GENERALIZED NETWORK PROBLEMS YIELDING TOTALLY BALANCED CASES

by

Aimil Kadai and Elisar Zemel

June 1980

Northwestern University

Rough Draft February, 1980
Revised May, 1980

This research was supported in part by the J. L. Kellogg Center for Advanced Studies in Managerial Economics. In addition Kadai's work was supported by National Science Foundation Grant #NSC-7807542 and Zemel's work was supported by National Science Foundation Grant # ENG-7907556.
ABSTRACT

A class of multi-person mathematical optimization problems is considered and is shown to generate cooperative games with nonempty cores. The class includes, but is not restricted to, numerous versions of network flow problems. It is shown that for games generated by Linear Programming optimization problems, optimal dual solutions correspond to points in the core. Also a special class of network flow problems for which every point in the core corresponds to an optimal dual solution is exhibited.
INTRODUCTION

Network flow models have been applied extensively for analyzing systems in the areas of communication, transportation, distribution, integrated production and the like. In systems of this type one can usually identify certain objects, not necessarily physical, which could be described as "flowing" through the system. For instance, one may consider a fluid flowing in a pipeline network, traffic moving or goods being transformed through a transportation network, currencies being exchanged through a network of exchange dealers, telephone calls transmitted through a telephone network, physical "inputs" being transformed through a production system into "finished goods" etc. Typically, the nodes of the network correspond to the different possible "states" in which the "flow" can be found while the arcs correspond to the active elements in the system which can transform the "flow" from "state" to "state".

Given a system of this type, one is naturally interested in employing it in the most efficient way possible. For instance, given a transportation system, one likes to find a transportation pattern which maximizes the flow of traffic between two specified terminals or, if costs are present, to find a transportation pattern which supports a given intensity of flow at the lowest possible cost. Similarly, in production problems with many alternative paths of production, one would like to find a production schedule which maximizes total output (or profits)
without exceeding production capacities and without disturbing the order of operations in which production should take place.

This sentiment is aptly reflected in the U.S. literature concerning network flow problems. Originating with the pioneering work of Ford and Fulkerson, [8], a vast number of algorithms for finding optimal flows have been proposed in the last 25 years. To date, we have available a wide choice of extremely fast algorithms for two important versions of such models namely the maximal flow problem and the minimum cost flow problem, [1,5,7,8,10,12,15,21]. Practical applications of these models abound and problems involving many thousands of variables and constraints are being solved routinely.

A closer examination of the above mentioned literature reveals, however, one common underlying feature. Almost without exception, these models are predicated on the assumption that the network is fully controlled by one individual or by a group of individuals with identical interests. When we analyze networks in which various components are controlled by different individuals, with different, and possibly conflicting, objectives, we soon realize that the optimization problem becomes a multiperson optimization problem and game theoretic considerations arise.

Consider, for example, a maximal flow problem in a network in which arcs are owned by different individuals. We can easily find the optimal flow in this network by the classical methods. However, to sustain this optimal flow we must secure the cooperation of some critical arc owners. Their cooperation can
be secured provided that they get paid "enough". However the concept of "enough" requires careful consideration. A natural criterion suggested by the game theory literature is that the payoff will be such that no group of owners can generate a higher payoff for themselves when acting on their own by using only their portion of the network. In game theoretic language we require the payoff distribution to be in the core of the resulting cooperative game. This seems like a natural necessary condition on the payoff distribution because a coalition of arc owners that can generate more profits for themselves than was allocated to them by the grand coalition will tend to break cooperation and act on their own. However, the existence of such a payoff distribution is not an obvious fact. One can easily generate examples of games for which the requirements of the various coalitions are inconsistent resulting in an empty core.

The most elementary problem to analyze is that of a maximum flow problem of one commodity through a network possessing a single source and a single sink. Every arc has a flow capacity constraint and is owned by some player. A unit flow from source to sink yields a unit profit. It was shown in Kalai-Samet [14] that for such problems the resulting core is not empty (for example payoff distributions that correspond to minimum cuts in the network are always core allocations).

In this paper we study broad generalizations of the maximal flow problem which possess this property. The resulting class of problems contains various network flow problems involving, for instance, costs (profits) on the individual arcs, multiple
sources and sinks, multi-commodity flows, networks with losses and gains, networks with production nodes, etc. In addition, various types of other optimization problems which are not related to networks are also covered. A similar approach of this type was taken earlier in Owen [16], Bernsow [13] and Millers [3]. However the family of optimization problems and the resulting games are different there. Also, simultaneously and independently of our work, Dube and Shapley [7a] have developed a different set of sufficient conditions for optimization problems to guarantee that they generate games with non-empty cores. Many of the examples of games with this property can be shown to fit into both models (the Dube-Shapley model as well as ours). Thus it seems that both models are very general.

The organization of the paper is as follows. In section II we introduce some concepts from Game Theory, and examine the notion of totally balanced games. In section III we discuss a class of optimization problems which yield games having this property. In section IV we discuss an economic example - The Shapley - Shubik Market Games. In section V we discuss games arising from certain types of linear programming problems, which includes a large variety of network flow problems as well as other types of problems. In section VI we discuss a certain class of network flow problems for which the core of the resulting games can be fully described in terms of optimal dual prices. We conclude the paper with a few examples.
II. Concepts from Game Theory

Let \( N = \{1, \ldots, n\} \) (\( n > 0 \)) be a set of players. A coalition consists of any non-empty subset of \( N \). We denote the set of all coalitions of \( N \) by \( C \). An \( n \)-person cooperative game with sidepayments (a game for short) is a function \( V \) from the set of coalitions to the set of real numbers. For a coalition \( S \in C \), \( V(S) \) may be thought of as the value (monetary or otherwise) that the coalition \( S \) can generate for its members if it operates on its own. For example, in the maximal flow problem described earlier, \( V(S) \) may stand for the maximum flow that \( S \) can sustain using only its portion of the network.

For a game \( V \), the core of \( V \) is defined by

\[
\text{CORE}(V) = \{ x \in \mathbb{R}^n : \sum x_i = V(N), \sum x_i > V(S) \text{ for every } S \subset N \}
\]

A point in the core corresponds to a distribution of the total profits (or costs) to the different players of the game. The constraints imposed on \( \text{CORE}(V) \) ensure that no coalition would have an incentive to split from the grand coalition, \( N \), and do better on its own. Thus, for an allocation of \( V(N) \) to belong to \( \text{CORE}(V) \) it is a necessary condition for long-term cooperation between the players. Attempting to implement profit distributions which violate some of the core constraints is likely, in the long run, to result in some coalition breaking cooperation.

In general, a game \( V \) may or may not have a non-empty core. In light of the discussion of the preceding paragraph, it is
apparent that establishing whether or not \( \text{CORE} (V) = \emptyset \) is of considerable practical relevance. Moreover, given that \( \text{CORE} (V) \neq \emptyset \) we would obviously like to find one or several points which belong to this set. Both these issues can be settled by solving the following, typically huge, linear program:

\[
\begin{align*}
\text{(1)} & \quad \min \sum_{j \in \mathcal{N}} x_j \\
\text{s.t.} & \quad \sum_{j \in S} x_j > v(S) \text{ for every } S \subseteq \mathcal{N}
\end{align*}
\]

It is quite obvious that \( \text{CORE} (V) \neq \emptyset \) if and only if the optimal solution of (1) and (2) is equal to \( v(N) \), in which case any optimal solution to this program lies in \( \text{CORE} (V) \).

Alternatively, taking the linear programming dual of (1) – (2) we get an equivalent necessary and sufficient condition for \( \text{CORE} (V) \neq \emptyset \) which is based on the concept of balanced sets. If \( \mathcal{B} = \{b_1, \ldots, b_N\} \) is a collection of coalitions and \( \lambda = (\lambda_1, \ldots, \lambda_N) \) is a set of non-negative real numbers such that for every \( i \in \mathcal{N} \) \( \sum_{j \in \mathcal{N}} \lambda_j = 1 \), then \( v \) is called a \textit{balanced collection}. The set of coefficients \( \lambda \) is referred to in this case as \textit{balancing weights}. We call a game \( v \) \textit{balanced} if, for every balanced collection \( \mathcal{B} \) and every corresponding set of balancing weights \( \lambda \) we have

\[
\sum_{S \subseteq \mathcal{N}} \lambda_S v(S) < v(N).
\]
A game has a non empty core if and only if it is balanced.

Let $V$ be a game with $n$ players, and $S \in C$ a coalition. We denote by $V^S$ the $|S|$ players game obtained by restricting $V$ to coalitions $I \subseteq S$. ($V^S$ is the restriction of the function $V$ to $2^S$). We will be concerned in this paper with the family of highly stable games for which

$$\text{Core}(V^S) \neq \emptyset \text{ for every } S \in C.$$ 

Applying Theorem 1 we get that a game belong to this family, iff each of its subgames, $V^S$, $S \in C$ is balanced. Games with this property are called totally balanced games.

III. Optimization Problems Resulting in Totally Balanced Games

In this section we describe a general class of mathematical optimization problems with the property that the cooperative games induced by them are totally balanced.

Verbally, the games may be described as follows. Every player controls a set of variables. Every coalition has a feasible set constraining the variables of its members. Denote this feasible set by $V^S$. There is a common objective function which is used by all coalitions. The value of a coalition $S$, $V(S)$, will be the maximum of the objective function over the
feasible set of $S$, $Y^S$.

Formally, for every player $i \in N$ we let $d^i$ be a positive integer denoting the number of variables under $i$'th control. Let $d = \sum_{i=1}^{n} d^i$. Also, let $D_i = \{j \mid 1 \leq j \leq d_i \}$ be the set of variables under $i$'s control. For every coalition $S$ let $X^S = \{y \in \mathbb{R}^d : y^j = 0, \text{ for } j \in D_i \}$ be the subspace that assigns value zero to variables which are not in $S$'s control. For a point $y \in \mathbb{R}^d$ let $y^S$ be its orthogonal projection on $X^S$. Let $Y = \{Y^S\}_{S \subseteq C}$ be a collection of sets in $\mathbb{R}^d$ with the property that $y^S \in X^S$. We call the collection $Y$, balanced (see Billera and Sturmfels [2] for definitions and references) if for every balanced collection of coalitions $S = (S_1, \ldots, S_k)$ with balancing set of weights $\lambda = (\lambda_1, \ldots, \lambda_k)$ and for every collection of feasible points $(y_1, \ldots, y_k)$ with $y^j \in Y^S$, one has $\sum_{i=1}^{k} \lambda_i^j y_i^j \in Y^n$. The collection $Y$ is called totally balanced if the same property holds for the restriction of $Y$ to the subsets of every coalition $S \subseteq C$.

Having defined the feasible set for an optimization problem we now define the objective function. Let $f$ be a real valued function on $\mathbb{R}^d$. $f$ is called super balanced if for every balanced collection $S = (S_1, \ldots, S_k)$ with balancing set of weights $\lambda = (\lambda_1, \ldots, \lambda_k)$ and for every collection of feasible points $(y_1, \ldots, y_k)$ with $y^j \in Y^S$, one has $\sum_{i=1}^{k} \lambda_i^j f(y_i^j) > \sum_{i=1}^{k} \lambda_i^j f(y^j)$. $f$ is called totally super balanced if the same property holds for every restriction of the set of players. Finally, $f$ is called (totally) bounded on $Y$ if $\sup_y f(y)$ (sup$_{Y^S} f(y)$) for every $Y \subseteq C$ is finite. yet yet Defining the game associated with pair $(Y,f)$ by
\[ V(S) = \sup_{y \in S} f(y) \]

**Lemma 1.** If \( V \) is totally balanced and \( f \) is totally bounded and totally super placed on \( Y \) then the game induced by \((f,f)\) is totally balanced.

**Proof.** We show that the induced game, \( V \), is balanced. The totally balanced part is proved in the same way. Let \( S = (S_1, S_2, \ldots, S_k) \) be a balanced collection of coalitions with balancing weights \((\lambda_1, \lambda_2, \ldots, \lambda_k)\). Let \( e > 0 \) be arbitrary. We will show that \( \sum_{i=1}^{k} \lambda_i V(S_i) - k e < V(S) \). For every \( i \), \( 1 \leq i \leq k \), there is a \( y_1 \in \mathbb{R}^{S_i} \) with \( f(y_1) > V(S_i) - e \). Consider \( y = \sum_{i=1}^{k} \lambda_i y_i \in \mathbb{R}^S \). \( f(y) > \sum_{i=1}^{k} \lambda_i f(y_i) \). Therefore \( V(S) > f(y) > \sum_{i=1}^{k} \lambda_i V(S_i) - e \). Hence \( \sum_{i=1}^{k} \lambda_i V(S_i) - k e < V(S) \).

**Remark 1.** The reader who is familiar with the theory of cooperative games without side payments will notice that this lemma can be easily formulated for such games. This can be done by considering \( n \) objective functions \( u_1, u_2, \ldots, u_n \) representing the utilities of the \( n \) players with properties similar to \( f \) above to induce totally balanced games without side payment (see Sillera-Bixby [2]).

**Remark 2.** The converse of Lemma 1 is also true. This was first established by Shapley-Shubik [19] (see Section 15 below). It also follows from the results of Kalai-Zemel [14].
IV. An Economic Example: Shapley – Shubik Market Game

The following games where discussed in Shapley – Shubik [19]. Using the terminology of Lemma I, they can be described as follows.

We assume that all the $d_i$'s of the players are identical and equal $c$. e may be thought of as the number of commodities that are traded in some market. With each player $i \in N$ we associate a pair $(v_i, u_i)$. $v_i$ is a point in the non-negative orthant of $R^c$ and represents the initial amounts that player $i$ holds of the various commodities. $u_i: R^c \rightarrow R$ which is concave and continuous, may be thought of as the utility or the monetary gain that player $i$ can get out of a given bundle. For every $y \in R^c$ we define $f(y) = \sum_{i=1}^{n} u_i(y^i)$. In other words the value of an allocation $y$ is the sum of the values that the individual players have for it. For every coalition $s$ we define $y^s$ by

$$y^s = \{y(y_1, y_2, \ldots, y_n) \in R^c; \sum_{i \in s} y_i = \sum_{i \not\in s} y_i \quad \text{and} \quad y_i \geq 0 \quad \text{for} \quad i = 1, 2, \ldots, n\}.$$ 

It is easy to check that $(Y, f)$ satisfies the assumptions of Lemma I and thus the induced market game is totally balanced and has non empty cores for all its subgames. Shapley and Shubik, [19], have established also the converse of this statement i.e. that every totally balanced game can be generated by a pair $(Y, f)$ arising from a market game of this type.
IV. Linear Programming Games

We now turn our attention to a special class of optimization problems, which can be expressed as linear programs and which satisfy the conditions of lemma 1. This class includes, among other problems, all the generalizations of network flow problems referred to in the introduction. Thus, it will be shown that the games associated with such optimization problems are totally balanced. Moreover, the linear nature of the representation of such games leads to a natural and computationally feasible procedure to identify some points which are known to be in the core.

We define the optimization problem as follows. For each player \( i \in N \) let \( d_i \) denote the dimension of the space of variables controlled by this player. Let \( d = \sum_{i \in N} d_i \). Let also \( c^i \in \mathbb{R}^{d_i} \) be an arbitrary profit (or cost if the entries are negative) vector for player \( i \). Let \( A_i \) and \( b_i \) be two arbitrary matrices associated with this player of dimensions \( A_i \in \mathbb{R}^{n \times d_i} \) and \( b_i \in \mathbb{R}^{n} \) respectively, with \( \{ A_i, b_i \} \subseteq N \), and \( m \) arbitrary nonnegative integers. Finally, for each \( i \in N \), let \( b_i \in \mathbb{R}^{d_i} \) be an arbitrary vector of right hand sides. For every coalition \( S \subset N \) we let

\[
(1) \quad V(S) = \max_{x^i} \sum_{i \in S} c^i x^i
\]

s.t.

\[
(2) \quad A^i x^i \leq b^i \text{ for every } i \in S,
\]

\[
(3) \quad \sum_{i \in S} b^i x^i = 0
\]
For $S - N$ one can think of problem (1) - (3) as a problem of decentralization with the different players serving as subsidiaries or subcontractors. Such problems are amenable to solution by various decomposition techniques (e.g., Dantzig-Wolfe, [6]) which stress the flow of information and control between the main organization and the individual subunits, i.e., the players. We note that numerous versions of network flow problem conform to this format. In the simplest versions of such problems, the constants (4) correspond to the individual arc capacity constraints while (5) represents the requirement that flow be conserved at each intermediate node. Obviously, the formulation (3) - (5) allows for significant generalization of such constraints. In the rest of this section we restrict our attention to optimization problems which yield $V(S) < w$, for $S \in C$, and $V(N) > -w$. It is a simple matter to establish the following:

Theorem 1 The game $V$ derived from the optimization problem (3) - (5) is totally balanced.

Proof We show that the game satisfy the conditions of Lemma 1. The objective function is a sum of linear functions and therefore totally super balanced. The assumption on the boundedness of $V(S)$, $S \in C$ imply that $f$ is totally bounded as well. To show that the feasible regions are totally balanced, we note that showing balanceess is enough, since the problem restricted to subcoalitions of any coalition $S \in C$ has the same structure as
the original problem.

Let \((S_1, \ldots, S_k)\) and \((\lambda_1, \ldots, \lambda_k)\) be a balanced set of coalitions and the corresponding weights. For \(i = 1, \ldots, k\), consider the optimization problem defined with respect to \(S_i\). Let \(x_i^1\) be a feasible solution to this problem. We assume that the \(x_i^1\)'s are augmented by zeros so that they all belong to \(\mathbb{R}^d\). Consider the vector \(x = \sum_{i=1}^k \lambda_i x_i^1\). We have to show that \(x\) satisfies (4) - (3) with respect to \(S = N\). But this is manifest: each \(x_i^1\) satisfies constraints (5) by our assumption, and therefore so does \(x\) which is a linear combination of the \(x_i^1\). Also each \(x_i^1\) satisfies (4) for all \(j \in S_i\). Let \(x_i^j \in \mathbb{R}^d_{+}\) be the component of \(x_i^1\) which corresponds to player \(j\). By the assumption of balance, \(x_i^j\) is a convex combination of the set \(\{x_i^j : j \in S_i\}\); thus, \(x_i^j\) satisfies (4).

The proof of Theorem 1 is non-constructive in nature. It asserts that \(\text{CORE} (V) \neq \emptyset\) but does not indicate how a point in this set can be found. Theorem 2 below addresses itself to this issue. It establishes a connection between some points in \(\text{CORE} (V)\) and the optimal dual solutions to problem (3) - (5) defined with respect to \(V\).

**Theorem 2** Let \(u, v\) be an optimal dual solution to (3) - (5), defined w.r.t. \(V\) with \(u = (u^i : i \in S)\). Let \(x = (x_i^1 : i \in N)\) be given by \(x_i^1 = u^i y^i\). Then \(x \in \text{CORE} (V)\).

**Proof** We first note that

\[
\forall i \in S \quad \sum_{i=1}^k \lambda_i x_i^1 \sum_{t \in T} u^t b^{i, t} = u^i b = v(S)
\]
as it should. Consider a coalition $S \in C$. Let $u^S = (u^i; i \in S)$. It can be easily verified that $(u^S, w)$ is a feasible dual solution for $(3)-(5)$ defined w.r.t. $S$. Thus, by the weak duality theorem of linear programming

$$\sum_{i \in S} x^i = \sum_{i \in S} u^i - \sum_{i \in S} b^i > V(S).$$

Theorem 2 enables us to compute points in the core of the game $V$ without having to compute first the $2^n$ constants $V(S)$, $S \in C$. (The first example of such efficiency was demonstrated for assignment games by Shapley-Shubik [20].) In addition, the allocations suggested in Theorem 2 lend themselves to economic interpretation consistent with traditional LP interpretations of shadow prices.

Is the converse of theorem 2 true? The following counterexample settles this question in the negative. However, in section V, we identify a class of network games for which every core allocation corresponds to an optimal dual vector.

Example 1 Consider the network of figure 1 where each arc is labeled by its index, (a letter), and its capacity.

![Figure 1](image-url)
For each $S \not= \emptyset \{a, b, c\}$, let $\psi(S)$ be the maximal flow from $a$ to $t$ through the network containing the arcs of $S$ only.

Obviously

$$\psi(i) = 0 \text{ for } i = a, b, c,$$
$$\psi(a, b) = \psi(a, c) = 1,$$
$$\psi(b, c) = 0,$$
$$\psi(a, b, c) = 2.$$

The maximal flow problem on this network has two extreme optimal dual solutions, corresponding to the two minimal cuts of the network. These yield the two core allocations

$$x^1_a = 1, \quad x^1_b = 0, \quad x^1_c = 0,$$
$$x^2_a = 0, \quad x^2_b = 1, \quad x^2_c = 1.$$

However the point

$$x^3_a = 1, \quad x^3_b = 1, \quad x^3_c = 0$$

which is also a core allocation, does not correspond to a dual optimal solution. This is not entirely surprising since the relation between the game $\psi$ and the optimisation problem which yield this game is not one to one. A given game may have several
linear programming representations, each possibly yielding a
different optimal dual set. In the following section we present
a certain standardized set of optimization problems for which this
discrepancy does not arise. Another class of problems with this
property was given in Shapley and Shubik, [20].

V. Simple Networks

Consider a directed network \( G = G(E, L) \) with one source and
one sink, \( s \) and \( t \) respectively. For \( j \in L \) let \( u_j \) be the capacity
of this edge and \( c_j \) the associated objective value coefficient.

We do not impose any condition on the sign of \( c_j \). In practical
applications one may expect some components to be negative
reflecting the cost of flow, while others are positive to account
for the associated revenues. Finally, for each arc
\( j \in L \) let \( o_j \in \mathbb{N} \) be the identity of the player which controls
this arc.

The network \( G \) defines a game \( V \) in a natural way. For each
\( s \in G \) denote by \( G^s \) the network restricted to arcs whose owners
belong to \( s \). \( V(s) \) can be defined then as the value of the optimal
(i.e. maximal with respect to \( c^s \) (\( c_j : j \in E \)) \( s \) to \( t \) flow in
\( G^s \). It is a simple matter to observe that the standard arc-flow
formulation of this problem, in which the variables correspond to
the flows on the individual arcs, satisfies the conditions of
Theorem 2.

We call the network \( G \) simple, if \( u_j = 1 \) for every, \( j \in L \) and
if each arc is owned by a different player, i.e. we can identify
the set of arcs with the set of players. We will restrict our
attention in this section to games resulting from such networks.

Let $P = (p_1, \ldots, p_k)$ be the set of all simple paths from $s$ to $t$
on $G$, each regarded as a subset of edges i.e. a coalition. For
each $p \in P$ let $c^p = \sum_{i \in p} c_i$. Obviously $V(p) = \max \{0, c^p\}$ for every
$p \in P$ and $V(i) > 0$ for every $i \in N$.

Thus, every $x \in \text{CORE}(V)$ must satisfy:

\begin{align*}
(6) & \quad x_i = V(N) \\
(7) & \quad x_i \geq c^p \quad p \in P \\
(8) & \quad x_i > 0 \quad i \in N.
\end{align*}

The following proposition asserts that for simple networks the
converse of this statement is also true

**Proposition 1.** Let $V$ be the game associated with a simple network
$G$. Then

$$
\text{CORE}(V) = \{ x : x \text{ satisfies (6), (7) and (8)} \}.
$$

**Proof.** We have to show that $x_S \geq V(S)$ for every coalition
$S \subseteq C$. Let $y^S = (y^S_j : j \in S)$ be an optimal solution which yields
the value $V(S)$, and such that $y^S_j \in (0, 1)$ for each $j \in S$. The fact that such an
integral valued solution exists follows from the fact that the
underlying matrix is unimodular. It follows immediately that the
non-zero elements of $y^S$ define a collection of edge-wise
disjoint paths $F'$ from $s$ to $t$ such that the edges of these paths belong to $S$. Hence,

$$V(S) = \sum_{j \in S} c_j^{F'} \quad \sum_{j \in F^c} c_j^{F} \quad \sum_{j \in F^c} \sum_{j \in S} R_{j}$$

**Remark 3** Proposition 1 lies at the heart of the spectral behavior of simple networks. The reader may wish to verify that the proposition does not hold for networks which are not simple such as the one of Example 1.

**Remark 4** An alternative representation of the core is as the optimal set of the linear program \( \min \sum_{j \in N} x_j \) subject to (7) and (8). This representation yields the following complementary slackness conditions:

(a) If there exists any optimal flow $F$ with $f_j = 0$ for some $j \in N$ then $x_j = 0$ in any core allocation $\alpha$.

(b) If there exists any optimal flow $F$, with $f_j = 1$ for all the arcs of a given path $p$, then $\sum_{j \in p} x_j = c^p$ in every core allocation $\alpha$.

Proposition 1 is of little use from an algorithmic point of view since the cardinality of $P$ is typically huge. Proposition 2 below can serve as a practical basis for deciding, for a given $\alpha \in \mathbb{R}^n$, whether or not $\alpha \in \text{CORE}(V)$.

For $\alpha \in \mathbb{R}^n$, let $G^\alpha$ be the network obtained from $G$ by replacing $c$ with $c - \alpha$.

**Proposition 2** Let $f$ be any optimal flow on $G$ and let
x ∈ R^n_+ satisfy \( \sum_j \frac{f_j}{x_j} = V(x) \). Then \( x \in \text{CORE}(V) \) if and only if \( f \) is optimal for \( G^x \) with optimal value 0.

**Proof.** We first note that changes in the objective function leave \( f \) feasible. Its value with respect to the new objective function is

\[
f^*(x) = \sum_j \frac{f_j(x_j - x_j)}{x_j} = \sum_j \frac{f_j x_j}{x_j} = \sum_j \frac{f_j x_j}{x_j} = V(x) - \sum_j \frac{f_j x_j}{x_j}
\]

assuming that \( x \in \text{CORE}(V) \). Then, by the complementary slackness condition (a) of Remark 2, \( x_j > 0 \) \( \Rightarrow f_j = 1 \). Thus

\[
\sum_j \frac{f_j x_j}{x_j} = \sum_j \frac{f_j x_j}{x_j} - V(x)
\]

To complete the proof we note, that if \( f \) and \( x \) are such that the value of \( f \) w.r.t. network \( G^x \) is zero, \( f \) is optimal in this network iff for every path \( p \in P \)

\[
\sum_j \frac{f_j(x_j - x_j)}{x_j} < 0
\]

i.e. iff

\[
\sum_j \frac{f_j x_j}{x_j} > 0 \text{ for every } p \in P
\]

Using, Proposition 1, we note that the last condition holds iff \( x \in \text{CORE}(V) \).

We finally come to the main theorem of this section. It states that for simple networks, every core allocation
corresponds to an optimal dual solution for the corresponding
optimization problem. We recall the arc-flow formulation of this
problem:

\[(9) \quad \max \sum_{j \in A} c_j x_j \]
\[(10) \quad \text{s.t.} \quad x_j \leq 1 \quad j \in N \]
\[(11) \quad \sum_{j \in \text{IN}_i} x_j - \sum_{j \in \text{OUT}_i} x_j = 0 \quad \text{for every } i \in S \text{ with } s \leq i \leq t. \]
\[(12) \quad x_j > 0 \quad j \in N \]

Where, for each node \(i\), we denote by \(\text{IN}_i\) and \(\text{OUT}_i\) the set of
edges coming into and going out of \(i\) respectively. The linear
programming dual of (9) - (11) is

\[(13) \quad \min \sum_{j \in N} u_j \]
\[(14) \quad \text{s.t.} \quad u_j + \sum_{j \in \text{IN}_i} w_i - \sum_{j \in \text{OUT}_i} w_i \geq c_j \quad j \in N \]
\[(15) \quad u_j > 0 \quad j \in \delta \]

Theorem 3 Let \(u \in \text{CORE}(V)\). Then, there exists \(w = (w_i : i \in \delta)\)
such that \((u, w)\) is an optimal solution for (13) - (15).

Proof \(u \in \text{CORE}(V)\) implies that \(\sum_{j \in N} u_j = V(N)\) and \(u_j > 0\), \(j \in N\).
Hence, all we have to show is that there exists \(w\) such that (14)
is satisfied. Consider the network \(G^u\). By proposition 3, the
optimal value for the optimal flow problem on this network is 0. Let \((u', w')\) be any dual solution with respect to this network. Then

\[
\sum_{j \in N} u'_j = 0
\]

and

\[
u'_j > 0, \quad j \in N
\]

which imply that \(u'_j = 0, \quad j \in N\). By the dual feasibility of this solution we have that

\[
\sum_{j \in I} \text{IN}_j w'_j = \sum_{j \in \text{OUT}_j} w'_j > c_j, \quad j \in N
\]

which in turn implies that

\[
u'_j + \sum_{j \in \text{IN}_j} w'_j > \sum_{j \in \text{OUT}_j} w'_j > c_j
\]

i.e. that \((u, w')\) is the required dual optimal solution.

We conjecture that Theorem 3 can serve as a practical basis for calculating the nucleolus (see Schmeidler [17a]) of the game \(V\). If the objective function of problem (9) - (11) is to maximize flow (i.e. \(c_j = 1, \quad j \in \text{OUT}_j, \quad c_j = 0\) otherwise) then theorem 3 simplifies to

**Theorem 4.** Let \(G\) be a simple network such that for each coalition \(S\), \(V(S)\) is equal to the maximal s to t flow possible using the arcs of \(S\) only. Then the extreme points of \(\text{CORE}(V)\) are precisely the points \(x = (x_j: j \in N)\) such that
\[ x_j = \begin{cases} 1 & \text{if } j \in K \\ 0 & \text{otherwise} \end{cases} \]

where \( K \) is a minimal \( s \) to \( t \) cut in \( G \).

The proof of theorem 4 follows immediately from theorem 3 and from the fact that the minimal cuts of \( G \) constitute the extreme dual solutions to the maximum flow problem on this network. For details see Ford and Fulkerson, [9] and Fulkerson [16].

VI. Examples

We conclude the paper with a few examples.

**Example 2** Consider the network \( G_2 \) of Figure 2, where again each arc is labeled by its name (a letter) and its capacity. If each arc is owned by a different player, the network is simple. Consider the game defined by the max flow problem on this network. By Theorems 3 or 4, we note that the unique point in \( \text{COKE}(V) \) corresponds to the unique minimal cut in \( G_1 \). This point is given by

\[ x_a = x_b = x_c = 0, \quad x_d = x_e = 1 \]

**Example 3** To see the complications which arise when the network is not simple, consider the network \( G_3 \) of Figure 3, which is obtained by letting one player, say \( a \), control the three arcs \( a, b, c \), of \( G_2 \), (or equivalently, replacing these 3 arcs by a unique arc of capacity 1). Again examine the maximal flow
problem on $G_3$. The unique minimum cut in $G_3$ consists, as previously, of the arcs $d$, and $e$. This yields the core allocation

$$x_a = 0, \quad x_d = x_e = 1$$

However, 3 additional extreme points of CORE (V) are

$$x_a = 2, \quad x_d = x_e = 0$$
$$x_a = 1, \quad x_d = x_e > 0$$
$$x_a = 1, \quad x_d = 0, \quad x_e = 1$$

Cores of network games exhibit certain non-monotonicities with respect to the game data. The following two examples demonstrate this behavior.

**Example 1** Consider the network $G_4$ of Figure 4, obtained from $G_2$ by increasing to 2 the capacity of arc $d$. This increase, however, may not be in the interest of player $d$ as the following allocation

$$x_a = x_b = x_e = 1, \quad x_d = x_n = 0$$

belongs to the core of the new game.
Example 5 Consider network $G_5$ of Figure 5. Let all the per unit cost on the arcs be 0 except for $c_a$ and $c_e$ which are set to 2. Consider the game obtained from this network if we value each unit of flow from source to sink at one unit. (This can be achieved, say, by setting $c_a = -2$, $c_b = 0$, $c_c = 0$, $c_d = 1$, $c_e = 1 - 2 = -1$.) The optimal flow in this network is through the path $b, c, d$ and yields an objective function of 1. The extreme points of CORE (7) are given by

$$
\begin{align*}
  x_b &= 1, & x_j &= 0, & j \neq b \\
  x_c &= 1, & x_j &= 0, & j \neq c \\
  x_d &= 1, & x_j &= 0, & j \neq d
\end{align*}
$$

Now let us increase the value of a unit of flow from $s$ to $t$ to 4. A new optimal solution for the problem utilizes the paths $a - d$ and $b - c$. The unique point in the core is now

$$
\begin{align*}
  x_b &= x_d = 2, & x_j &= 0, & j \neq b, d
\end{align*}
$$

Thus, the increase in the per unit revenue is detrimental from the point of view of player $c$. 

-.26-
REFERENCES


