B = (\tfrac{1}{2}, \tfrac{1}{2}), Y_B = (\tfrac{3}{16}, \tfrac{13}{16})$$

Negotiations start from the safety point.

EXAMPLE 5.23 (cont'd)

- Figure 5.11 shows the safety point and the associated feasible payoff pairs above and to the right of the dark lines. The shaded region in Figure 5.11 is the convex hull of the pure payoffs, namely, the feasible set, and is the set of all possible negotiated payoffs.
- The region of dot points is the set of noncooperative payoff pairs if we consider the use of all possible mixed strategies.
- A negotiated set of payoffs will benefit both players and will be on the line farthest to the right, which is the Pareto-optimal boundary.

EXAMPLE 5.23 (cont'd)

- Player I would love to get (17, 2), while player II would love to get (2,17). That probably won't occur but they could negotiate a point along the line connecting these two points and compromise on obtaining, say, the midpoint

$$\frac{1}{2}(2, 17) + \frac{1}{2}(17, 2) = (9.5, 9.5).$$

- So they could negotiate to get 9.5 each if they agree that each player would use the pure strategies $X = (1,0) = Y$ half the time and play pure strategies $X = (0,1) = Y$ exactly half the time. They have an incentive to cooperate.

EXAMPLE 5.23 (cont'd)

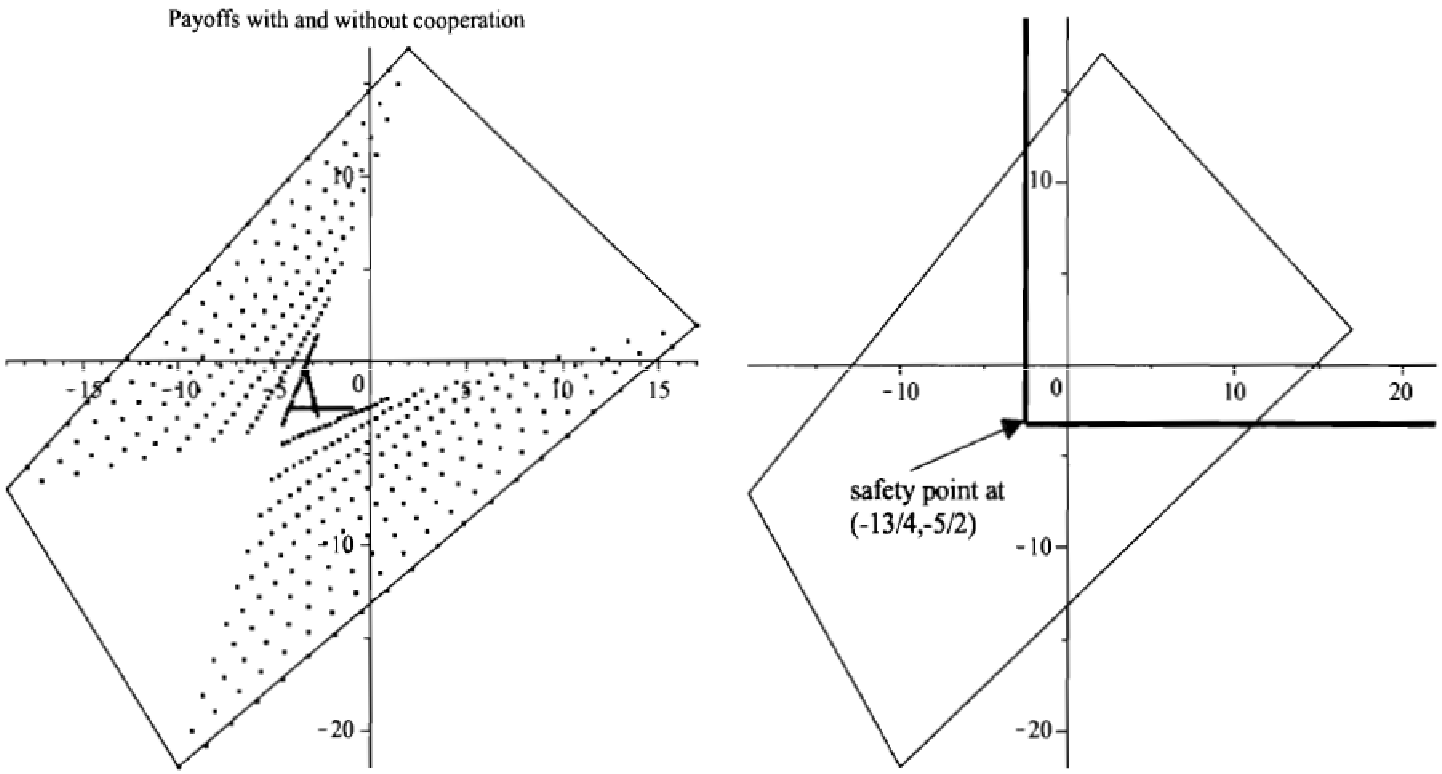


Figure 5.11 Achievable payoff pairs with cooperation; safety point $= \left(\frac{13}{4}, -\frac{5}{2}\right)$.

5.4.1 The Nash Model with Security Point

The Nash Model with Security Point

- We start with any old security status quo point (u^*, v^*) for a two-player cooperative game with matrices *A*, *B*. This leads to a feasible set of possible negotiated outcomes depending on the point we start from (u^*, v^*). This may be the safety point $u^* = value(A), v^* = value(B^T)$, or not. For any given such point and feasible set *S*, we are looking for a negotiated outcome, call it $(\overline{u}, \overline{v})$. This point will depend on (u^*, v^*) and the set *S*, so we may write $(\overline{u}, \overline{v}) = f(S, u^*, v^*)$.

The Nash Model with Security Point (cont'd)

- The question is how to determine the point $(\overline{u}, \overline{v})$? John Nash proposed the following requirements for the point to be a negotiated solution:
  - **Axiom 1**. We must have $\overline{u} \geq u^*$ and $\overline{v} \geq v^*$. Each player must get at least the status quo point.
  - **Axiom 2.** The point $(\overline{u}, \overline{v}) \in S$, that is, it must be a feasible point.
  - **Axiom 3.** If (u, v) is any point in S, so that $u \geq \overline{u}$ and $v \geq \overline{v}$, then it must be the case that $u = \overline{u}, v = \overline{v}$. In other words, there is no other point in S, where **both** players receive more. This is **Pareto-optimality**.
  - **Axiom 4.** If $(\overline{u}, \overline{v}) \in T \subset S$ and $(\overline{u}, \overline{v}) = f(T, u^*, v^*)$ is the solution to the bargaining problem with feasible set T, then for the larger feasible set S, either $(\overline{u}, \overline{v}) = f(S, u^*, v^*)$ is the bargaining solution for S, or the actual bargaining solution for S is in $S - T$. We are assuming that the security point is the same for T and S. So, if we have more alternatives, the new negotiated position can't be one of the old possibilities.

The Nash Model with Security Point (cont'd)

- **Axiom 5.** If T is an affine transformation of $S, T = aS + b = \varphi(S)$ and $(\overline{u}, \overline{v}) = f(S, u^*, v^*)$ is the bargaining solution of S with security point (u^*, v^*), then $(a\overline{u} + b, a\overline{v} + b) = f(T, au^* + b, av^* + b)$ is the bargaining solution associated with T and security point $(au^* + b, av^* + b)$. This says that the solution will not depend on the scale or units used in measuring payoffs.

- **Axiom 6.** If the game is symmetric with respect to the players, then so is the bargaining solution. In other words, if $(\overline{u}, \overline{v}) = f(S, u^*, v^*)$ and (i) $u^* = v^*$, and (ii) $(u, v) \in S \Rightarrow (v, u) \in S$, then $\overline{u} = \overline{v}$. So, if the players are essentially interchangeable they should get the same negotiated payoff.

- The amazing thing that Nash proved is that if we assume these axioms, there is one and only one solution of the bargaining problem.

Theorem 5.4.4

- **Theorem 5.4.4**

Let the set of feasible points for a bargaining game be nonempty and convex, and let $(u^*, v^*) \in S$ *be the security point. Consider the nonlinear programming problem*

$$\text{Maximize } g(u,v) := (u - u^*)(v - v^*)$$
$$\text{subject to } (u,v) \in S, u \geq u^*, v \geq v^*.$$

Assume that there is at least one point $(u,v) \in S$ *with* $u > u^*, v > v^*$. *Then there exists one and only one point* $(\overline{u}, \overline{v}) \in S$ *that solves this problem, and this point is the unique solution of the bargaining problem* $(\overline{u}, \overline{v}) = f(S, u^*, v^*)$ *that satisfies the axioms* $1 - 6$. *If, in addition, the game satisfies the symmetry assumption, then the conclusion of axiom 6 tells us that* $\overline{u} = \overline{v}$

Theorem 5.4.4 (cont'd)

Proof. We will only sketch a part of the proof and skip the rest.
1. **Existence.** Define the function $g(u, v) \equiv (u - u^*)(v - v^*)$. The set

$$S^* = \{(u, v) \in S \mid u \geq u^*, v \geq v^*\}$$

is convex, closed, and bounded. Since $g : S^* \to \mathbb{R}$ is continuous, a theorem of analysis (any continuous function on a closed and bounded set achieves a maximum and a minimum on the set) guarantees that g has a maximum at some point $(\overline{u}, \overline{v}) \in S^*$. By assumption there is at least one feasible point with $u > u^*, v > v^*$. For this point $g(u, v) > 0$ and so the maximum of g over S^* must be > 0 and therefore does not occur at the safety points $u = u^*$ or $v = v^*$.

Theorem 5.4.4 (cont'd)

2. **Uniqueness.** Suppose the maximum of g occurs at two points $0 < M = g(u', v') = g(u'', v'')$. If $u' = u''$, then

$$g(u', v') = (u' - u^*)(v' - v^*) = g(u'', v'') = (u'' - u^*)(v'' - v^*),$$

so that dividing out $u' - u^*$, implies that $v' = v''$ also. So we may as well assume that $u' < u''$ and that implies $v' > v''$ because $(u' - u^*)(v' - v^*) = (u'' - u^*)(v'' - v^*) = M > 0$ so that

$$\frac{u'' - u^*}{u' - u^*} = \frac{v' - v^*}{v'' - v^*} > 1 \implies v' - v^* > v'' - v^* \implies v' > v''.$$

Set $(u, v) = \frac{1}{2}(u', v') + \frac{1}{2}(u'', v'')$. Since S is convex, $(u, v) \in S$ and $u > u^*, v > v^*$. So $(u, v) \in S^*$. Some simple algebra shows that

$$g(u, v) = M + \frac{(u' - u'')(v'' - v')}{4} > M, \text{ since } u'' > u', v'' < v'.$$

This contradicts the fact that (u', v') provides a maximum for g over S^* and so the maximum point must be unique.

Theorem 5.4.4 (cont'd)

3. **Pareto-optimality.** We show that the solution of the nonlinear program, say, $(\overline{u}, \overline{v})$, is Pareto-optimal. If it is not Pareto-optimal, then there must be another feasible point $(u', v') \in S$ for which either $u' > \overline{u}$ and $v' \geq \overline{v}$, or $v' > \overline{v}$ and $u' \geq \overline{u}$. We may as well assume the first possibility. Since $\overline{u} > u^*, \overline{v} > v^*$, we then have $u' > u^*$ and $v' > v^*$ and so $g(u', v') > 0$. Next, we have

$$g(u', v') = (u' - u^*)(v' - v^*) > (\overline{u} - u^*)(\overline{v} - v^*) = g(\overline{u}, \overline{v}).$$

But this contradicts the fact that $(\overline{u}, \overline{v})$ maximizes g over the feasible set. Hence $(\overline{u}, \overline{v})$ is Pareto-optimal.

EXAMPLE 5.24

– In an earlier example we considered the game with bimatrix

	II_1	II_2
I_1	$(2, 17)$	$(-10, -22)$
I_2	$(-19, -7)$	$(17, 2)$

The safety levels $u^* = value(A) = -\frac{13}{4}, v^* = value(B^T) = -\frac{5}{2},$

– Figure 5.12 for this problem shows the safety point and the associated feasible payoff pairs above and to the right.

– We need the equation of the lines forming the Pareto-optimal boundary.

In this example it is simply $v = -u + 19$, which is the line with negative slope to the right of the safety point. It is the only place where both players cannot simultaneously improve their payoffs.

EXAMPLE 5.24 (cont'd)

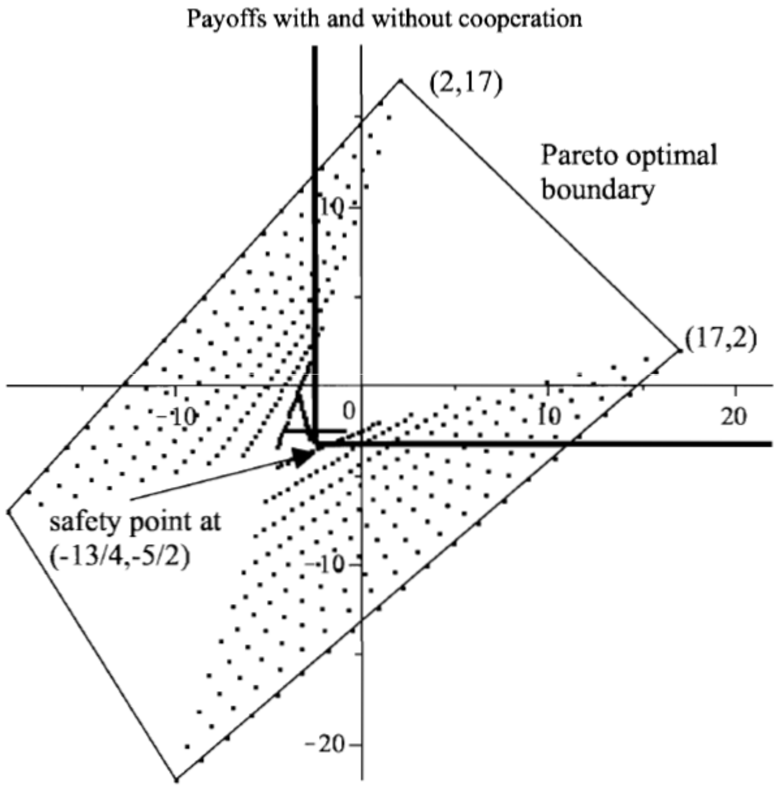


Figure 5.12 Pareto-optimal boundary is line connecting $(2, 17)$ and $(17, 2)$.

EXAMPLE 5.24 (cont'd)

- To find the bargaining solution for this problem, we have to solve the nonlinear programming problem

$$\text{Maximize } \left(u + \frac{13}{4}\right)\left(v + \frac{5}{2}\right)$$
$$\text{subject to } u \geq -\frac{13}{4}, \quad v \geq -\frac{5}{2}, \quad v \leq -u + 19$$

- The Maple commands used to solve this are

```
> with(Optimization):
> NLPSolve((u+13/4)*(v+5/2),
     {u>=-13/4,v>=-5/2,v<=-u+19},maximize);
```

This gives the optimal bargained payoff pair $(\overline{u} = \frac{73}{8} = 9.125, \overline{v} = \frac{79}{8} = 9.875)$. The maximum of g is $g(\overline{u}, \overline{v}) = 153.14$, which we do not really use or need.

EXAMPLE 5.24 (cont'd)

- The bargained payoff to player I is $\overline{u} = 9.125$ and the bargained payoff to player II is $\overline{v} = 9.875$. We do not get the point we expected, namely, (9.5,9.5); that is due to the fact that the security point is not symmetric. Player II has a small advantage.
- You can see in the Maple generated Figure 5.13 that the solution of the problem occurs just where the level curves, or contours of *g* are tangent to the boundary of the feasible set. Since the function *g* has concave up contours and the feasible set is convex, this must occur at exactly one point.

EXAMPLE 5.24 (cont'd)

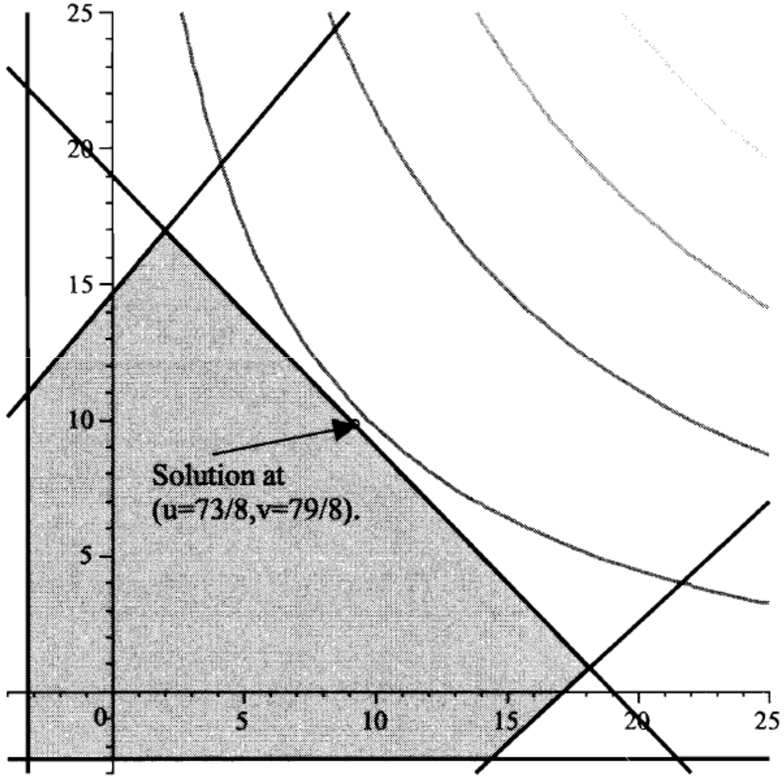


Figure 5.13 Bargaining solution where curves just touch Pareto boundary at $(9.125, 9.875)$.

EXAMPLE 5.24 (cont'd)

– The Maple commands used to get Figure 5.13 are as follows.

```
> f:=(x,y)->(x+13/4)*(y+5/2);
> gcnst:={x >=-13/4,y>=-5/2,y<=-x+19,
                  y<=24/21*x+14.71,y>=24/27*x-15.22};
> with(plots):with(plottools):
> cp:=contourplot(f(x,y),x=0..25,y=0..25,
                  axes=normal,thickness=2,contours=4):
> ineq:=inequal( gcnst,x=-4..25, y=-3..25,
    optionsfeasible=(color=yellow),
    optionsopen=(color=blue,thickness=2),
    optionsclosed=(color=green, thickness=2),
    optionsexcluded=(color=white),labels=[x,y] ):
> pointp:=pointplot([73/8,79/8],thickness=5,symbol=circle):
> t1:=textplot([16,13,"(73/8,79/8)"],align={BELOW,LEFT});
> display3d(cp,ineq,t1,pointp,title="Bargaining Solution",
                  labels=['u','v'] );
```

EXAMPLE 5.24 (cont'd)

– Finally, knowing that the optimal point must occur on the Pareto-optimal bound-ary means we could solve the nonlinear programming problem by calculus. We want to maximize

$$f(u) = g(u, -u + 19) = (u + \frac{13}{4})(-u + 19 + \frac{5}{2}), \text{ on the interval } 2 \leq u \leq 17.$$

This is an elementary calculus maximization problem.

EXAMPLE 5.25

– We will work through another example from scratch. We start with the follow-ing bimatrix:

	II_1	II_2
I_1	$(1,3)$	$(-4,-2)$
I_2	$(-1,-3)$	$(2,1)$

– 1. **Find the security point.** To begin we find the values of the associated matrices

$$A = \begin{bmatrix} 1 & -4 \\ -1 & 2 \end{bmatrix}, \qquad B^T = \begin{bmatrix} 3 & -3 \\ -2 & 1 \end{bmatrix}.$$

Then, $value(A) = -\frac{1}{4}$ and $value(B^T) = -\frac{1}{3}$. Hence the security point is $(u^*, v^*) = \left(-\frac{1}{4}, -\frac{1}{3}\right)$.

EXAMPLE 5.25 (cont'd)

– 2. **Find the feasible set.** The feasible set, taking into account the security point, is

$$S^* = \{(u,v) \mid u \geq -\frac{1}{4}, v \geq -\frac{1}{3}, 0 \leq 10 + 5u - 5v, 0 \leq 10 + u + 3v,$$
$$0 \leq 5 - 4u + 3v, 0 \leq 5 - 2u - v\}.$$

– 3. **Set up and solve the nonlinear programming problem.** The nonlinear programming problem is then

$$\text{Maximize } g(u,v) \equiv \left(u + \frac{1}{4}\right)\left(v + \frac{1}{3}\right)$$
$$\text{subject to } (u,v) \in S^*.$$

Maple gives us the solution $\overline{u} = \frac{29}{24} = 1.208, \overline{v} = \frac{31}{12} = 2.583$. If we look at Figure 5.14 for S^*, we see that the Pareto-optimal boundary is the line $v = -2u + 5, 1 \leq u \leq 2$.

EXAMPLE 5.25 (cont'd)

– 4. **Find the strategies giving the negotiated solution.** How should the players cooperate in order to achieve the bargained solutions we just obtained? To find out, the only points in the bimatrix that are of interest are the endpoints of the Pareto-optimal boundary, namely, $(1,3)$ and $(2,1)$. So the cooperation must be a linear combination of the strategies yielding these payoffs. Solve

$$\left(\frac{29}{24}, \frac{31}{12}\right) = \lambda(1,3) + (1-\lambda)(2,1),$$

to get $\lambda = \frac{19}{24}$. This says that (I,II) must agree to play (row 1,col 1) with probability $\frac{19}{24}$ and (row 2, col 2) with probability $\frac{5}{24}$.

EXAMPLE 5.25 (cont'd)

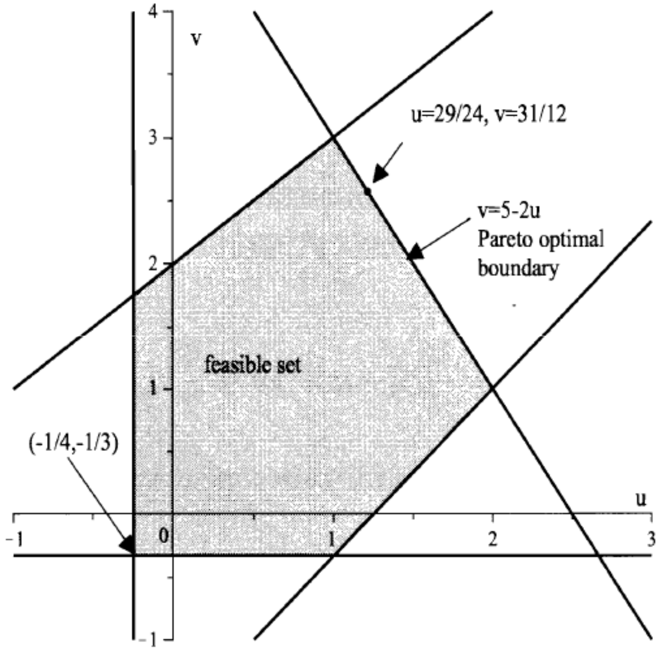


Figure 5.14 Security point $\left(-\frac{1}{4}, -\frac{1}{3}\right)$, Pareto boundary $v = -2u + 5$, solution $(1.208, 2.583)$.

EXAMPLE 5.26

- Suppose that two persons are given $1000, which they can split if they can agree on how to split it. If they cannot agree they each get nothing.
- One player is rich, so her payoff function is

$$u_1(x) = \frac{x}{2}, \quad 0 \leq x \leq 1000$$

  because the receipt of more money will not mean that much.
- The other player is poor, so his utility function is

$$u_2(y) = \ln(y+1) \quad , 0 \leq y \leq 1000,$$

  because small amounts of money mean a lot but the money has less and less impact as he gets more but no more than $ 1000.

EXAMPLE 5.26

- We want to find the bargained solution. The safety points are taken as (0,0) because that is what they get if they can't agree on a split.
- The feasible set is

$$S = \{(x, y) \mid 0 \leq x, y \leq 1000, x + y \leq 1000\}$$

EXAMPLE 5.26 (cont'd)

Figure 5.15 illustrates the feasible set and the contours of the objective function.

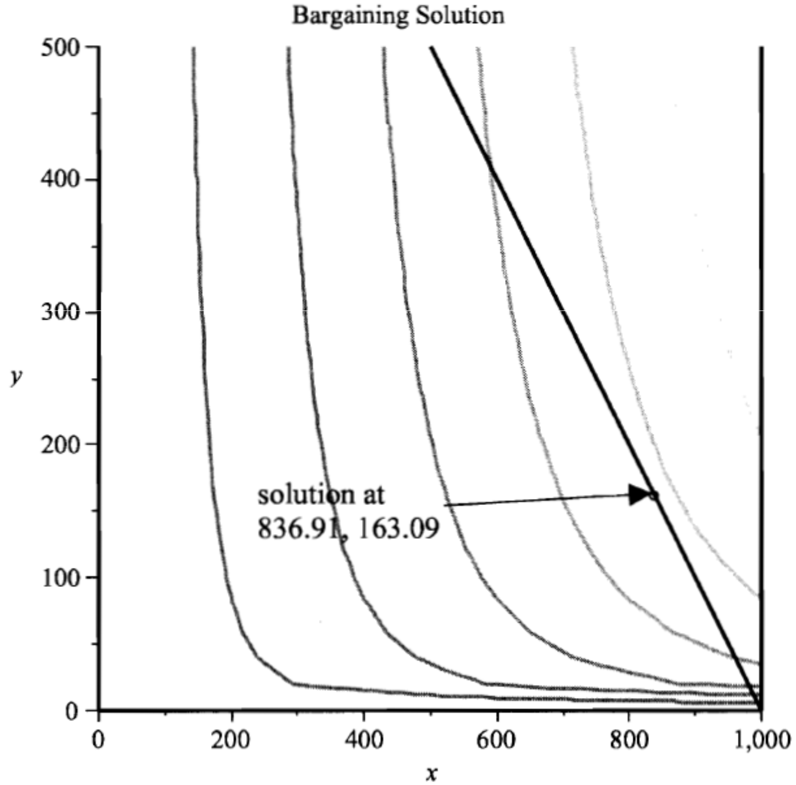


Figure 5.15 Rich and poor split $1000: solution at (836.91, 163.09).

EXAMPLE 5.26 (cont'd)

– The solution is obtained using Maple as follows.

```
> f:=(x,y)->(x/2)*(ln(1+y));
> cnst:={0<=x, x<=1000, 0 <=y, y<=1000, x+y <=1000};
> with(Optimization):
> NLPSolve(f(x,y),cnst,assume=nonnegative,maximize);
```

– Maple tells us that the maximum is achieved at $x = 836.91$, $y = 163.09$, so the poor man gets \$163 while the rich woman gets \$837. Figure 5.15 shows the feasible set as well as several level curves of $f(x, y) = k$.

– The optimal solution is obtained by increasing k until the curve is tangent to the Pareto-optimal boundary.
  - That occurs here at the point (836.91,163.09). The actual value of the maximum is of no interest to us.

5.4.2 Threats Problems

Threats

- Negotiations of the type considered in the previous section do not take into account the relative strength of the positions of the players in the negotiations.

- As mentioned earlier, a player may be able to force the opposing player to play a certain strategy by threatening to use a strategy that will be very detrimental for the opponent. These types of threats will change the bargaining solution.

EXAMPLE 5.27

– We will consider the two-person game with bimatrix

	II_1	II_2
I_1	$(2,4)$	$(-3,-10)$
I_2	$(-8,-2)$	$(10,1)$

– Player I's and II's payoff matrix are

$$A = \begin{bmatrix} 2 & -3 \\ -8 & 10 \end{bmatrix} \quad B^T = \begin{bmatrix} 4 & -2 \\ -10 & 1 \end{bmatrix}$$

– $value(A) = -\frac{4}{23}, value(B^T) = -\frac{16}{17}$

so the security point is

$$(u^*, v^*) = (-\frac{4}{23}, -\frac{16}{17}).$$

EXAMPLE 5.27 (cont'd)

– With this security point we solve the problem

$$\text{Maximize } g(u,v) = \left(u + \frac{4}{23}\right)\left(v + \frac{16}{17}\right)$$
$$\text{subject to } u \geq -\frac{4}{23}, v \geq -\frac{16}{17}, v \geq \frac{11}{13}u - \frac{97}{13},$$
$$v \leq -\frac{3}{8}u + \frac{38}{8}, v \leq \frac{6}{10}u + \frac{28}{10}.$$

In the usual way we get the solution $\overline{u} = 7.501, \overline{v} = 1.937$.

– This is achieved by players I and II agreeing to play the pure strategies (I_1,II_1) 31.2% of the time and pure strategies (I_2,II_2) 68.8% of the time.

– Figure 5.16 below is a three-dimensional diagram of the contours of *g*(*u, v*) over the shaded feasible set. The dot shown on the Pareto boundary is the solution to our problem.

EXAMPLE 5.27 (cont'd)

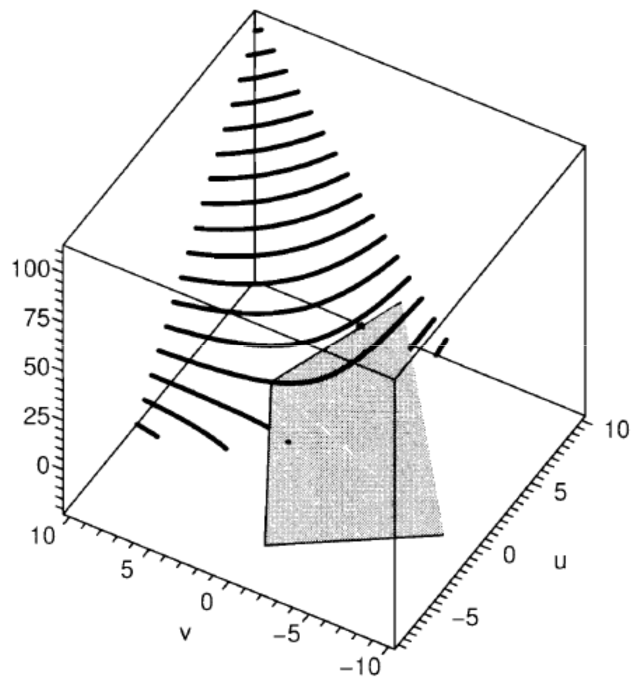


Figure 5.16 The feasible set and level curves in three dimensions. Solution is at $(7.5, 1.93)$ for security point $\left(-\frac{4}{23}, -\frac{16}{17}\right)$.

Finding the Threat Strategies

- In a threat game we replace the security levels (u^*, v^*), which we have so far taken to be the value of the associated games $u^* = value(A), v^* = value(B^T)$, with the expected payoffs to each player if threat strategies are used.

- Suppose that in the bimatrix game player I has a threat strategy X_t and player II has a threat strategy Y_t. The new status quo or security point will be the expected payoffs to the players if they both use their threat strategies:

$$u^* = E_A(X_t, Y_t) = X_t A Y_t^T \text{ and } v^* = E_B(X_t, Y_t) = X_t B Y_t^T.$$

Finding the Threat Strategies (cont'd)

- Then we return to the cooperative bargaining game and apply the same procedure as before but with the new threat security point; that is, we seek to

$$\text{Maximize } g(u,v) := (u - X_t A Y_t{}^T)(v - X_t B Y_t{}^T)$$
$$\text{subject to } (u,v) \in S, u \geq X_t A Y_t{}^T, v \geq X_t B Y_t{}^T.$$

Finding the Threat Strategies (cont'd)

- In the Example 5.27
  - Let's suppose that the threat strategies are $X_t = (0, 1)$ and $Y_t = (1, 0)$
  - Then the expected payoffs give us the safety point

$$u^* = {X_t}^T A Y_t = -8 \text{ and } v^* = X_t B {Y_t}^T = -2$$

  - Changing to this security point increases the size of the feasible set and changes the objective function to $g(u, v) = (u+8)(v+2)$.
  - When we solved this example with the security point $(-\frac{4}{23}, -\frac{16}{17})$ we obtained the payoffs 7.501 for player I, and 1.937 for player II. The solution of the threat problem is $\overline{u} = 5 < 7.501, \overline{v} = 2.875 > 1.937$.
    - This reflects the fact that player II has a credible threat and therefore should get more than if we ignore the threat.

Finding the Threat Strategies (cont'd)

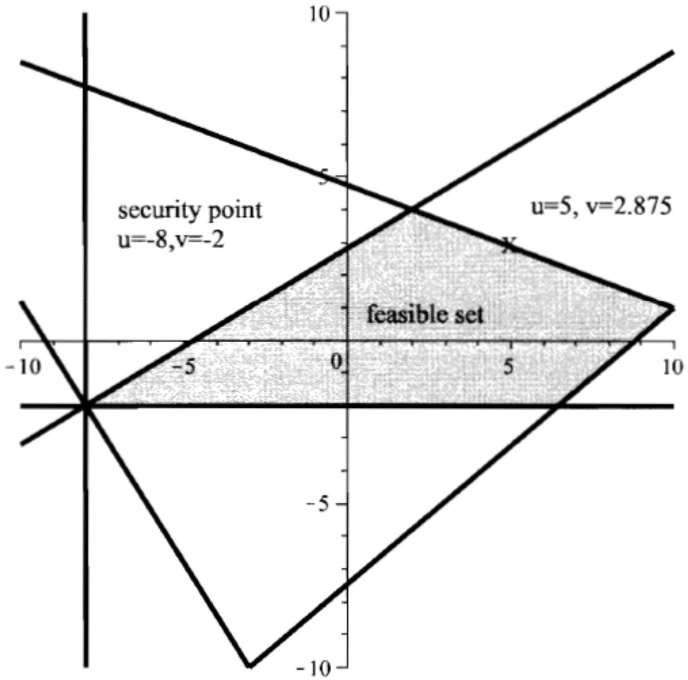


Figure 5.17 Feasible set with security point $(-8, -2)$ using threat strategies.

Finding the Threat Strategies (cont'd)

- The question now is how to pick the threat strategies? How do we know in the previous example that the threat strategies we chose were the best ones? We continue our example to see how to solve this problem.
  - We look for a different security point associated with threats that we call the **optimal threat security point.**

Finding the Threat Strategies (cont'd)

- The Pareto-optimal boundary for our problem is the line segmen $v = -\frac{3}{8}u + \frac{38}{8}, 2 \leq u \leq 10$ with slope $m_p = -\frac{3}{8}$.
  - Consider now a line with slope $-m_p = \frac{3}{8}$ through **any possible threat security point in the feasible set** (u^t, v^t) .
  - Referring to Figure 5.18, the line will intersect the Pareto-optimal boundary line segment at some possible negotiated solution $(\overline{u}, \overline{v})$.
  - The line with slope $-m_p$ through (u^t, v^t) , whatever the point is, has the equation

$$v - v^t = -m_p(u - u^t).$$

Finding the Threat Strategies (cont'd)

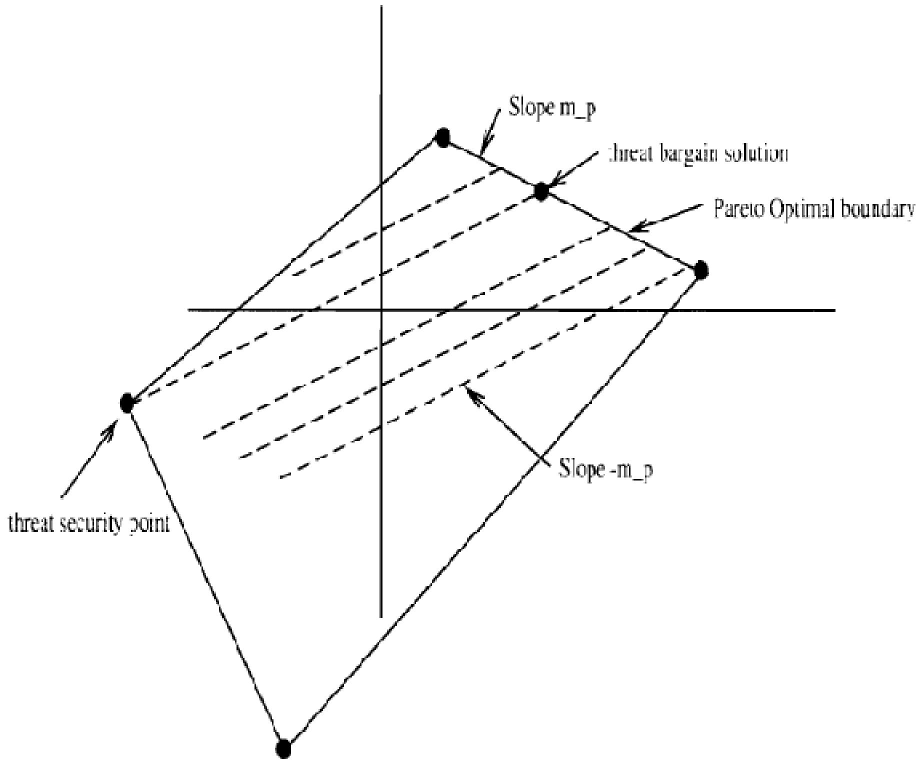


Figure 5.18 Lines through possible threat security points.

Finding the Threat Strategies (cont'd)

– The equation of the Pareto-optimal boundary line is

$$v = m_p u + b = -\frac{3}{8}u + \frac{38}{8},$$

so the intersection point of the two lines will be at the coordinates

$$\overline{u} = \frac{m_p\ u^t + v^t - b}{2m_p} = \frac{3u^t - 8v^t + 38}{6},$$

$$\overline{v} = \frac{1}{2}(m_p u^t + v^t + b) = \frac{-(3u^t - 8v^t) + 38}{16}.$$

Now, remember that we are trying to find the best threat strategies to use, but the primary objective of the players is to maximize their payoffs $\overline{u}, \overline{v}$.

Finding the Threat Strategies (cont'd)

- Player I will **maximize** $\overline{u}$ if she chooses threat strategies to maximize the quantity $m_p\, u^t + v^t = -\frac{3}{8}\, u^t + v^t$.
- Player II will maximize $\overline{v}$ if he chooses threat strategies to **minimize** the same quantity $m_p\, u^t + v^t$ because the Pareto-optimal boundary will have $m_p < 0$, so the sign of the term multiplying u^t will be opposite in $\overline{u}$ and $\overline{v}$.

Summary Approach for Bargaining with Threat Strategies

- Here is the general procedure for finding u^t, v^t and the optimal threat strategies as well as the solution of the bargaining game:

1. Identify the Pareto-optimal boundary of the feasible payoff set and find the slope of that line, call it m_p. This slope should be < 0.

2. Construct the new matrix for a zero sum game

$$-m_p u^t - v^t = -m_p(X_t A Y_t^T) - X_t B Y_t^T = X_t(-m_p A - B)Y_t^T$$

with matrix $-m_p A - B$.

Summary Approach for Bargaining with Threat Strategies (cont'd)

3. Find the optimal strategies X_t, Y_t for that game and compute $u^t = X_t A Y_t^T$ and $v^t = X_t B Y_t^T$. This (u^t, v^t) is the threat security point to be used to solve the bargaining problem.

4. Once we know (u^t, v^t), we may use the following formulas for $(\overline{u}, \overline{v})$:

$$\overline{u} = \frac{m_p\ u^t + v^t - b}{2m_p}, \quad \overline{v} = \frac{1}{2}(m_p u^t + v^t + b). \tag{5.4.1}$$

   Alternatively, we may apply the nonlinear programming method with security point (u^t, v^t) to find $(\overline{u}, \overline{v})$.

EXAMPLE 5.27, continued

– Carrying out these steps for our example, $m_p = -\frac{3}{8}, b = \frac{38}{8}$, we find

$$\frac{3}{8}A - B = \begin{bmatrix} -\frac{26}{8} & \frac{71}{8} \\ -1 & \frac{22}{8} \end{bmatrix}$$

$value(\frac{3}{8}A - B) = -1$

There is a saddle point at the second row and first column, optimal threat strategies $X_t = (0,1), Y_t = (1,0)$.

$u^t = X_t A Y_t^T = -8$, and $v^t = X_t B Y_t^T = -2$

– Once we know that, we can use the formulas above to get

$$\overline{u} = \frac{-\frac{3}{8}(-8) + (-2) - \frac{38}{8}}{2(-\frac{3}{8})} = 5,$$

$$\overline{v} = \frac{1}{2}(-\frac{3}{8}(-8) + (-2) + \frac{38}{8}) = 2.875.$$

EXAMPLE 5.27, continued (cont'd)

– This matches with our previous solution in which we simply took the threat security point to be (-8, -2). Now we see that (-8, -2) is indeed the **optimal threat** security point.

EXAMPLE 5.28

- Consider the cooperative game with bimatrix

	II_1	II_2
I_1	$(-1,-1)$	$(1,1)$
I_2	$(2,-2)$	$(-2,2)$

  - The individual matrices are

$$A = \begin{bmatrix} -1 & 1 \\ 2 & -2 \end{bmatrix}, \qquad B = \begin{bmatrix} -1 & 1 \\ -2 & 2 \end{bmatrix}.$$

  - $value(A) = 0, value(B^T) = 1$
  - The security point for this game is at $(u^*, v^*) = (0,1)$

EXAMPLE 5.28 (cont'd)

– The problem we then need to solve is

$$\text{Maximize } u(v-1),$$
$$\text{subject to } (u,v) \in S^*,$$

$$S^* = \{(u,v) \mid v \leq (-\frac{1}{3})u + \frac{4}{3}, v \leq -3u+4, u \geq 0, v \geq 1\}.$$

– The solution of this problem is at the unique point $(\overline{u}, \overline{v}) = (\frac{1}{2}, \frac{7}{6})$, which you can see in the Figure 5.20.

EXAMPLE 5.28 (cont'd)

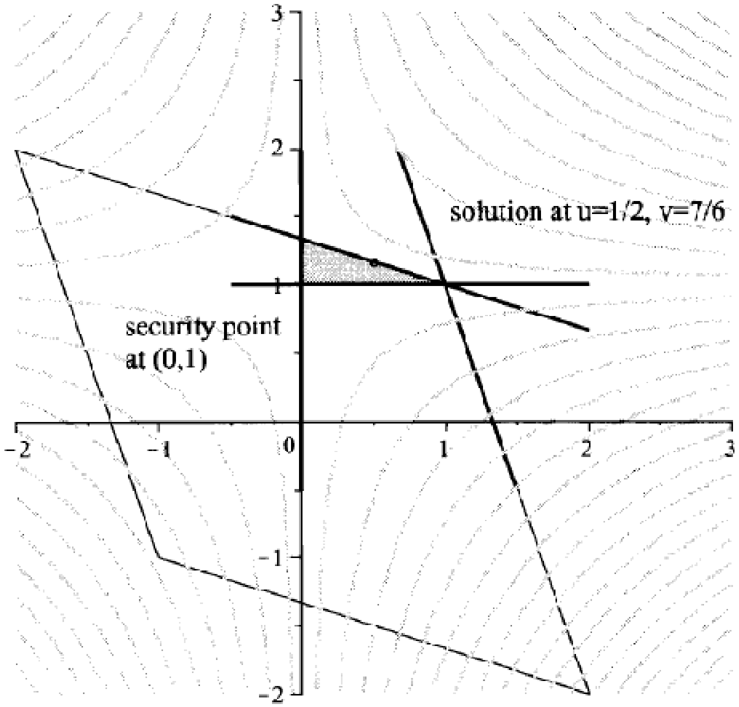


Figure 5.20 Security point $(0, 1)$: cooperative solution $(\overline{u} = \frac{1}{2}, \overline{v} = \frac{7}{6})$.

EXAMPLE 5.28 (cont'd)

- Figure 5.20 was created with the following Maple commands

```
> mypoints:=[[-1, -1], [-2, 2], [1, 1], [2, -2], [-1, -1]];
> constr:={0 <=x,-3*x-y<=4, x+3*y<=4, 3*x+y<=4, -x-3*y<=4, 1<=y};
> z:=(x+0)*(y-1);
> iplot2:=plots[inequal](constr,x=-0.5..2, y=-0.5..2,
    optionsfeasible=(color=white),
    optionsclosed=(color=black, thickness=2),
    optionsexcluded=(color=white),title="Feasible Set
                                          with (0,1) security"):
> pol:=plots[polygonplot](mypoints, color=yellow):
> cp:=plots[contourplot](z, x=-2..3,y=-2..3,
                                   contours=40, axes=boxed,thickness=2):
> plots[display](iplot2, pol,cp);
```

EXAMPLE 5.28 (cont'd)

- The solution of the problem is given by the Maple commands:

```
> with(Optimization):
> NLPSolve(z,constr,maximize);
```

  - We get from these commands that $z = 0.083$, $x = u = 0.5$, $y = v = 1.167$.

EXAMPLE 5.28 (cont'd)

- Next, to find the threat strategies we note that we have two possibilities because we have two line segments in Figure 5.20 as the Pareto-optimal boundary.
- We have to consider both $m_p = -\frac{1}{3}, b = \frac{4}{3}$ and $m_p = -3, b = 4$.

- Let's use $m_p = -3, b = 4$
  - We look for the value of the game with matrix $3\,A - B$:

$$3\,A - B = \begin{bmatrix} -2 & 2 \\ 8 & -8 \end{bmatrix}$$

  - $value(3\,A - B) = 0$
  - The optimal threat strategies are $X_t = (\frac{1}{2}, \frac{1}{2}) = Y_t$

EXAMPLE 5.28 (cont'd)

– Then the security threat points are

$$u^t = X_t A Y_t^T = 0 \text{ and } v^t = X_t B Y_t^T = 0.$$

This means that each player threatens to use (X_t, Y_t) and receive 0 rather than cooperate and receive more.

– Now the maximization problem becomes

$$\text{Maximize } uv,$$
$$\text{subject to } (u, v) \in S^t,$$

$$S^t = \{(u, v) \mid v \leq (-\frac{1}{3})u + \frac{4}{3}, v \leq -3u + 4, u \geq 0, v \geq 0\}.$$

EXAMPLE 5.28 (cont'd)

– The solution of this problem is at the unique point $(\overline{u}, \overline{v}) = (1, 1)$. You can see in Figure 5.21 how the level curves have bent over to touch at the vertex.

EXAMPLE 5.28 (cont'd)

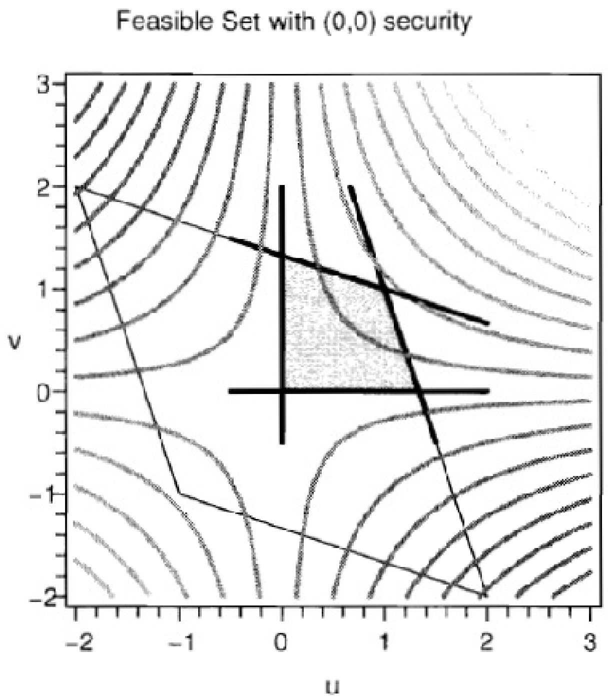


Figure 5.21 Security point $(0, 0)$; cooperative solution $(\overline{u} = 1, \overline{v} = 1)$.

EXAMPLE 5.28 (cont'd)

- Let's look at $m_p = -\frac{1}{3}, b = \frac{4}{3}$
  - The matrix is

$$\frac{1}{3}A - B = \begin{bmatrix} \frac{2}{3} & -\frac{2}{3} \\ \frac{8}{3} & -\frac{8}{3} \end{bmatrix}$$

  - $value(\frac{1}{3}A - B) = -\frac{2}{3}$
  - The security threat points are

$$u^t = X_t A Y_t^T = 1 \text{ and } v^t = X_t B Y_t^T = 1.$$

    This point is exactly the vertex of the feasible set.

EXAMPLE 5.28 (cont'd)

– Now the maximization problem becomes

$$\text{Maximize } (u-1)(v-1),$$
$$\text{subject to } (u,v) \in S^t,$$

$$S^t = \{(u,v) | v \le (-\frac{1}{3})u + \frac{4}{3},\ v \le -3u + 4,\ u \ge 1,\ v \ge 1\}.$$

– But this set has exactly one point, and it is (1,1), so we immediately get the solution $(\overline{u} = 1, \overline{v} = 1)$.

EXAMPLE 5.28 (cont'd)

- What happens if we try to use the formulas (5.4.1) for the threat problem?
  - This question arises now because the contours of *g* are hitting the feasible set right at the point of intersection of two lines.
  - The two lines have the equations

$$v = -3u + 4 \text{ and } v = -\frac{1}{3}u + \frac{4}{3}.$$

  - Let's calculate for both $m_p = -3, b = 4$, or $m_p = -\frac{1}{3}, b = \frac{4}{3}$

EXAMPLE 5.28 (cont'd)

– For $m_p = -3, b = 4, u^t = v^t = 0$, we have

$$\overline{u} = \frac{m_p\ u^t + v^t - b}{2m_p} = \frac{-3(0) + (0) - 4}{2(-3)} = \frac{2}{3},$$

$$\overline{v} = \frac{1}{2}(m_p u^t + v^t + b) = \frac{1}{2}(-3(0) + (0) + 4) = 2.$$

The point $(\frac{2}{3}, 2)$ is not in S^t because $(-\frac{1}{3})(\frac{2}{3}) + \frac{4}{3} = \frac{10}{9} < 2$. So we no longer consider this point. However, because the point $(u^t, v^t) = (0, 0)$ is inside the cone region formed by the lines through $(1, 1)$ with slopes $\frac{1}{3}$ and 3, we know that the threat solution should be $(1, 1)$.

EXAMPLE 5.28 (cont'd)

– For $m_p = -\frac{1}{3}, b = \frac{4}{3}, u^t = v^t = 1,$

$$\overline{u} = \frac{m_p\ u^t + v^t - b}{2m_p} = \frac{-\frac{1}{3}(1) + (1) - \frac{4}{3}}{2(-\frac{1}{3})} = 1,$$

$$\overline{v} = \frac{1}{2}(m_p u^t + v^t + b) = \frac{1}{2}(-\frac{1}{3}(1) + (1) + \frac{4}{3}) = 1.$$

This gives $(\overline{u} = 1, \overline{v} = 1)$, which is the correct solution.

EXAMPLE 5.29

- At the risk of undermining your confidence, this example will show that the Nash bargaining solution can be totally unrealistic, and in an important problem.
  - Suppose that there is a person, Moe, who has owes money to two creditors, Larry and Curly. He owes more than he can pay. Let's say that he can pay at most \$100 but he owes a total of \$150 > \$100 dollars, \$90 to Curly and \$60 to Larry. The question is how to divide the \$100 among the two creditors. We set this up as a bargaining game and use Nash's method to solve it.

EXAMPLE 5.29 (cont'd)

– First, the feasible set is

$$S = \{(u,v) \mid u \leq 60, v \leq 90, u+v \leq 100\},$$

where *u* is the amount Larry gets, and *v* is the amount Curly will get.

– The objective function we want to maximize at first is $g(u,v) = uv$ because if Larry and Curly can't agree on the split, then we assume that they each get nothing.

– For the solution, we want to maximize $g(u,v)$ subject $(u,v) \in S, u \geq 0, v \geq 0$. It is straightforward to show that the maximum occurs at $\overline{u} = \overline{v} = 50$, as shown in Figure 5.22.

EXAMPLE 5.29 (cont'd)

- In fact, if we take any safety point of the form $u^* = a = v^*$. we would get the exact same solution. This says that even though Moe owes Curly \$90 and Larry \$60, they both get the same amount as a settlement. That doesn't seem reasonable, and I'm sure Curly would be very upset.
- Now let's modify the safety point to $u^* = -60$ and $v^* = -90$, which is still feasible and reflects the fact that the players actually lose the amount owed in the worst case, that is, when they are left holding the bag.

EXAMPLE 5.29 (cont'd)

- This case is illustrated in Figure 5.23. The solution is now obtained from maximizing $g(u, v) = (u + 60)(v + 90)$ subject to $(u, v) \in S, u \geq -60, v \geq -90$, and results in $\overline{u} = 60 \text{ and } \overline{v} = 40$.
  - This is ridiculous because it says that Larry should be paid off in full while Curly, who is owed more, gets less than half of what he is owed.

EXAMPLE 5.29 (cont'd)

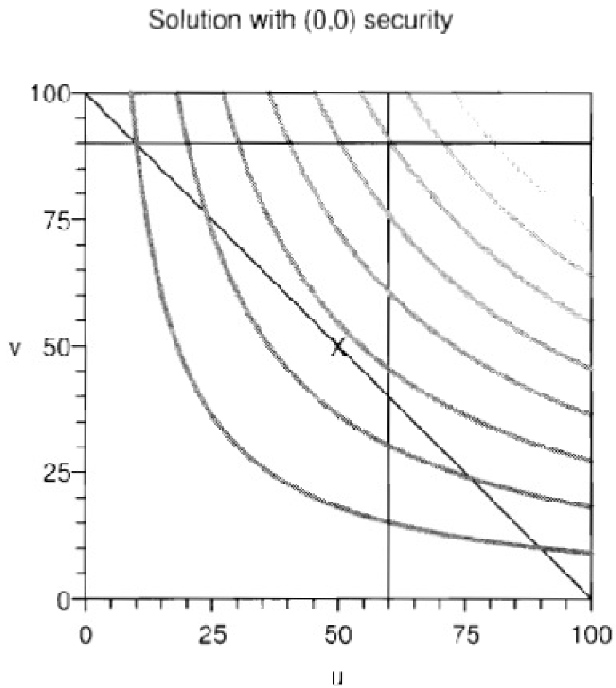


Figure 5.22 Moe pays both Curly and Larry $50 each

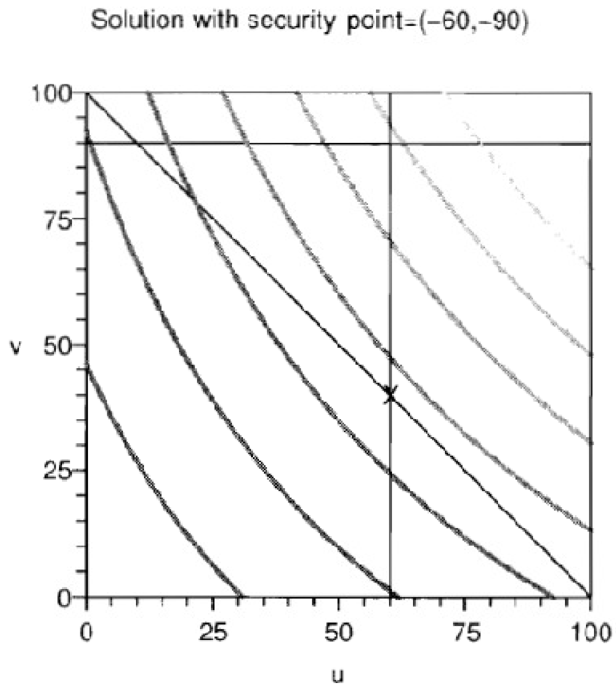


Figure 5.23 Moe pays Larry $60 and pays Curly $40.