Maximal Flow Problems and Cooperative Games

A Senior Comprehensive Project
by
Jennifer R. Johnson
Allegheny College
Meadville,Pa.

April 29, 1997

Submitted to the Department of Mathematics in partial fulfillment of the requirements for the degree of Bachelor of Science.

Project Advisor: David Housman

I hereby recognize and pledge to fulfill my responsibilities, as defined in the Honor Code, and to maintain the integrity of both myself and the College community as a whole.

Pledge:

____

Jennifer R. Johnson

Acknowledgements

There are many people who helped me during the writing of this document. First, I would like to express my deepest thanks to David Housman, who advised me on this project. He provided me with a lot of guidance and encouragement. I would also like to thank my faculty advisor, Rick Holmgren, who has helped me a great deal through my entire undergraduate career.

Finally, I would like to thank my wonderful parents, my sister Allie, my best friend Jen and Melissa without whose love and support I wouldn't be who or where I am today. You guys are the best! Thank you.

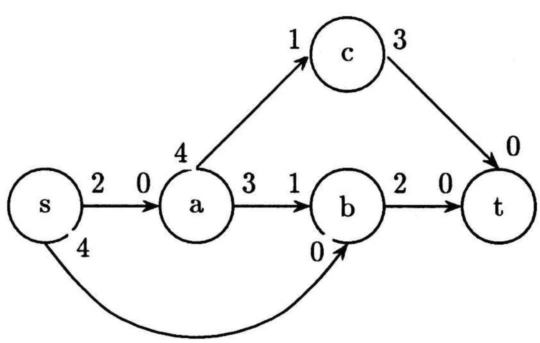
Abstract

This project involves looking at maximal flow problems and cooperative games. First, a maximal flow problem involves a graph that has flow moving along its arcs from a beginning node to an end node. A cooperative game involves a set of players who can combine their assets to form coalitions. Each of the coalitions has a value, the worth of that coalition, and there are methods to determine how to distribute this among the players. Combining these two concepts, we can arrive at a max flow game where the arcs of the graph are players and their worth is the maximal flow through those arcs. Then, we can look at various methods of distribution to the players in these games.

1 Chapter One: Maximal Flow Problems

A *graph* is a set of junction points (nodes) with lines or branches connecting certain pairs of points, and a *network* is a graph with a flow of some type in its branches. The upper limit to the feasible amount of flow in a branch in a specified direction is the *flow capacity* of the branch in that direction. These definitions provide the background for the maximal flow problem. The maximal flow problem involves a connected network having a single source (a node where the flow moves away from it) and a single sink (a node where the flow moves towards it). At all other nodes what flows in must also flow out. The rate (or total quantity) of flow along branch (i, j) from node i to node j can be any nonnegative quantity not exceeding the specified flow capacity c_{ij} , and the objective is to determine the feasible pattern of flows through the network that maximizes the total flow from the source to the sink. In other words, the maximal flow problem seeks to find the largest amount that can be moved from the source to the sink when there are limits on how much can be moved along each branch.

The following is an example of a maximal flow problem where the node designated by s is the source and the node designated by t is the sink:



The numbers on the branches represent the remaining flow capacity on the branch in each direction. For example, 3 units of flow may go along the path from a to b and one unit of flow may go from b to a. The maximal flow that can be assigned to a path from the source to the sink is the smallest remaining flow capacity for any branch on that path. Taking the path s-a-b-t, there are flow capacities of 2, 3, and 2 left on the branches along that path. The maximal flow that can be assigned can only be as great as the smallest capacity along this path, so the maximal flow that can be assigned along this path is 2.

Hillier and Lieberman (1986) present an efficient algorithm that can be used to solve a maximal flow problem. The basic idea behind this algorithm is to keep choosing paths from the source to the sink with positive flow capacities and assigning maximal flows to the paths until no more such paths exist. However, there is one adjustment that the algorithm makes. Because the paths are arbitrarily selected each time, certain better flow assignments may be overlooked. Thus, this algorithm allows for certain assignments to be undone, so other more optimal assignments can be made. The adjustment allows the assignment of fictional flows in the opposing direction on a branch which in reality acts to cancel out all or a portion of the earlier assignment. Therefore, part of the algorithm entails not just decreasing the remaining flow capacity of the branches in the direction of the path, but also increasing the remaining flow capacity in the opposite directions along the branches in the paths. The benefit of this will be demonstrated in the example. The Ford-Fulkerson algorithm is as follows:

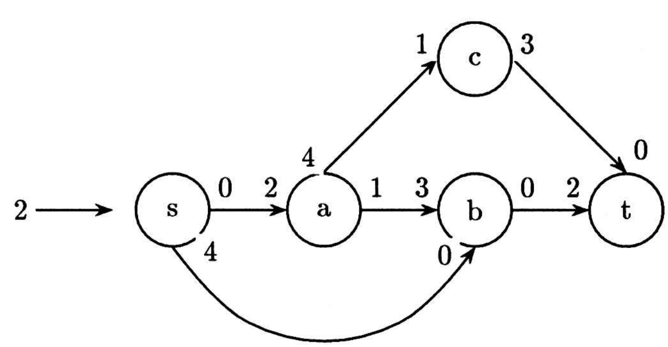
1. Find a path from source to sink with *strictly positive flow capacity.* (If none exists, the net flows already assigned constitute an optimal flow pattern.)

2. Search this path for the branch with the *smallest remaining flow capacity* denote this capacity as c^*, and *increase* the flow in this path by c^*.

3. *Decrease* by c^* the *remaining flow capacity* of each branch in the path. *Increase* by c^* the *remaining flow capacity* in the opposite direction for each branch in the path. Return to step 1.

Now, this algorithm can be applied to the example with an arbitrary path selection each time.

Assign a flow of 2 to s→a→b→t. The network is updated by indicating a flow of 2 entering, decreasing by 2 the remaining flow capacities for each branch along that path and increasing the flow capacities by 2 in the opposite direction for each branch in the path. Then the resulting network is



Assign a flow of 3 to s→b→a→c→t. Then the resulting network is

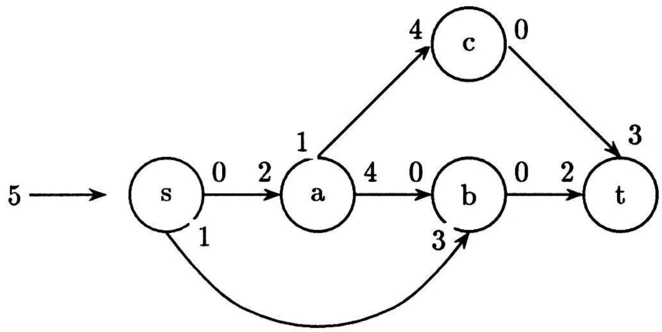
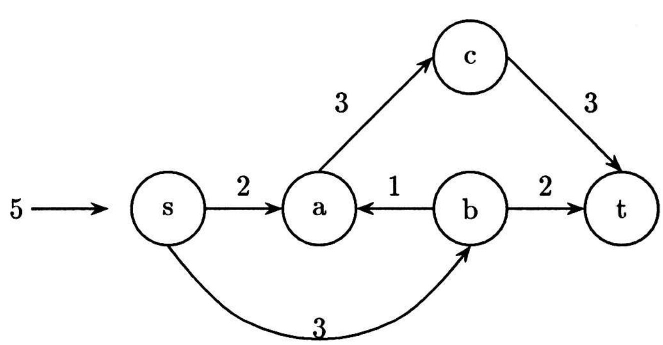


Figure 1: The optimal solution

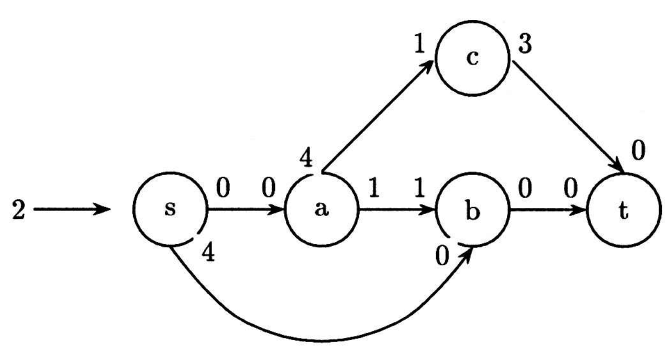
There is no flow available from s→ a. There is one unit of flow available from s→b, but then there is no flow available either from b→a or from b→t. Thus, there are no more paths with a strictly positive flow capacity so optimality has been reached. Thus, the optimal solution to the example is



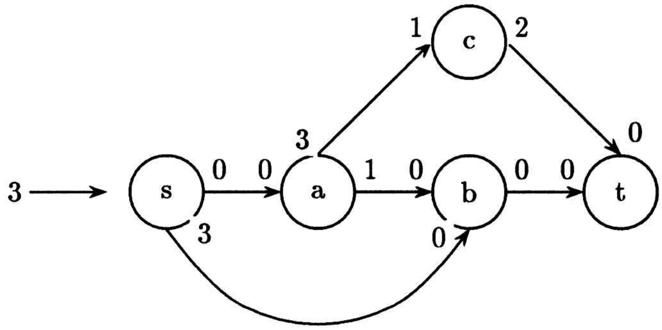
This flow pattern can be arrived at by comparing the final iteration with the original problem. A branch's net flow direction is the direction in which the remaining flow capacity was decreased, and the amount of the net flow is the amount by which that capacity was decreased. For example, at the beginning the branch between a and b had remaining flow capacity of 3 from a→b and of 1 from b→a. At the end, there was a remaining flow capacity of 4 from a→b and 0 from b→a. Thus, the net flow between these branches has a magnitude of 1 and is in the direction of b→a.

An important thing to note about this example is the way it demonstrates the need for adjusting for better flows by allowing previous assignments to be undone. If it were just a matter of arbitrarily assigning maximal flows to paths with positive remaining flow capacities (with no allowance for undoing assignments), then optimality would not have been reached. Following the same iterations as before, but without increasing the remaining flow capacities in the opposite direction would have given way to the following solution.

Assign a flow of 2 to s→a→b→t. Then the resulting network is



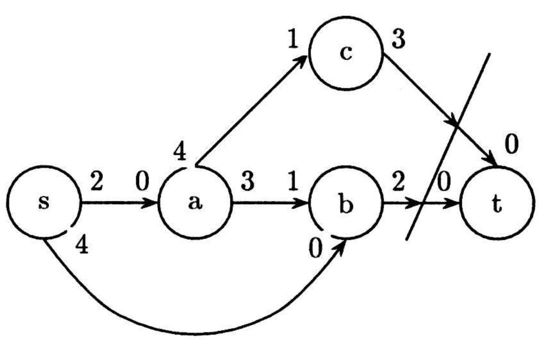
Assign a flow of 1 to s→b→a→c→t. Then the resulting network is



Then, there are no paths with a strictly positive flow capacity left. However, this is not the optimal solution, which is why an adjustment is necessary to allow for undoing assignments that may not lead to optimality because of the arbitrary way in which they were selected. Using the algorithm allowed a flow of 3 to be assigned to s→b→a→c→t instead of just the 1 allowed when not adjusting. Thus, the assignment of a flow of 2 through a→b was undone and an additional assignment of 1 was made through b→a which allowed for the better flow. The final solution had the effect of assigning a flow of 2 to s→a→c→t, a flow of 1 to s→b→a→c→t and a flow of 2 to s→b→t which gives the maximal flow of 5 for the problem.

Hillier and Lieberman also present an important theorem related to the maximal flow problem called the max-flow min-cut theorem. This theorem, again developed by Ford and Fulkerson, states that for any network with a single source and sink, the *maximum feasible flow* from source to sink equals the *minimum cut value* for all the cuts in the network. A *cut* is any set of oriented branches containing at least one branch from every path from source to sink. The *cut value* is defined as the sum of the flow capacities of the branches (in the specified direction) of the cut. This theorem provides an upper bound to the amount of flow from source to sink. The maximum value of the amount of flow is the smallest of the cut values. So, if in the original network a cut can be found to equal the amount of flow in a current solution, optimality has been reached. When networks are increasingly large, this can help to prevent a search for a path with positive flow when there are no such paths left. Or if a cut can be found in a current solution with a value of 0 regarding the remaining flow capacities then optimality has also been reached. This can be demonstrated with the example as well.

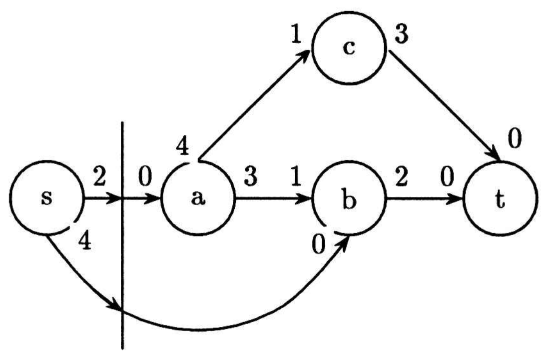
Looking at the original problem,



The value of this cut is $(3 + 2) = 5$, which was the maximum value of the amount of flow from the source to the sink. Then, this cut is a minimal cut. Also, looking back at the final iteration (Figure 1) in finding the maximal flow for the example, this same cut would have a value equal to

0 for the remaining flow capacities in that direction. Either way, the upper bound was indeed reached signalling optimality.

An example of a cut that is not minimal is:



The value of this cut is $(2 + 4) = 6$, which shows an upper limit to the maximal flow through the problem, but it is not the minimal cut.

The maximal flow problem provides an interesting basis for further exploration. We can formally define it (in a more specialized way) as follows: a *maximal flow problem* consists of a finite, nonempty set V, called vertices; a set A of ordered pairs of vertices called arcs; a source $s \in V$; a sink $t \in V - \{s\}$; and a function $c : A \to \mathbb{R}_+$,where we call $c(a)$ the capacity of arc a. This definition is more specialized than the problems shown so far because it restricts the flow to a single direction. Further, the capacity is expressed in relation to the set of arcs rather than the vertices. Earlier, we expressed capacity as $c_{i,j}$, but in this problem where flow is restricted to one direction, it suffices to express capacity in relation to the arcs themselves. We will use this later when we turn the maximal flow problem into a cooperative game. In order to investigate this concept further, it is necessary to turn our attention to cooperative games. The next chapter will provide the background needed to relate the maximal flow problem to a cooperative game.

2 Chapter Two: Cooperative Games

A cooperative game involves a set $N = \{1, 2, \ldots, n\}$, and a function w from nonempty subsets of N to real numbers. A *player* is an element $i \in N$, a *coalition* is a nonempty subset S of N, and the *worth* of the coalition S is $w(S)$. The worth is how much the players in S benefit by a cooperative effort regardless of the actions of the players in N but not in S. For a game with $N = \{1, 2, 3\}$, the proper notation for the worth function is $w(\{1, 2, 3\})$, but this will be written as $w(123)$. Our example game has $w(123) = 630$, $w(12) = 360$, $w(13) = 330$, $w(23) = 120$, and $w(1) = w(2) = w(3) = 0$.

Now one major question that cooperative game theory explores is determining how much each player should receive as a payoff for playing the game. An *allocation* or *payoff vector* for the game (N, w) is a vector $x \in \mathbb{R}^n$, the set of n-tuples of real numbers, so the ith player receives a payoff of x_i. An *allocation method* or *value* is a function ϕ from games to allocations, and a *solution* is a function σ from games to subsets of allocations. Again, the proper notation is $\phi[(N, w)]$, but this will be written simply as $\phi[w]$. Two methods for determining allocations are the Shapley value and the nucleolus. We will be following Straffin(1993) in looking at both of these.

First there are some necessary definitions. An *imputation* of the game (N, w) is an n-dimensional allocation vector x that satisfies two conditions. The first condition is collective rationality: $\sum_{i \in N} x_i = w(N)$. The second is individual rationality: $x_i \geq w(i)$ for all $i \in N$. A *preimputation* satisfies the first condition of collective rationality, but not necessarily the second. Shapley sought to determine a fair allocation method, φ, that obtains preimputations which satisfy three fairness axioms.

Axiom 1. φ should depend only on w, and should respect any symmetries in w. So if players i and j have symmetric roles in w, that is, if $w(S-i) = w(S-j)$ whenever S is a coalition with $i, j \in S$, then $\varphi_i[w] = \varphi_j[w]$.

Axiom 2. If $w(S) = w(S-i)$ for all coalitions $S \subseteq N$, that is, if player i is a *dummy* who doesn't add any value to the coalition, then $\varphi_i[w] = 0$. Adding a dummy player to a game doesn't change the value of $\varphi_j[w]$ for other players j in the game.

The final axiom involves the concept of adding two games together. If there are two games (N, v) and (N, w) with the same set N of players, then the *sum game* $v+w$ is defined as $(v+w)(S) = v(S) + w(S)$ for all coalitions S. Then there are three games and the preimputations for these games are $\varphi[v]$, $\varphi[w]$, and $\varphi[v+w]$. This gives way to the third axiom involving the property of additivity.

Axiom 3. $\varphi[v+w] = \varphi[v] + \varphi[w]$.

Using these three axioms, Shapley could prove an important theorem.

Theorem 1 (Shapley, 1953). *There is one and only one allocation method of assigning a preimputation to a game which satisfies Axioms 1,2, and 3.*

Proof. Sketch of the proof: For a game (N, w), Shapley demonstrated how a unique preimputation $\varphi[w]$ is forced by these axioms by breaking down an arbitrary game into a sum of games all of whose players play symmetric roles or are dummies. The example mentioned earlier can be used to illustrate the method that Shapley used.

$$w(1) = w(2) = w(3) = 0$$

$$w(12) = 360 \quad w(13) = 330 \quad w(23) = 120$$

$$w(123) = 630$$

For any coalition $S \subseteq N$ and for a real number α, let αw^S be the game defined as follows:

$$\alpha w^S(T) = \begin{cases} \alpha & \text{if } T \supseteq S \\ 0 & \text{if } T \not\supseteq S \end{cases}$$

So, T needs to contain all the members of S in order to get α units, else it gets 0. For example, $360w^{\{12\}}(T) = 360$, if $T = \{1, 2\}$ or $\{1, 2, 3\}$. For all other T, $360w^{\{12\}}(T) = 0$. All the players in S have symmetric roles in αw^S, and all the players not in S are dummies. Thus, Axioms 1 and 2 give

$$\varphi_i[\alpha w^S] = \begin{cases} \alpha/s & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

Here, s is the number of players in S. For example, $\varphi_i[360w^{\{12\}}] = \frac{360}{2} = 180$ if $i = 1, 2$ else $\varphi_i[360w^{\{12\}}] = 0$. Starting with the original game w, look at a new game

$$w' = w - 360w^{\{12\}} - 330w^{\{13\}} - 120w^{\{23\}}$$

For this game,

$$w'(1) = w'(2) = w'(3) = w'(12) = w'(13) = w'(23) = 0$$

$$w'(123) = 630 - 360 - 330 - 120 = -180$$

So, $w' = -180w^{\{123\}}$, which yields

$$w = 360w^{\{12\}} + 330w^{\{13\}} + 120w^{\{23\}} - 180w^{\{123\}}$$

Axiom 3 gives $\varphi[w] = \varphi[360w^{\{12\}}]+\varphi[330w^{\{13\}}]+\varphi[120w^{\{23\}}]+\varphi[-180w^{\{123\}}]$

$$\varphi_1[w] = \frac{360}{2} + \frac{330}{2} + 0 + \frac{-180}{3} = 285$$

$$\varphi_2[w] = \frac{360}{2} + 0 + \frac{120}{2} + \frac{-180}{3} = 180$$

$$\varphi_3[w] = 0 + \frac{330}{2} + \frac{120}{2} + \frac{-180}{3} = 165$$

The three axioms forced this unique preimputation, and the method works for any general game. □

This value, the Shapley value, can be computed in other ways as Shapley points out. The players form coalitions adding one player at a time. As a new player joins the coalition, that person should receive the value that her joining adds to the coalition. What the players receive can be considered as a preimputation. It must also be considered that there are $n!$ different ways that the coalition of all n players could form since order does matter. If the preimputations given by all these different orderings were to be averaged, we would find φ.

The example can again be used to demonstrate this. All the possible orders are listed, and the values added by each member joining the coalition in each order are determined.

Order	1	2	3
123	0	360	270
132	0	300	330
213	360	0	270
231	510	0	120
312	330	300	0
321	510	120	0
	----	----	----
	1710	1080	990

$$\varphi[w] = \frac{1}{6}(1710, 1080, 990) = (285, 180, 165)$$

which is the same as the answer arrived at using the other method. In order to arrive at the above numbers, take the order 321 as an example:

$$3 : w(3) - w(\emptyset) = 0 - 0 = 0$$

$$2 : w(23) - w(3) = 120 - 0 = 0$$

$$1 : w(123) - w(23) = 630 - 120 = 510$$

So, as each player joins, the value that player receives is the value of the coalition after she joins minus the value of the coalition before she joins. This way to conceptualize the Shapley value is that the value player i receives is the average amount the player adds to the grand coalition with all ordering formations being just as likely. So each player receives her average contribution. Trying to complete such a table for bigger games would be quite an enormous task, but a simpler way to calculate the values this way is to look at individual players. When a player i joins the coalition, she along with the other players who have joined make up a coalition S of size s. The value of i's contribution is $w(S) - w(S - i)$, and this occurs for only those orderings where the $s - 1$ other players in S joined before i, followed by the $n - s$ players in $N \backslash S$. The number of orderings where this occurs is $(s-1)!(n-s)!$. So the Shapley value for player i can be written as follows where s is the size of S and n is the size of N:

$$\varphi_i(N, w) = \frac{1}{n!} \sum_{i \in S} (s-1)!(n-s)![w(S) - w(S - i)]$$

For example, we can again look at φ_1.

Coalition S	$(s-1)!(n-s)!$	$w(S)-w(S-i)$	Product
1	$1\times 2=2$	$0-0=0$	0
12	$1\times 1=1$	$360-0=360$	360
13	$1\times 1=1$	$330-0=330$	330
123	$2\times 1=2$	$630-120=510$	1020

			1710

$$\varphi_1[w] = \frac{1}{6}(1710) = 285$$

.

The Shapley value is based on a concept of fairness, but there are other considerations that can give way to solutions as well. The *nucleolus*, which David Schmeidler proposed, is an imputation that is based on the idea of bargaining. The *core* is the set of all imputations $x = (x_1, \ldots, x_n)$ which satisfy $\sum_{i\in S} x_i \geq w(S)$ for every $S \subseteq N$. The core of some games may be empty, in which case, there is no imputation that satisfies these constraints. However, Schmeidler sought to make the largest violation as small as possible. The *excess* of S at x for every imputation x and coalition $S \subseteq N$ is defined as $e_S(x) = w(S) - \sum_{i\in S} x_i$. This measures the difference between how much S could get if they broke from the grand coalition and what x gives them so it could be viewed as their unhappiness with or complaint against x.

The nucleolus is the imputation that lexicographically minimizes the largest complaint. In other words, for a game (N, w) and payoff vector $x = (x_1, \ldots, x_n)$, the 2^n-vector $\theta(x)$ is defined as the vector composed of the excesses of the 2^n subsets $S \subset N$ in decreasing order. $\theta_k(x) = e(S_k, x)$ where $S_1, S_2, \ldots S_{2^n}$ are the subsets of N, arranged by $e(S_k, x) \geq e(S_{k+1}, x)$. The vectors $\theta(x)$ can be ordered lexicographically as Jianhua(1988) shows. Given two vectors $\alpha = (\alpha_1, \ldots, \alpha_q)$ and $\beta = (\beta_1, \ldots, \beta_q)$, α is lexicographically smaller than β if there is some integer k, $1 \leq k \leq q$, such that

$$\alpha_l = \beta_l \quad \text{for} \quad 1 \leq l \leq k,$$

$$\alpha_k < \beta_k.$$

This relation can be written as $\alpha <_L \beta$. We write $\alpha \leq_L \beta$ if $\alpha <_L \beta$ or $\alpha = \beta$. These relationships can be used to order the allocation vectors. Thus,

$$x \preceq y$$

if and only if $\theta(x) \leq_L \theta(y)$ and

$$x \prec y$$

if an only if $\theta(x) <_L \theta(y)$. The nucleolus can then be defined using these orderings. In a game (N, w) with a set of allocation vectors X, the nucleolus of w over the set X is the set $\nu(X)$ defined by

$$\nu(X) = \{x \in X | \text{if} \quad y \in X, \text{ then} \quad x \preceq y\}.$$

The nucleolus of w over the set of imputations X^1 is the set $\nu(X^1)$ defined by

$$\nu(X^1) = \{x \in X^1 | \text{if} \quad y \in X^1, \text{then} \quad x \preceq y\}.$$

The *prenucleolus* in the game (N, w) over the set of preimputations X^0 is the set $\nu(X^0)$ defined by

$$\nu(X^0) = \{x \in X^0 | \text{if} \quad y \in X^0, \text{then} \quad x \preceq y\}.$$

Going back to the example, we can try to minimize the maximal coalitional complaints and determine the nucleolus. Starting with some imputation, say $x = (340, 150, 140)$, we find the excesses of the coalitions. We try to lower the largest excess, the second largest excess, the third largest excess until we get to the unique imputation that is the nucleolus.

$$\begin{aligned}
e_1(x) &= 0 - 340 &&= -340 \\
e_2(x) &= 0 - 150 &&= -150 \\
e_3(x) &= 0 - 140 &&= -140 \\
e_{12}(x) &= 360 - (340 + 150) &&= -130 \\
e_{13}(x) &= 330 - (340 + 140) &&= -150 \\
e_{23}(x) &= 120 - (150 + 140) &&= -170 \\
e_{123}(x) &= 630 - (340 + 150 + 140) &&= 0
\end{aligned}$$

The largest coalitional complaint is 0 but that's always the complaint of the grand coalition so it cannot be lowered. Now, $e_{12}(x) = -130$ is the largest coalitional complaint that can be lowered by giving more to coalition 12 and taking from coalition 3. Since $e_{13} > e_{23}$, it should be taken from player 3 and given to player 1. Now, $e_{12} = -130$ and $e_3 = -140$, the best that can be done is to make these equal by taking 5 from player 3 and giving it to player 1. Thus, we now have $y = (345, 150, 135)$, with

$$
\begin{aligned}
e_1(y) &= 0 - 345 && = -345 \\
e_2(y) &= 0 - 150 && = -150 \\
e_3(y) &= 0 - 135 && = -135 \\
e_{12}(y) &= 360 - (345 + 150) && = -135 \\
e_{13}(y) &= 330 - (345 + 135) && = -150 \\
e_{23}(y) &= 120 - (150 + 135) && = -165 \\
e_{123}(y) &= 630 - (345 + 150 + 135) && = 0
\end{aligned}
$$

The excesses of coalitions 3 and 12 can't be lowered simultaneously since lowering one would raise the other. Neither can the excesses of 2 or 13 be lowered. Since every excess has been minimized as much as possible, we are down to the unique imputation y which is the nucleolus.

Kohlberg proves a theorem which allows for an interesting characterization of both the nucleolus and the prenucleolus.

Theorem 2 (Kohlberg, 1971). *Let $\mathcal{B}_1, \mathcal{B}_2, \ldots$ be the sets of coalitions of highest excess at x, second highest, third highest, etc. Let $\mathcal{D}_t = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_t$. A necessary and sufficient condition for x to be the prenucleolus is that each $\mathcal{D}_t$ is a balanced collection.*

A collection $\mathcal{C} = \{S_1, S_2, \ldots, S_m\}$ of subsets of N is called *balanced* if there exist positive real numbers (called balancing weights) $\lambda_1, \lambda_2, \ldots, \lambda_m$ such that, for each $i \in N$,

$$\sum_{\substack{j \\ i \in S_j}} \lambda_j = 1.$$

For example, the collection $\{\{1\}, \{2,3\}, \{2,4\}, \{2,5\}, \{3,4,5\}\}$ is balanced with $\lambda = (1, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3})$. However, $\{\{1,2,3\}, \{3,4,5\}\}$ is not balanced since $\lambda_1 = 1$ and $\lambda_1 + \lambda_2 = 1$, which is impossible since each λ_i was required to be a positive number.

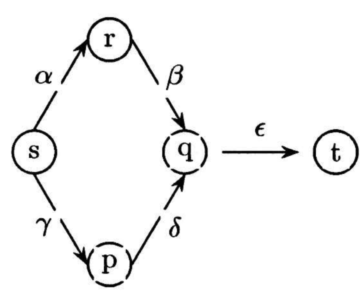
Proposition 1 (Housman, 1989). *If (N, w) is a superadditive game, then the prenucleolus and nucleolus yield the same imputation.*

So, if a game (N, w) satisfies $w(S)+w(T) \leq w(S \cup T)$ for all disjoint coalitions S and T, and if each $\mathcal{D}_t$ is shown to be a balanced collection, then this leads to both the prenucleolus and the nucleolus.

Consider the example with y. The sets of coalitions with descending excesses are $\mathcal{B}_1 = \{\{1,2\}, \{3\}\}, \mathcal{B}_2 = \{\{1,3\}, \{2\}\}, \mathcal{B}_3 = \{\{2,3\}\}, \mathcal{B}_4 = \{\{1\}\}$. Then, $\mathcal{D}_1$ is balanced with $\lambda = (1,1), \mathcal{D}_2$ is balanced with $\lambda = (.5, .5, .5, .5), \mathcal{D}_3$ is balanced with $\lambda = (.5, .25, .5, .25, .25)$, and $\mathcal{D}_4$ is balanced with $\lambda = (.25, .5, .25, .5, .25, .5)$. Thus, each $\mathcal{D}_t$ is a balanced collection which verifies that this is indeed the nucleolus. The example with x is not balanced because $\mathcal{B}_1 = \{\{1,2\}\}$ does not contain any set with 3 in it.

3 Chapter Three: Maximal Flow Problems as Cooperative Games

Setting up the concept of a maximal flow problem and cooperative games leads to the idea of a max flow game. Recall that a *maximal flow problem* consists of a finite, nonempty set V, called vertices; a set A of ordered pairs of vertices called arcs; a source $s \in V$; a sink $t \in V - \{s\}$; and a function $c : A \to \mathbb{R}_+$,where we call $c(a)$ the capacity of arc a. Given a max flow problem (V, A, s, t, c), we define the associated *max flow game* by (A, v) where $v(S)$ is the max flow in the problem (V, S, s, t, c). Thus, the arcs are the players and the worth of a coalition is the maximal flow along the paths defined by the players in the coalition. Here's an example:



This is a five player game with $V = \{s, r, q, p, t\}$ and $A = \{\alpha, \beta, \gamma, \delta, \epsilon\}$. We denote the capacities as follows: $c(\alpha) = a$, $c(\beta) = b$, $c(\gamma) = c$, $c(\delta) = d$, and $c(\epsilon) = e$. Then, the worths of the various coalitions are as follows: $v(\alpha\beta\epsilon) = v(\alpha\beta\gamma\epsilon) = v(\alpha\beta\delta\epsilon) = min\{a, b, e\}$, $v(\alpha\gamma\delta\epsilon) = v(\beta\gamma\delta\epsilon) = v(\gamma\delta\epsilon) = min\{c, d, e\}$, $v(\alpha\beta\gamma\delta\epsilon) = min\{(min\{a, b\} + min\{c, d\}), e\}$, and $v(S) = 0$ for all other coalitions.

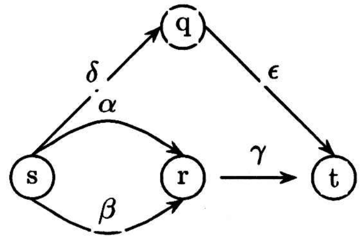
3.1 General Results

We will first look at some results involving these max flow games, and then look at the Shapley values and nucleolus values for general games involving one to four players.

Theorem 1. *Max flow games are superadditive. That is, for a max flow game (A, v), it follows that $v(S) + v(T) \leq v(S \cup T)$ for all disjoint coalitions S and T.*

Proof. If $v(S) = s$ and $v(T) = t$, then the maximal flow along the paths defined by the arcs in S and T are s and t, respectively. Now, S and T are disjoint coalitions, so for an $a \in S$, $a \notin T$. So, the flow along the arcs in S is distinct from the flow along the arcs in T, and thus, the maximal flow found along the paths defined by the arcs in $S \cup T$ must be at least as much as the maximal flow through S and T individually, or $v(S \cup T) \geq s + t$. It may be that the union of S and T allows more flow through by completing a path from the source to the sink, but the maximal flow of $S \cup T$ is at least as great as the maximal flow of S and T. That is, $v(S) + v(T) \leq v(S \cup T)$. □

The max flow problem $(V, A, s, t, c) = M$ is *separable* if there are max flow problems $(V_1, A_1, s, t, c) = M_1$ and $(V_2, A_2, s, t, c) = M_2$ such that $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{s, t\}$, $A_1 \cup A_2 = A$, and $A_1 \cap A_2 = \emptyset$. Here's an example of a max flow problem that is separable:



In this original problem, we have $M = (V, A, s, t, c)$ with $V = \{s, r, q, t\}$, and $A = \{\alpha, \beta, \gamma, \delta, \epsilon\}$. This is separable into $M_1 = (V_1, A_1, s, t, c)$ with $V_1 = \{s, q, t\}$ and $A_1 = \{\delta, \epsilon\}$ and $M_2 = (V_2, A_2, s, t, c)$ with $V_2 = \{s, r, t\}$ and $A_2 = \{\alpha, \beta, \gamma\}$ since M_1 and M_2 satisfy the criteria. That is, $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{s, t\}$, $A_1 \cup A_2 = A$, and $A_1 \cap A_2 = \emptyset$.

Using this definition of separability leads to the following theorem:

Theorem 2. *Suppose M is separable into M_1 and M_2 and $(A, v), (A_1, v_1), (A_2, v_2)$ are the max flow games associated with M, M_1, M_2, respectively. Then, $v(S) = v_1(S \cap A_1) + v_2(S \cap A_2)$ for all $S \subseteq A$.*

Proof. Now, $v(S) = v(S \cap A)$ since $S \subseteq A$. Since $A = A_1 \cup A_2$, $v(S \cap A) = v(S \cap (A_1 \cup A_2)) = v((S \cap A_1) \cup (S \cap A_2))$. Then by the superadditivity of flow games, $v((S \cap A_1) \cup (S \cap A_2)) \geq v(S \cap A_1) + v(S \cap A_2)$, so $v(S) \geq v(S \cap A_1) + v(S \cap A_2)$. Now, $S \cap A_1 \subseteq A_1$ and $S \cap A_2 \subseteq A_2$ so $v(S \cap A_1) = v_1(S \cap A_1)$ and $v(S \cap A_2) = v_2(S \cap A_2)$. Thus, we have shown that $v(S) \geq v_1(S \cap A_1) + v_2(S \cap A_2)$.

We will now show that $v(S) \leq v_1(S \cap A_1) + v_2(S \cap A_2)$. From the max-flow min-cut theorem, we know that a cut of S provides an upper bound on $v(S)$. If we can show that a cut of $S \cap A_1$ plus a cut of $S \cap A_2$ is actually a cut of S, then we could show the other inequality. First, we know that there is a cut of $S \cap A_1$ with a cut value of $v_1(S \cap A_1)$ and a cut of $S \cap A_2$ with a cut value of $v_2(S \cap A_2)$ by the max-flow min-cut theorem. Now, let's assume there is some path in S not in either of these cuts (that is, these two cuts combined are not a cut of S). The path can't be all in A_1 or all in A_2 or it would be included in the cuts of $S \cap A_1$ or $S \cap A_2$. So, it must be that part of the path is in A_1 and part is in A_2. This means there is some w such that $w \in V_1 - \{s,t\}$ and $w \in V_2 - \{s,t\}$. However, by definition of separability, $V_1 \cap V_2 = \{s,t\}$, so this is impossible. Thus, it must be that the combined cuts of $S \cap A_1$ and $S \cap A_1$ are a cut of S. By the max-flow min-cut theorem, $v(S)$ is less than or equal to the value of all cuts of S. So, $v(S) \leq$ the combined cut values $S \cap A_1$ and $S \cap A_2$. That is, $v(S) \leq v_1(S \cap A_1) + v_2(S \cap A_2)$.

We have shown both that $v(S) \geq v_1(S\cap A_1)+v_2(S\cap A_2)$ and $v(S) \leq v_1(S\cap A_1)+v_2(S\cap A_2)$. Thus, it must be that $v(S) = v_1(S\cap A_1)+v_2(S\cap A_2)$ and we are done.

□

Theorem 3. *If a given max flow problem (V, A, s, t, c) is separable, then*

$$\varphi_i(v) = \begin{cases} \varphi_i(v_1), & \text{if } i \in A_1 \\ \varphi_i(v_2), & \text{if } i \in A_2 \end{cases} \tag{1}$$

Proof. Since (V, A, s, t, c) is separable, there exist max flow problems (V_1, A_1, s, t, c) and (V_2, A_2, s, t, c) such that $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{s,t\}$, $A_1 \cup A_2 = A$, and $A_1 \cap A_2 = \emptyset$. Now, the general formula to find the Shapley value is:

$$\varphi_i(v) = \frac{1}{n!}\sum_{i\in S}(s-1)!(n-s)![v(S) - v(S-i)]$$

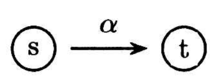
$$= \frac{1}{n!}\sum_{i\in S}(s-1)!(n-s)![v_1(S\cap A_1)+v_2(S\cap A_2)-v_1(S\cap A_1-i)-v_2(S\cap A_2-i)]$$

since we have just shown $v(S) = v_1(S\cap A_1)+v_2(S\cap A_2)$. Now, if $i \in S\cap A_1$, then this equals $\frac{1}{n!}\sum_{i\in(S\cap A_1)}(s-1)!(n-s)![v_1(S\cap A_1)-v_1(S\cap A_1-i)]$ since $i \in S\cap A_1 \Rightarrow i \notin A_2 \Rightarrow i \notin S\cap A_2 \Rightarrow v_2(S\cap A_2) = v_2(S\cap A_2 - i)$. This is, of course, $\varphi_i(v_1)$. Similarly, we can show that if $i \in A_2$, then $\varphi_i(v) = \varphi_i(v_2)$. We have shown that (1) holds. □

I conjecture that a similar result holds true for the nucleolus.

3.2 One and Two Player Games

Now, we can look at the various one to four player games that are defined by various max flow problems. First, looking at all max flow problems with one arc:



This is the simplest max flow problem, with $V = \{s,t\}$, and $A = \{\alpha\}$. This would yield a max flow game where α is the only player. Then, $v(\alpha) = c(\alpha)$. For convenience, we shall think of $c(\alpha) = a$. The Shapley value is $\varphi = (a)$ and the nucleolus is $\nu = (a)$.

There are two max flow problems with two arcs:

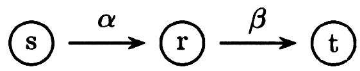


Figure 1: The first max flow problem with two arcs

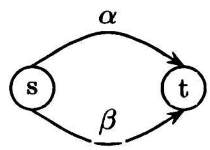


Figure 2: The second max flow problem with two arcs

Now, the max flow problem defined by Figure 1 has $A = \{\alpha, \beta\}$ and $V = \{s, r, t\}$. Again, $c(\alpha) = a$, and $c(\beta) = b$. The max flow game is (A, v). Then, $v(\alpha) = v(\beta) = 0$ and $v(\alpha\beta)$ =min$\{a, b\}$. From now on, the minimum of numbers will be indicated by a $\wedge$. That is, the min$\{a, b\}$ will be written as: $a \wedge b$. The Shapley value for both α and β is $\frac{a\wedge b}{2}$ so $\varphi = (\frac{a\wedge b}{2}, \frac{a\wedge b}{2})$. Similarly, the nucleolus is $\nu = (\frac{a\wedge b}{2}, \frac{a\wedge b}{2})$.

The max flow problem defined by Figure 2 has $A = \{\alpha, \beta\}$ and $V = \{s, t\}$. The max flow game is (A, v), but this is separable into the single player games with α as the sole player in one game and β as the sole player in another. By Theorem 2, and the one player results, it follows that $\varphi = (a, b)$ and $\nu = (a, b)$ in this game.

3.3 Three Player Games

There are four basic maximal flow problems with 3 arcs.

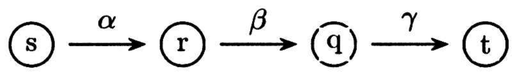


Figure 3: The first max flow problem with three arcs

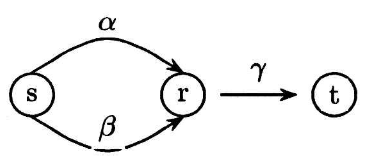


Figure 4: The second max flow problem with three arcs

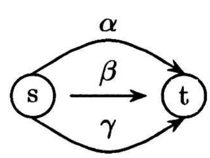


Figure 5: The third max flow problem with three arcs

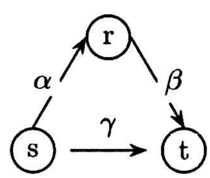


Figure 6: The fourth max flow problem with three arcs

The max flow problem defined by Figure 3 has $A = \{\alpha, \beta, \gamma\}$ and $V = \{s, r, q, t\}$. The capacities again shall be denoted as follows: $c(\alpha) = a$, $c(\beta) = b$, and $c(\gamma) = c$. The max flow game is (A, v) with $v(\alpha\beta\gamma) = a \wedge b \wedge c$ and $v(S) = 0$ for all $S \subset A$. Again this leaves us with $\varphi = \nu = (\frac{a \wedge b \wedge c}{3}, \frac{a \wedge b \wedge c}{3}, \frac{a \wedge b \wedge c}{3})$.

The max flow problem defined by Figure 5 has $A = \{\alpha, \beta, \gamma\}$ and

$V = \{s, t\}$. The max flow game is (A, v), but again it is separable into 3 single player games. Thus, $\varphi = (a, b, c)$ and $\nu = (a, b, c)$ in this game.

The max flow problem defined by Figure 6 has $A = \{\alpha, \beta, \gamma\}$ and $V = \{s, r, t\}$. The max flow game is (A, v), but again it is separable into the two-player game with α and β (like in Figure 1) and the single player game with γ. Thus $\varphi = (\frac{a \wedge b}{2}, \frac{a \wedge b}{2}, c)$ and $\nu = (\frac{a \wedge b}{2}, \frac{a \wedge b}{2}, c)$ in this game.

The max flow problem defined by Figure 4 allows for more interesting coalitional formations. Now, $A = \{\alpha, \beta, \gamma\}$ and $V = \{s, r, t\}$, and the capacities shall be denoted the same as before. This max flow game of (A, v) allows for coalitional formations for which we can look at four different cases in considering both the Shapley value and the nucleolus. The worths of the coalitions are as follows: $v(\alpha\gamma) = a \wedge c$, $v(\beta\gamma) = b \wedge c$, $v(\alpha\beta\gamma) = (a + b) \wedge c$, and $v(S) = 0$ for all other coalitions. Using the general formula of finding the Shapley value, we arrive at the following Shapley values for each arc:

$$\begin{aligned}
\varphi_\alpha &= \frac{1}{6}(a \wedge c) + \frac{1}{3}[(a + b) \wedge c - (b \wedge c)] \\
\varphi_\beta &= \frac{1}{6}(b \wedge c) + \frac{1}{3}[(a + b) \wedge c - (a \wedge c)] \\
\varphi_\gamma &= \frac{1}{6}(a \wedge c) + \frac{1}{6}(b \wedge c) + \frac{1}{3}[(a + b) \wedge c]
\end{aligned}$$

Using these, we can look at the four possible cases. We can arbitrarily assume that $b \geq a$ without loss of generality.

Case 1: $c \geq a + b$

$$\begin{aligned}\varphi_\alpha &= \frac{1}{6}(a \wedge c) + \frac{1}{3}[(a+b) \wedge c - (b \wedge c)] \\ &= \frac{1}{6}a + \frac{1}{3}[(a+b) - b] \\ &= \frac{1}{2}a\end{aligned}$$

$$\begin{aligned}\varphi_\beta &= \frac{1}{6}(b \wedge c) + \frac{1}{3}[(a+b) \wedge c - (a \wedge c)] \\ &= \frac{1}{6}b + \frac{1}{3}[(a+b) - a] \\ &= \frac{1}{2}b\end{aligned}$$

$$\begin{aligned}\varphi_\gamma &= \frac{1}{6}(a \wedge c) + \frac{1}{6}(b \wedge c) + \frac{1}{3}[(a+b) \wedge c] \\ &= \frac{1}{6}a + \frac{1}{6}b + \frac{1}{3}a + \frac{1}{3}b \\ &= \frac{1}{2}a + \frac{1}{2}b\end{aligned}$$

Case 2: $a + b > c \geq b \geq a$

$$\begin{aligned}\varphi_\alpha &= \frac{1}{3}c + \frac{1}{6}a - \frac{1}{3}b \\ \varphi_\beta &= \frac{1}{3}c + \frac{1}{6}b - \frac{1}{3}a \\ \varphi_\gamma &= \frac{1}{3}c + \frac{1}{6}a + \frac{1}{6}b\end{aligned}$$

Case 3: $b \geq c \geq a$

$$\varphi_\alpha = \frac{1}{6}a$$
$$\varphi_\beta = \frac{1}{2}c - \frac{1}{3}a$$
$$\varphi_\gamma = \frac{1}{2}c + \frac{1}{6}a$$

Case 4: $b \geq a \geq c$

$$\varphi_\alpha = \frac{1}{6}c$$
$$\varphi_\beta = \frac{1}{6}c$$
$$\varphi_\gamma = \frac{2}{3}c$$

The nucleolus can be looked at with these four cases as well. However, finding a generalized version of the nucleolus is a bit more complicated, but still possible. The nucleolus will be $\nu = (x_\alpha, x_\beta, x_\gamma)$. Further, $x_\alpha + x_\beta + x_\gamma = (a + b) \wedge c$ since this is $v(\alpha\beta\gamma)$. The coalitional complaints are as follows:

$$e_{\alpha\beta}(\nu) = -x_\alpha - x_\beta$$

$$e_{\alpha\gamma}(\nu) = (a \wedge c) - x_\alpha - x_\gamma$$

$$e_{\beta\gamma}(\nu) = (b \wedge c) - x_\beta - x_\gamma$$

$$e_\alpha(\nu) = -x_\alpha$$

$$e_\beta(\nu) = -x_\beta$$

$$e_\gamma(\nu) = -x_\gamma$$

This being the nucleolus, the maximum coalitional complaints should be minimized, and we can work from the idea that certain complaints will be equal. Working in this manner, we can arrive at a possible nucleolus and then check to ensure that we have indeed found it. We begin by setting $e_{\beta\gamma}(\nu) = e_\alpha(\nu)$. That is, $(b \wedge c) - x_\beta - x_\gamma = -x_\alpha$. Solving our equation involving the sum of the allocations, $x_\alpha + x_\beta + x_\gamma = (a+b) \wedge c$, for $-x_\beta - x_\gamma$ gives us $-x_\beta - x_\gamma = x_\alpha - (a+b) \wedge c$, which can be substituted into our complaint equation. Thus, $(b \wedge c) + x_\alpha - (a+b) \wedge c = -x_\alpha$. Solving for x_α, we get $x_\alpha = \frac{(a+b)\wedge c - (b \wedge c)}{2}$. Similarly, by setting $e_{\alpha\gamma} = e_\beta(\nu)$, and using a similar process, we get $x_\beta = \frac{(a+b)\wedge c - (a \wedge c)}{2}$. We can get x_γ by solving $x_\gamma = (a+b) \wedge c - x_\alpha - x_\beta$. This yields $x_\gamma = \frac{(b \wedge c) + (a \wedge c)}{2}$. In each of the four cases, this can be shown to be the nucleolus.

Case 1: $c \geq a + b$

$$\begin{aligned}
x_\alpha &= \frac{(a+b) \wedge c - (b \wedge c)}{2} \\
&= \frac{(a+b) - b}{2} \\
&= \frac{a}{2} \\
x_\beta &= \frac{(a+b) \wedge c - (a \wedge c)}{2} \\
&= \frac{(a+b) - a}{2} \\
&= \frac{b}{2} \\
x_\gamma &= \frac{(b \wedge c) + (a \wedge c)}{2} \\
&= \frac{b+a}{2}
\end{aligned}$$

We can show that this is indeed the nucleolus for this case by using Kohlberg's Theorem. First, finding the coalitional complaints in this case with the proposed nucleolus:

$$e_{\alpha\beta}(\nu) = \frac{-a-b}{2}$$
$$e_{\alpha\gamma}(\nu) = \frac{-b}{2}$$
$$e_{\beta\gamma}(\nu) = \frac{-a}{2}$$
$$e_{\alpha}(\nu) = \frac{-a}{2}$$
$$e_{\beta}(\nu) = \frac{-b}{2}$$
$$e_{\gamma}(\nu) = \frac{-a-b}{2}$$

Recalling that $b \geq a$, we have $e_{\alpha}(\nu) = e_{\beta\gamma}(\nu) \geq e_{\beta}(\nu) = e_{\alpha\gamma}(\nu) \geq e_{\gamma}(\nu) = e_{\alpha\beta}(\nu)$. Using Kohlberg's Theorem, we have $\mathcal{B}_1 = \{\{\alpha\},\{\beta,\gamma\}\}$, $\mathcal{B}_2 = \{\{\beta\},\{\alpha,\gamma\}\}$, and $\mathcal{B}_3 = \{\{\gamma\},\{\alpha,\beta\}\}$, since those are the sets of coalitions of highest excess, second highest excess, and third highest. Now, if we can show that each $\mathcal{D}_t = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_t$ is a balanced collection, then this is the prenucleolus. Further, since this is a superadditive game, Housman's Proposition proves that the prenucleolus is the same as the nucleolus. Now, $\mathcal{D}_1$ is balanced with $\lambda = (1,1)$, and $\mathcal{D}_2$ is balanced with $\lambda = (\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2})$, and $\mathcal{D}_3$ is balanced with $\lambda = (\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})$. Thus, this is indeed the nucleolus for this case. It works out that a similar argument can be made in each of the other cases to show that it is the nucleolus in each of those cases.

Case 2: $(a+b) > c \geq b \geq a$

$$x_\alpha = \frac{c-b}{2}$$
$$x_\beta = \frac{c-a}{2}$$
$$x_\gamma = \frac{a+b}{2}$$

Case 3: $b \geq c \geq a$

$$x_\alpha = 0$$
$$x_\beta = \frac{c-a}{2}$$
$$x_\gamma = \frac{c+a}{2}$$

Case 4: $b \geq a \geq c$

$$x_\alpha = 0$$
$$x_\beta = 0$$
$$x_\gamma = c$$

We can show that the general nucleolus we arrived at is indeed the nucleolus for all cases (similar to how we showed that it is the nucleolus in the first case). First here are two lemmas that will help in our proof.

Lemma 1. *With our proposed nucleolus, $\nu = (\frac{(a+b)\wedge c-(b\wedge c)}{2}, \frac{(a+b)\wedge c-(a\wedge c)}{2}, \frac{(b\wedge c)+(a\wedge c)}{2})$ it can be shown that $-x_\alpha \geq -x_\beta$.*

Proof. Now, $-x_\alpha = \frac{-(a+b)\wedge c+(b\wedge c)}{2}$ and $-x_\beta = \frac{-(a+b)\wedge c+(a\wedge c)}{2}$. To show that $-x_\alpha \geq -x_\beta$, we must show $-(a+b)\wedge c+(b\wedge c) \geq -(a+b)\wedge c+(a\wedge c)$. That is we want $b\wedge c \geq a\wedge c$. Looking at each case (and recalling that $b \geq a$) we get:

Case 1: $c \geq a+b$ Then, $b\wedge c = b$ and $a\wedge c = a$. Since, $b \geq a$, $b\wedge c \geq a\wedge c$.

Case 2: $(a+b) > c \geq b \geq a$ Again, $b\wedge c = b$ and $a\wedge c = a$. Since, $b \geq a$, $b\wedge c \geq a\wedge c$.

Case 3: $b \geq c \geq a$ Then, $b\wedge c = c$ and $a\wedge c = a$. By the case assumption that $c \geq a$, $b\wedge c \geq a\wedge c$.

Case 4: $b \geq a \geq c$ Then, $b\wedge c = c$ and $a\wedge c = c$. Since these are equal, then again we can say $b\wedge c \geq a\wedge c$.

Thus, in each of the four cases, we have shown that $b\wedge c \geq a\wedge c$ which was sufficient to show that $-x_\alpha \geq -x_\beta$. □

Lemma 2. *With our proposed nucleolus, $\nu = (\frac{(a+b)\wedge c-(b\wedge c)}{2}, \frac{(a+b)\wedge c-(a\wedge c)}{2}, \frac{(b\wedge c)+(a\wedge c)}{2})$ it can be shown that $-x_\gamma \leq -x_\alpha - x_\beta$.*

Proof. Now, $-x_\gamma = \frac{-(b\wedge c)-(a\wedge c)}{2}$ and $-x_\alpha - x_\beta = \frac{-(a+b)\wedge c+(b\wedge c)}{2} + \frac{-(a+b)\wedge c+(a\wedge c)}{2}$. To show that $-x_\gamma \leq -x_\alpha - x_\beta$, we must show $-(b\wedge c)-(a\wedge c) \leq -(a+b)\wedge c+(b\wedge c)-(a+b)\wedge c+(a\wedge c)$. Simplifying a bit, we find that we need to show $(a+b)\wedge c \leq b\wedge c + a\wedge c$ in order to prove $-x_\gamma \leq -x_\alpha - x_\beta$. Looking at each case (and recalling that $b \geq a$) we get:

Case 1: $c \geq a+b$ Then, $(a+b)\wedge c = (a+b)$ and $b\wedge c + a\wedge c = b+a$. Since these are equal, we can definitely say $(a+b)\wedge c \leq b\wedge c + a\wedge c$.

Case 2: $(a+b) > c \geq b \geq a$ Then, $(a+b) \wedge c = c$ and $b \wedge c + a \wedge c = b + a$. Since in this case, we assumed $(a+b) > c$, we have $(a+b) \wedge c < b \wedge c + a \wedge c$.

Case 3: $b \geq c \geq a$ Then, $(a+b) \wedge c = c$ and $b \wedge c + a \wedge c = c + a$. Since $c \leq c + a$, we can again say $(a+b) \wedge c \leq b \wedge c + a \wedge c$.

Case 4: $b \geq a \geq c$ Then, $(a+b) \wedge c = c$ and $b \wedge c + a \wedge c = 2c$. Since $c \leq 2c$, we have $(a+b) \wedge c \leq b \wedge c + a \wedge c$.

Thus, in each of the four cases, we have shown that $(a+b) \wedge c \leq b \wedge c + a \wedge c$ which was sufficient to show that $-x_\gamma \leq -x_\alpha - x_\beta$. □

Now we can use these lemmas to prove that the proposed nucleolus is the actual nucleolus in all four cases.

Theorem 4. *The proposed nucleolus* $\nu = \left(\frac{(a+b)\wedge c-(b\wedge c)}{2}, \frac{(a+b)\wedge c-(a\wedge c)}{2}, \frac{(b\wedge c)+(a\wedge c)}{2}\right)$ *is the actual nucleolus for the max flow game from the max flow problem in Figure 4.*

Proof. Using the proposed nucleolus, the excesses of the coalitions in this max flow game were defined as follows:

$$e_{\alpha\beta}(\nu) = -x_\alpha - x_\beta$$

$$e_{\alpha\gamma}(\nu) = (a \wedge c) - x_\alpha - x_\gamma$$

$$e_{\beta\gamma}(\nu) = (b \wedge c) - x_\beta - x_\gamma$$

$$e_\alpha(\nu) = -x_\alpha$$

$$e_\beta(\nu) = -x_\beta$$

$$e_\gamma(\nu) = -x_\gamma$$

Our proposed nucleolus $(\frac{(a+b)\wedge c-(b\wedge c)}{2}, \frac{(a+b)\wedge c-(a\wedge c)}{2}, \frac{(b\wedge c)+(a\wedge c)}{2})$ was set up so that $e_\alpha(\nu) = e_{\beta\gamma}(\nu) = -x_\alpha$ and $e_\beta(\nu) = e_{\alpha\gamma}(\nu) = -x_\beta$. From Lemma 1, we know that $-x_\alpha \geq -x_\beta$, so $e_\alpha(\nu) = e_{\beta\gamma}(\nu) \geq e_\beta(\nu) = e_{\alpha\gamma}(\nu)$.
Further, $e_{\alpha\beta}(\nu) = -x_\alpha - x_\beta$. Now, $e_\alpha(\nu) = e_{\beta\gamma}(\nu) \geq e_{\alpha\beta}(\nu)$ since $-x_\alpha \geq -x_\alpha - x_\beta$. Also, $e_\beta(\nu) = e_{\alpha\gamma}(\nu) \geq e_{\alpha\beta}(\nu)$ since $-x_\beta \geq -x_\alpha - x_\beta$.
From Lemma 2, we know that $-x_\gamma \leq -x_\alpha - x_\beta$, so consequently $e_\gamma(\nu) \leq e_{\alpha\beta}(\nu)$.

Thus, we have set up orders of excesses and we can look at the collections involved to determine if they are balanced. Using Kohlberg's Theorem, we have $\mathcal{B}_1 = \{\{\alpha\}, \{\beta, \gamma\}\}$, $\mathcal{B}_2 = \{\{\beta\}, \{\alpha, \gamma\}\}$, $\mathcal{B}_3 = \{\{\alpha, \beta\}\}$, and $\mathcal{B}_4 = \{\{\gamma\}\}$ since those are the sets of coalitions of highest excess, second highest excess, third highest and fourth highest. Now, if we can show that each $\mathcal{D}_t = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \cdots \cup \mathcal{B}_t$ is a balanced collection, then ν is the prenucleolus. Further, since this is a superadditive game, Housman's

Proposition proves that the prenucleolus is the same as the nucleolus. Now, $\mathcal{D}_1$ is balanced by $\lambda = (1,1)$, $\mathcal{D}_2$ is balanced by $\lambda = (\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2})$, $\mathcal{D}_3$ is balanced by $\lambda = (\frac{1}{4},\frac{1}{2},\frac{1}{4},\frac{1}{2},\frac{1}{4})$, and $\mathcal{D}_4$ is balanced by $\lambda = (\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3},\frac{1}{3})$.

Thus, we have shown that our proposed nucleolus, ν, is the actual nucleolus for all cases of the defined game. □

3.4 Four Player Games

There are several games with four arcs, but many of these are separable and thus the Shapley value and the nucleolus for those games are easily obtained from our previous work and Theorem 2. Thus, we will only be looking at the five basic four-player games which are not separable into smaller games. They are as follows:

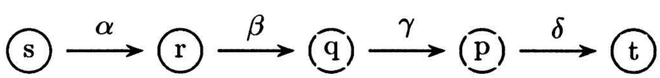


Figure 7: The first max flow problem with four arcs

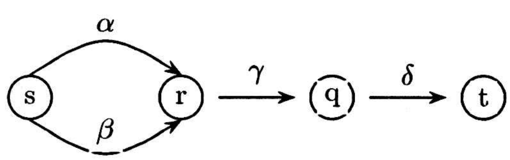


Figure 8: The second max flow problem with four arcs

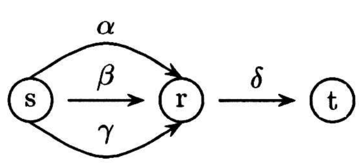


Figure 9: The third max flow problem with four arcs

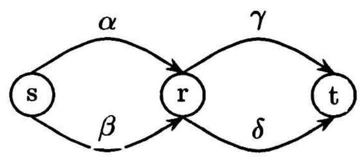


Figure 10: The fourth max flow problem with four arcs

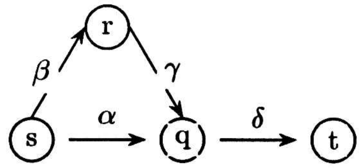


Figure 11: The fifth max flow problem with four arcs

The max flow problem defined by Figure 7 has $A = \{\alpha, \beta, \gamma, \delta\}$ and $V = \{s, r, q, p, t\}$. The capacities again shall be denoted as follows: $c(\alpha) = a$, $c(\beta) = b$, $c(\gamma) = c$, and $c(\delta) = d$. The max flow game is (A, v) with $v(\alpha\beta\gamma\delta) = a \wedge b \wedge c \wedge d$ and $v(S) = 0$ for all $S \subset A$. Again this leaves us with $\varphi = \nu = (\frac{a\wedge b\wedge c\wedge d}{4}, \frac{a\wedge b\wedge c\wedge d}{4}, \frac{a\wedge b\wedge c\wedge d}{4}, \frac{a\wedge b\wedge c\wedge d}{4})$.

The other four player games get much more complicated in terms of looking at all the possible cases. We will just look at the general Shapley values for each of the four player coalitions.

The max flow problem defined by Figure 8 has $A = \{\alpha, \beta, \gamma, \delta\}$ and $V = \{s, r, q, t\}$. The capacities again shall be denoted as follows: $c(\alpha) = a$, $c(\beta) = b$, $c(\gamma) = c$, and $c(\delta) = d$. The max flow game is (A, v) with the worth of the coalitions as follows: $v(\alpha\gamma\delta) = a \wedge c \wedge d$, $v(\beta\gamma\delta) = b \wedge c \wedge d$, $v(\alpha\beta\gamma\delta) = (a + b) \wedge c \wedge d$, and $v(S) = 0$ for all other coalitions. The general form for finding the Shapley value for each of the arcs is:

$$\varphi_\alpha = \frac{1}{12}(a \wedge c \wedge d) + \frac{1}{4}((a+b) \wedge c \wedge d - b \wedge c \wedge d)$$
$$\varphi_\beta = \frac{1}{12}(b \wedge c \wedge d) + \frac{1}{4}((a+b) \wedge c \wedge d - a \wedge c \wedge d)$$
$$\varphi_\gamma = \frac{1}{12}(a \wedge c \wedge d) + \frac{1}{12}(b \wedge c \wedge d) + \frac{1}{4}((a+b) \wedge c \wedge d)$$
$$\varphi_\delta = \frac{1}{12}(a \wedge c \wedge d) + \frac{1}{12}(b \wedge c \wedge d) + \frac{1}{4}((a+b) \wedge c \wedge d)$$

The max flow problem defined by Figure 9 has $A = \{\alpha, \beta, \gamma, \delta\}$ and $V = \{s, r, t\}$ with the capacities again designated $c(\alpha) = a$, $c(\beta) = b$, $c(\gamma) = c$, and $c(\delta) = d$. The worth of the coalitions is as follows: $v(\alpha\delta) = a \wedge d$, $v(\beta\delta) = b \wedge d$, $v(\gamma\delta) = c \wedge d$, $v(\alpha\beta\delta) = (a+b) \wedge d$, $v(\alpha\gamma\delta) = (a+c) \wedge d$, $v(\beta\gamma\delta) = (b+c) \wedge d$, and $v(\alpha\beta\gamma\delta) = (a+b+c) \wedge d$, with $v(S) = 0$ for all other coalitions. The general form for finding the Shapley value for each of the arcs is:

$$
\begin{aligned}
\varphi_\alpha &= \frac{1}{12}(a \wedge d) + \frac{1}{12}((a+b) \wedge d - (b \wedge d)) + \frac{1}{12}((a+c) \wedge d - (c \wedge d)) \\
&+ \frac{1}{4}((a+b+c) \wedge d - ((b+c) \wedge d)) \\
\varphi_\beta &= \frac{1}{12}(b \wedge d) + \frac{1}{12}((a+b) \wedge d - (a \wedge d)) + \frac{1}{12}((b+c) \wedge d - (c \wedge d)) \\
&+ \frac{1}{4}((a+b+c) \wedge d - ((a+c) \wedge d)) \\
\varphi_\gamma &= \frac{1}{12}(c \wedge d) + \frac{1}{12}((a+c) \wedge d - (a \wedge d)) + \frac{1}{12}((b+c) \wedge d - (b \wedge d)) \\
&+ \frac{1}{4}((a+b+c) \wedge d - ((a+b) \wedge d)) \\
\varphi_\delta &= \frac{1}{12}(a \wedge d) + \frac{1}{12}(b \wedge d) + \frac{1}{12}(c \wedge d) + \frac{1}{12}((a+b) \wedge d) \\
&+ \frac{1}{12}((a+c) \wedge d) + \frac{1}{12}((b+c) \wedge d) + \frac{1}{4}((a+b+c) \wedge d)
\end{aligned}
$$

The max flow problem defined by Figure 10 has $A = \{\alpha, \beta, \gamma, \delta\}$ and $V = \{s, r, t\}$ with the capacities again designated $c(\alpha) = a$, $c(\beta) = b$, $c(\gamma) = c$, and $c(\delta) = d$. The worth of the coalitions is as follows: $v(\alpha\gamma) = a \wedge c$, $v(\alpha\delta) = a \wedge d$, $v(\beta\delta) = b \wedge d$, $v(\beta\gamma) = b \wedge c$, $v(\alpha\beta\gamma) = (a+b) \wedge c$, $v(\alpha\beta\delta) = (a+b) \wedge d$, $v(\alpha\gamma\delta) = a \wedge (c+d)$, $v(\beta\gamma\delta) = b \wedge (c+d)$, and $v(\alpha\beta\gamma\delta) = (a+b) \wedge (c+d)$, with $v(S) = 0$ for all other coalitions. The general form for finding the Shapley value for each of the arcs is:

$$
\begin{aligned}
\varphi_\alpha &= \frac{1}{12}(a \wedge c) + \frac{1}{12}(a \wedge d) + \frac{1}{12}((a+b) \wedge c - (b \wedge c)) + \frac{1}{12}((a+b) \wedge d - (b \wedge d)) \\
&+ \frac{1}{12}(a \wedge (c+d)) + \frac{1}{4}((a+b) \wedge (c+d) - (b \wedge (c+d))) \\
\varphi_\beta &= \frac{1}{12}(b \wedge c) + \frac{1}{12}(b \wedge d) + \frac{1}{12}((a+b) \wedge c - (a \wedge c)) + \frac{1}{12}((a+b) \wedge d - (a \wedge d)) \\
&+ \frac{1}{12}(b \wedge (c+d)) + \frac{1}{4}((a+b) \wedge (c+d) - (a \wedge (c+d))) \\
\varphi_\gamma &= \frac{1}{12}(a \wedge c) + \frac{1}{12}(b \wedge c) + \frac{1}{12}((a+b) \wedge c) + \frac{1}{12}(a \wedge (c+d) - (a \wedge d)) \\
&+ \frac{1}{12}(b \wedge (c+d) - (b \wedge d)) + \frac{1}{4}((a+b) \wedge (c+d) - ((a+b) \wedge d)) \\
\varphi_\delta &= \frac{1}{12}(a \wedge d) + \frac{1}{12}(b \wedge d) + \frac{1}{12}((a+b) \wedge d) + \frac{1}{12}(a \wedge (c+d) - (a \wedge c)) \\
&+ \frac{1}{12}(b \wedge (c+d) - (b \wedge c)) + \frac{1}{4}((a+b) \wedge (c+d) - ((a+b) \wedge c))
\end{aligned}
$$

The max flow problem defined by Figure 11 has $A = \{\alpha, \beta, \gamma, \delta\}$ and $V = \{s, r, q, t\}$ with the capacities again designated $c(\alpha) = a$, $c(\beta) = b$, $c(\gamma) = c$, and $c(\delta) = d$. The worth of the coalitions is as follows: $v(\alpha\delta) = v(\alpha\beta\delta) = v(\alpha\gamma\delta) = (a \wedge d)$, $v(\beta\gamma\delta) = b \wedge c \wedge d$, $v(\alpha\beta\gamma\delta) = (a + (b \wedge c)) \wedge d$, and $v(S) = 0$ for all other coalitions. The general form for finding the Shapley value for each of the arcs is:

$$\varphi_\alpha = \frac{1}{4}(a \wedge d) + \frac{1}{4}((a + (b \wedge c)) \wedge d - (b \wedge c \wedge d))$$
$$\varphi_\beta = \frac{1}{12}(b \wedge c \wedge d) + \frac{1}{4}((a + (b \wedge c)) \wedge d - (a \wedge d))$$
$$\varphi_\gamma = \frac{1}{12}(b \wedge c \wedge d) + \frac{1}{4}((a + (b \wedge c)) \wedge d - (a \wedge d))$$
$$\varphi_\delta = \frac{1}{4}(a \wedge d) + \frac{1}{12}(b \wedge c \wedge d) + \frac{1}{4}((a + (b \wedge c)) \wedge d)$$

References

[1] Hillier, Frederick and Gerald Lieberman, *Introduction to Operations Research, Fourth Ed.*, Holden-Days Inc,1986.

[2] Housman, David, *The Nucleolus and Balanced Collections*, June, 1989.

[3] Jianhua, Wang, *The Theory of Games*, Oxford University Press, 1988.

[4] Kohlberg, Elon, *On the Nucleolus of a Characteristic Function Game*, SIAM J. Appl. Math, January, 1971: pp. 62-67.

[5] Shapley, L.S., *Contributions to the Theory of Games*, Annals of Mathematical Studies: pp. 307-17.

[6] Straffin, Philip D., *Game Theory and Strategy*, The Mathematical Association of America, 1993.