Buyer-Supplier Games: Optimization Over the Core

Nedialko B. Dimitrov* and C. Greg Plaxton**

University of Texas at Austin
1 University Station C0500
Austin, Texas 78712–0233
{ned,plaxton}@cs.utexas.edu

Abstract. In a buyer-supplier game, a special type of assignment game, a distinguished player, called the buyer, wishes to purchase some combinatorial structure. A set of players, called suppliers, offer various components of the structure for sale. Any combinatorial minimization problem can be transformed into a buyer-supplier game. While most previous work has been concerned with characterizing the core of buyer-supplier games, in this paper we study optimization over the set of core vectors. We give a polynomial time algorithm for optimizing over the core of any buyer-supplier game for which the underlying minimization problem is solvable in polynomial time. In addition, we show that it is hard to determine whether a given vector belongs to the core if the base minimization problem is not solvable in polynomial time. Finally, we introduce and study the concept of focus point price, which answers the question: If we are constrained to play in equilibrium, how much can we lose by playing the wrong equilibrium?

1 Introduction

In this paper, we study the core of a large set of games, a subset of assignment games, which we term buyer-supplier games [3, 22] [23, Chapter 6]. We are primarily concerned with efficient computations over the set of vectors belonging to the core of buyer-supplier games. Before diving into an overview of buyer-supplier games, we present some connections between our work and the existing literature.

1.1 Related Work

Though suggested by Edgeworth as early as 1881 [8], the notion of the core was formalized by Gillies and Shapley [11, 21], extending von Neumann and Morgenstern's work on coalitional game theory [24]. Recently, Goemans and Skutella studied the core of a cost sharing facility location game [12]. In their

* Supported by an MCD Fellowship from the University of Texas at Austin.
** Supported by NSF Grants CCR–0310970 and ANI–0326001.

paper, Goemans and Skutella are primarily interested in using core vectors as a cost sharing indicator, to decide how much each customer should pay for opening the facility used by the customer. Goemans and Skutella show that, in general, the core of the cost sharing facility location game they study is empty. In contrast, for the buyer-supplier games we study, the core is always nonempty. Additionally, in our work we do not view vectors in the core as an indication of cost shares but rather as rational outcomes of negotiation amongst the players in the buyer-supplier game. Pál and Tardos extend the work of Goemans and Skutella by developing a mechanism for the cost sharing facility location game which uses the concept of an approximate core [15].

There has been great interest in comparing the game's best outcome to the best equilibrium outcome, where the term best is based on some objective function. For example, one may wish to compare the outcome maximizing the net utility for all players in the game against the best possible Nash equilibrium, with respect to net utility. Papadimitriou termed one such comparative measure as the price of anarchy [16]. Roughgarden and Tardos have studied the price of anarchy in the context of routing [18–20].

In this paper, we introduce a quantity with a similar motivation to that of the price of anarchy. Solution concepts often yield multiple predictions, or equilibria. In actual game play, however, only one of the equilibria can be chosen by the game's players. Experiments show that conditions outside the game, such as societal pressures or undue attention to a specific player, focus the players' attention on the point of a single equilibrium, which then becomes the outcome of the game. This is a common notion in game theory called the focus point. A player may receive different payoffs in different equilibria. How much is the player willing to pay for a good focus point? We define the *focus point price* with respect to a given player as the difference between the maximum and minimum equilibrium payoffs to the player. Stated succinctly, focus point price answers the question: If we are constrained to play in equilibrium, how much can we lose by playing the wrong equilibrium?

Recently, Garg et al. studied transferable utility games they call coalitional games on graphs [10]. Coalitional games on graphs are a proper subset of buyer-supplier games, which can be derived by setting the buyer's internal cost, Bcost, to zero (see Section 1.3 and Lemma 1). For some buyer-supplier games, for example the buyer-supplier facility location game, it does not appear that the game can be described with Bcost fixed to zero.

Garg et al. study the concepts of "frugality" and "agents are substitutes." They show that suppliers are substitutes if and only if the core of the game forms a lattice. In buyer-supplier games, suppliers are not always substitutes. In Lemma 4, we show that if suppliers are substitutes, we can optimize over the core by solving a polynomially sized linear program. Garg et al. and, more recently, Karlin et al. study and characterize the frugality certain auction mechanisms; the focus point price concept introduced in this paper is quite different from frugality [13].

A third difference between Garg et al. and this work comes from the fact that, similarly to the economics literature, Garg et al. are mainly concerned with the characterization of the core: When does the core form a lattice? How do core vectors relate to auctions? We, on the other hand, are mainly concerned with characterizing optimization over the core. Our main results are in the flavor of Deng and Papadimitriou, in that we are interested with the complexity of computing using game theoretic characterizations [6].

Faigle and Kern study optimization over the core for submodular cost partition games [9]. Faigle and Kern exhibit a generic greedy-type algorithm for optimization of any linear function over the core of partition games whose value function is both submodular and *weakly increasing*, a property they define.

The greedy framework of Faigle and Kern captures certain buyer-supplier games, such as the buyer-supplier minimum spanning tree game. However, even some buyer-supplier games derived from problems that admit greedy solutions, such as the buyer-supplier shortest path game, are not amenable to the approach of Faigle and Kern. In this paper, we do not restrict ourselves to greedy algorithms. By making use of the ellipsoid method, we are able to give polynomial time algorithms for optimization over the core of any buyer-supplier game for which the underlying minimization problem is solvable in polynomial time.

To provide the reader with a simple, concrete example of optimization over the core of a buyer-supplier game, towards the end of this paper, we focus our attention on the buyer-supplier minimum spanning tree game. We give a simple greedy algorithm for this problem, which is a minor extension of Kruskal's minimum spanning tree algorithm. A greedy algorithm is provided by the work of Faigle and Kern, but their exposition involves a good deal of machinery. Our exposition is completely elementary.

Several methods, apart from buyer-supplier games, are known for transforming a combinatorial optimization problem into a game. The cores of these transformations have also been extensively studied. For example, Deng et al. show results on core non-emptyness, distinguishability of core vectors, and finding core vectors for one such transformation [5]. Caprara et al. continue the work of Deng et al. by considering a certain optimization over the set of core vectors for this alternate transformation [4].

1.2 Main Contributions

There has been increased interest from the theoretical computer science community in game theory. While problem-specific solutions may give us insight, to leverage the full power of decades of study in both research areas, we must find generic computational solutions to game theoretic problems. Indeed, others have already realized this need [1, 17]. In this paper, we continue this line of work by deriving generic results for computing with core solutions in a large class of games.

The core of buyer-supplier games in the transferable utility setting is characterized by Shapley and Shubik [22]. As a minor contribution, we extend their result by showing that the core in the non-transferable utility setting is the same as the core with transferable utilities. Our primary contributions are as follows:

1. While previous work in the economics literature has concentrated on characterizing the core of buyer-supplier games and relating core vectors to auctions, our main interest is in optimizing over the set of core vectors [3]. We provide a generally applicable algorithm, based on the ellipsoid method, for optimizing over the core. If the original minimization problem is solvable in polynomial time, we show that it is possible to optimize linear functions of core vectors in polynomial time.
2. We fully characterize optimization over the core of buyer-supplier games by using a polynomial time reduction to show that if the original minimization problem is not solvable in polynomial time, it is impossible, in polynomial time, to test if an arbitrary vector is in the core of the buyer-supplier game.
3. We introduce the concept of focus point price. Our positive computational results give a polynomial time algorithm for computing the buyer's focus point price in buyer-supplier games when the underlying minimization problem is solvable in polynomial time. When the underlying minimization problem is not solvable in polynomial time, we show that it is impossible to approximate the buyer's focus point price to within *any* multiplicative factor.

1.3 Overview of Buyer-Supplier Games

The definition of a buyer-supplier game, given in Section 2.1, is self-contained and does not require an argument. However, it is also possible to transform a combinatorial minimization problem into a buyer-supplier game. Consider a combinatorial minimization problem of the following form. We have some finite set of elements $\mathcal{C}$. We designate some subsets of $\mathcal{C}$ as feasible. To capture feasibility, we use a predicate $P : 2^{\mathcal{C}} \rightarrow \{0,1\}$, where the predicate is one on all feasible subsets of $\mathcal{C}$. With each feasible set $\mathcal{A} \subseteq \mathcal{C}$, we associate a nonnegative cost $f(\mathcal{A})$. The combinatorial minimization problem can then be captured by the function $\mathrm{MinProb} : 2^{\mathcal{C}} \rightarrow \Re_+$ defined by

$$\mathrm{MinProb}(\mathcal{B}) = \min_{\substack{\mathcal{A} \subseteq \mathcal{B} \\ P(\mathcal{A}) = 1}} f(\mathcal{A})$$

where $\Re_+$ denotes the nonnegative real numbers.

To transform the above minimization problem into a buyer-supplier game, we associate a player with each element of $\mathcal{C}$; we call such players suppliers. We also add another player whom we call the buyer. In the game, the buyer wishes to purchase a feasible subset of $\mathcal{C}$. The suppliers, on the other hand, are offering their membership to the buyer's set at a price.

To fully specify the game's model of a realistic interaction, we let M designate the maximum investment the buyer is willing to spend on a feasible set. We decompose f such that $f(\mathcal{A}) = \mathrm{Bcost}(\mathcal{A}) + \sum_{a \in \mathcal{A}} \tau(a)$, where $\tau(a)$ is the internal cost for supplier a to be present in the buyer's set and $\mathrm{Bcost}(\mathcal{A})$ is the internal cost to the buyer for purchasing this specific feasible set. In general, many such decompositions are possible, and they produce different games. However, when specifically applying the core solution concept, Lemma 1 shows that all such decompositions are equivalent. Though it is not necessary, to remove special cases in our statements, it is convenient to let $\mathrm{Bcost}(\mathcal{A}) = M$ when $\mathcal{A} = \emptyset$ or $\mathcal{A}$ is not feasible.

Now that we have determined the internal costs for the buyer and the suppliers, we can specify the game. The buyer-supplier game is specified by the tuple $(\mathcal{C}, \tau, \mathrm{Bcost})$. The strategy set for the buyer is the power set of $\mathcal{C}$. By playing $\mathcal{A} \subseteq \mathcal{C}$, the buyer chooses to purchase the membership of the suppliers in $\mathcal{A}$. The strategy set for every supplier $a \in \mathcal{C}$ is the nonnegative real numbers, indicating a bid or payment required from the buyer for the supplier's membership.

For any supplier $a \in \mathcal{C}$, we let $\beta(a)$ denote the associated bid. Let $\mathcal{A}$ be the set of suppliers chosen by the buyer. The payoff for the buyer is $M - \mathrm{Bcost}(\mathcal{A}) - \sum_{a \in \mathcal{A}} \beta(a)$. The payoff for a supplier not in $\mathcal{A}$ is 0. The payoff for a supplier a in $\mathcal{A}$ is $\beta(a) - \tau(a)$.

Since we are applying the solution concept of the core, one may think of the game play as follows. All the players in the game sit down around a negotiating table. All the players talk amongst themselves until they reach an agreement which cannot be unilaterally and selfishly improved upon by any subset of the players. Once such an agreement is reached, game play is concluded. Since no subset of the players can unilaterally and selfishly improve upon the agreement, rationality binds the players to follow the agreement.

The fully formal definition of a buyer-supplier game is given in Section 2.1. The transformation process described above can be used to create buyer-supplier games from most combinatorial minimization problems. For example, minimum spanning tree, Steiner tree, shortest path, minimum set cover, minimum cut, single- and multi-commodity flow can all be used to instantiate a buyer-supplier game. As a concrete example and interpretation of a buyer-supplier game, consider the buyer-supplier minimum spanning tree game. In this game, a company owns factories on every node of a graph. The company wishes to connect the factories by purchasing edges in the graph. Each edge is owned by a unique supplier player. Each supplier has an internal cost associated with the company's usage of the edge. The company has a maximum amount of money it is willing to spend on purchasing edges. Depending on the transportation conditions of a particular edge, the company may have some internal cost associated with choosing that particular edge. The buyer-supplier game paradigm yields similarly natural games when applied to other minimization problems.

In this paper we will be concerned with efficient computation over the set of core vectors. For the rest of the paper, when we say polynomial time, we mean time polynomial in the size of the parameter $\mathcal{C}$, which is also polynomial in the number of players of the buyer-supplier game.

1.4 Organization of the Paper

In Section 2 we define buyer-supplier games and the core of a game. In Section 3 we characterize the core of buyer-supplier games. In Section 4 we give positive computational results, namely the generic algorithm for optimizing over the set of core vectors. In Section 5 we give negative computational results by showing polynomial time equivalence between several related problems. In Section 6 we give the problem-specific combinatorial algorithm for the buyer-supplier game arising from the minimum spanning tree problem.

2 Definitions

2.1 Buyer-Supplier Games

Let $\mathcal{C}$ be a finite set and M be a nonnegative real number. Let τ be a function from $\mathcal{C}$ to $\Re_+$. Let Bcost be a function from $2^{\mathcal{C}}$ to $\Re_+$ such that $\text{Bcost}(\emptyset) = M$. The simplifying condition that $\text{Bcost}(\emptyset) = M$ is not required. We explain the condition's purpose later in this section. For $\mathcal{A} \subseteq \mathcal{C}$, let $\text{Eval}(\tau, \text{Bcost}, \mathcal{A})$ denote $\text{Bcost}(\mathcal{A}) + \sum_{a \in \mathcal{A}} \tau(a)$. For $\mathcal{A} \subseteq \mathcal{C}$, let $\text{MinEval}(\tau, \text{Bcost}, \mathcal{A})$ denote $\min_{\mathcal{B} \subseteq \mathcal{A}} \text{Eval}(\tau, \text{Bcost}, \mathcal{B})$. We will omit the parameters τ and Bcost from the functions $\text{Eval}(\tau, \text{Bcost}, \mathcal{A})$ and $\text{MinEval}(\tau, \text{Bcost}, \mathcal{A})$ when the value is clear.

Given a tuple $(\mathcal{C}, \tau, \text{Bcost})$, we proceed to define a buyer-supplier game. Associate a player with each element of $\mathcal{C}$. Call the players in $\mathcal{C}$ suppliers. Let there also be another player, μ, whom we call the buyer. Let $\mathcal{P} = \mathcal{C} \cup \{\mu\}$ be the set of players for the buyer-supplier game.

The strategy for supplier a is a tuple $(\beta(a), p_a)$ with $\beta(a) \in \Re_+$ and $p_a : \mathcal{P} \to \Re_+$. The first element, $\beta(a)$, represents supplier a's bid to the buyer, requiring the buyer to pay $\beta(a)$ for using the supplier's services. The second element, p_a, represents the nonnegative side payments supplier a chooses to make to the game's players. By $p_a(b)$ we denote the side payment a makes to player b.

The strategy for the buyer, μ, is a tuple $(\mathcal{A}, p_\mu)$ where $\mathcal{A} \in 2^{\mathcal{C}}$ and $p_\mu : \mathcal{P} \to \Re_+$. The first element, $\mathcal{A}$, represents the suppliers chosen by the buyer for a purchase. Similarly to a supplier, the second element, p_μ, represents the nonnegative side payments the buyer chooses to make to the game's players.

For each player $a \in \mathcal{P}$ we denote the player's strategy set by $\mathcal{S}_a$. For a set of players $\mathcal{A} \subseteq \mathcal{P}$, we denote the set of strategies $\bigotimes_{a \in \mathcal{A}} \mathcal{S}_a$ by $\mathcal{S}_{\mathcal{A}}$. We call elements of $\mathcal{S}_{\mathcal{A}}$ strategy vectors. We index strategy vectors from $\mathcal{S}_{\mathcal{A}}$ by the elements of $\mathcal{A}$.

We now define the utility function for each player. Suppose strategy $s \in \mathcal{S}_{\mathcal{P}}$ is played. Specifically, suppose that $(\mathcal{A}, p_\mu) \in \mathcal{S}_\mu$ and $(\beta(a), p_a) \in \mathcal{S}_a$ for each $a \in \mathcal{C}$ are played. The utility function for buyer is $u_\mu(s) = M - [\text{Bcost}(\mathcal{A}) + \sum_{a \in \mathcal{A}} \beta(a)] + [\sum_{b \in \mathcal{P}} p_b(\mu) - \sum_{b \in \mathcal{P}} p_\mu(b)]$. The utility for a supplier a in $\mathcal{A}$ is $u_a(s) = \beta(a) - \tau(a) + [\sum_{b \in \mathcal{P}} p_b(a) - \sum_{b \in \mathcal{P}} p_a(b)]$. The utility for a supplier a not in $\mathcal{A}$ is $u_a(s) = [\sum_{b \in \mathcal{P}} p_b(a) - \sum_{b \in \mathcal{P}} p_a(b)]$.

Interpreting, the buyer begins with a total of M utility and chooses to make a purchase from each supplier in $\mathcal{A}$. The buyer gives $\beta(a)$ to each supplier $a \in \mathcal{A}$

and loses an extra $\mathrm{Bcost}(\mathcal{A})$ from the initial M utility. Each supplier a in $\mathcal{A}$ receives the bid payment from the buyer and loses $\tau(a)$ because the supplier must perform services for the buyer. The distribution of sidepayments completes the utility functions. The requirement that $\mathrm{Bcost}(\emptyset) = M$ lets the strategy $\emptyset$ stand as a "don't play" strategy for the buyer. To remove the requirement, we could introduce a specific "don't play" strategy to the buyer's strategy set, however this creates a special case in most of our proofs.

Let the sidepayment game we have defined be denoted SP. Let NOSP denote the same game with the additional requirement that all sidepayments be fixed to zero. In other words, in NOSP we restrict the strategy set for each $a \in \mathcal{P}$ so that p_a is identically zero.

2.2 Game Theoretic Definitions

All of the definitions in this section closely follow those of Shubik [23, Chapter 6].

We call a vector in $\Re^{|\mathcal{P}|}$, indexed by $a \in \mathcal{P}$, a *payoff vector*.

Let π be a payoff vector and s be a strategy vector in $\mathcal{S}_{\mathcal{A}}$ for $\mathcal{A} \subseteq \mathcal{P}$. Let t be any strategy vector in $\mathcal{S}_{\mathcal{P}}$ such that the projection of t onto the coordinates in $\mathcal{A}$ is equal to s. If for all t and for all $a \in \mathcal{A}$ we have $\pi_a \leq u_a(t)$, we say that the players in $\mathcal{A}$ can *guarantee* themselves payoffs of at least π by playing s.

We use Shubik's alpha theory to define our characteristic sets [23, pp. 134-136]. Thus for a set of players $\mathcal{A} \subseteq \mathcal{P}$, we define the characteristic set, $V(\mathcal{A})$, to be the set of all payoff vectors π such that there is a strategy vector $s \in \mathcal{S}_{\mathcal{A}}$, possibly dependent on π, with which the players in $\mathcal{A}$ can guarantee themselves payoffs of at least π. In the transferable utility setting, SP, the characteristic sets can be replaced with a characteristic function. Given the definitions of the utility functions in Section 2.1, the characteristic function $\tilde{V}(\mathcal{A})$ for a set of players $\mathcal{A}$ is equal to $M - \mathrm{MinEval}(\tau, \mathrm{Bcost}, \mathcal{A} - \{\mu\})$.

We say that a set $\mathcal{A} \subseteq \mathcal{P}$ of *players are substitutes* if $\tilde{V}(\mathcal{P}) - \tilde{V}(\mathcal{P} - \mathcal{B}) \geq \sum_{a \in \mathcal{B}} \tilde{V}(\mathcal{P}) - \tilde{V}(\mathcal{P} - \{a\})$ for all $\mathcal{B} \subseteq \mathcal{A}$.

We say that a payoff vector π dominates a payoff vector ν through a set $\mathcal{A} \subseteq \mathcal{P}$ if $\pi_a > \nu_a$ for all $a \in \mathcal{A}$. In other words, π dominates ν through $\mathcal{A}$ when each player in $\mathcal{A}$ does better in π than in ν.

For a set of players $\mathcal{A} \subseteq \mathcal{P}$, we define $D(\mathcal{A})$ as the set of all payoff vectors which are dominated through $\mathcal{A}$ by a payoff vector in $V(\mathcal{A})$. Interpreting, the players in $\mathcal{A}$ would never settle for a payoff vector $\pi \in D(\mathcal{A})$ since they can guarantee themselves higher payoffs than those offered in π.

The *core* of a game consists of all $\pi \in V(\mathcal{P})$ such that $\pi \notin D(\mathcal{A})$ for all $\mathcal{A} \subseteq \mathcal{P}$.

3 A Characterization of the Core

The characterazation of the core of buyer-supplier games in the transferable utility setting was done by Shapley and Shubik [22]. In this seciton, we show the surprising result that the same characterization holds in the non-transferable utility setting. In general, it is not the case that the core of the transferable utility and non-transferable utility versions of a game are the same. For example, the buyer may be able to use bribes to alter the bidding strategies of some suppliers, and thus reduce the bids of other suppliers. The following theorem characterizes the core of buyer-supplier games.

Theorem 1. *A payoff vector π is in the core of a buyer-supplier game defined by $(\mathcal{C}, \tau, \mathrm{Bcost})$ if and only if it satisfies*

$$\pi_a \geq 0 \qquad \text{for all } a \in \mathcal{P}, \quad (1)$$

$$\sum_{a \in \mathcal{A}} \pi_a \leq \mathrm{MinEval}(\tau, \mathrm{Bcost}, \mathcal{C} - \mathcal{A}) - \mathrm{MinEval}(\tau, \mathrm{Bcost}, \mathcal{C}) \qquad \text{for all } \mathcal{A} \subseteq \mathcal{C}, \quad (2)$$

$$\pi_\mu = M - \mathrm{MinEval}(\tau, \mathrm{Bcost}, \mathcal{C}) - \sum_{a \in \mathcal{C}} \pi_a. \quad (3)$$

Because of space considerations, here and in the rest this paper we choose to present the intuition and a proof sketch for most of the stated results. Fully detailed proofs of all results are presented in the companion technical report [7].

We take as a given the result by Shapley and Shubik, which shows that under transferable utilities, the core is characterized by Theorem 1.

The intuition for the equivalence of the transferable utility core and the non-transferable utility core is as follows. Consider a payoff vector π satisfying Equation (1). A set of suppliers can only guarantee zero payoffs for themselves. Thus, for a set of players $\mathcal{A}$ to be able to truly improve upon the payoffs given in π, the buyer must be in $\mathcal{A}$. However, if the buyer is in $\mathcal{A}$, the players in $\mathcal{A}$ can simulate sidepayments amongst themselves by having the suppliers in $\mathcal{A}$ alter their bids to the buyer. Thus, the sidepayments do not add any additional power to the set of players $\mathcal{A}$.

As a corollary to Theorem 1, we have the following lemma, which shows that the core does not change depending on the decomposition chosen in the transformation from a combinatorial minimization problem to a buyer-supplier game.

Lemma 1. *Let* $\mathrm{Bcost}^*(\mathcal{A}) = \sum_{a \in \mathcal{A}} \tau(a) + \mathrm{Bcost}(\mathcal{A})$. *The core of the buyer supplier-games defined by* $(\mathcal{C}, \tau, \mathrm{Bcost})$ *and* $(\mathcal{C}, 0, \mathrm{Bcost}^*)$ *is the same.*

4 Polynomial Time Optimization Over the Core Vectors

We define the separation problem on a set of linear inequalities $\mathcal{A}$ as follows. Given a vector π, if π satisfies all of the inequalities in $\mathcal{A}$, then do nothing; otherwise, output a violated inequality $a \in \mathcal{A}$. It is well known that the separation problem is polynomial time equivalent to linear function optimization over the same set of inequalities [14, p. 161].

Let $(\mathcal{C}, \tau, \mathrm{Bcost})$ define a buyer-supplier game. In this section, to simplify the notation, we will omit the parameter Bcost from Eval and MinEval since it is fixed by the buyer-supplier game.

In this section, we will analyze an algorithm to solve the separation problem for the exponentially sized set of inequalities given in Equations (1), (2), and (3). We now give the algorithm, which we call the separation algorithm. Given the payoff vector π as input,

1 Iterate over Equations (1) and (3) to check that they hold. If some equation does not hold, output that equation and halt.
2 Compute $\mathcal{F} \subseteq \mathcal{C}$ such that $\mathrm{Eval}(\tau, \mathcal{F}) = \mathrm{MinEval}(\tau, \mathcal{C})$. If there is some $a \in \mathcal{C} - \mathcal{F}$ with $\pi_a > 0$, output the inequality from Equation (2) corresponding to $\{a\}$ and halt.
3 Define $\hat{\tau}(a) = \tau(a) + \pi_a$ for $a \in \mathcal{C}$. Now, compute $\hat{\mathcal{F}} \subseteq \mathcal{C}$ such that $\mathrm{Eval}(\hat{\tau}, \hat{\mathcal{F}}) = \mathrm{MinEval}(\hat{\tau}, \mathcal{C})$. If $\mathrm{Eval}(\hat{\tau}, \hat{\mathcal{F}}) < \mathrm{Eval}(\hat{\tau}, \mathcal{F})$, output the inequality from Equation (2) corresponding to $\mathcal{F} - \hat{\mathcal{F}}$. Otherwise, halt.

Theorem 2. *If given an input $\hat{\tau} : \mathcal{C} \rightarrow \Re_+$ it is possible to compute both $\mathrm{Eval}(\hat{\tau}, \mathcal{A})$ for any $\mathcal{A} \subseteq \mathcal{C}$ and $\mathcal{F} \subseteq \mathcal{C}$ such that $\mathrm{Eval}(\hat{\tau}, \mathcal{F}) = \mathrm{MinEval}(\hat{\tau}, \mathcal{C})$ in polynomial time, then the separation problem for Equations (1), (2), and (3) is solvable in polynomial time. By the equivalence of separation and optimization, optimizing any linear function of π over Equations (1), (2), and (3) is also possible in polynomial time.*

Proof. It is clear that given the theorem's assumptions, the separation algorithm runs in polynomial time. The statement follows from Lemmas 2 and 3.

Lemma 2. *If the separation algorithm returns an inequality on input π, then π violates the returned inequality.*

Proof. If the algorithm returns an inequality in step 1, then the inequality is violated since the algorithm performed a direct check.

If the algorithm returns an inequality in step 2, then the inequality is violated since $\pi_a > 0$, but $\mathrm{MinEval}(\tau, \mathcal{C} - a) = \mathrm{MinEval}(\tau, \mathcal{C}) = \mathrm{Eval}(\tau, \mathcal{F})$.

Suppose the algorithm returns an inequality in step 3. Thus, $\mathrm{Eval}(\hat{\tau}, \hat{\mathcal{F}}) < \mathrm{Eval}(\hat{\tau}, \mathcal{F})$. By applying the definitions of Eval and $\hat{\tau}$, we have $\sum_{a \in \hat{\mathcal{F}}} \pi_a + \mathrm{Eval}(\tau, \hat{\mathcal{F}}) < \sum_{a \in \mathcal{F}} \pi_a + \mathrm{Eval}(\tau, \mathcal{F})$.

Since the algorithm reaches step 3, we know that $\pi_a = 0$ for all $a \in \mathcal{C} - \mathcal{F}$. Thus, we have $\sum_{a \in \hat{\mathcal{F}} \cap \mathcal{F}} \pi_a + \mathrm{Eval}(\tau, \hat{\mathcal{F}}) < \sum_{a \in \mathcal{F}} \pi_a + \mathrm{Eval}(\tau, \mathcal{F})$, which in turn gives $\mathrm{Eval}(\tau, \hat{\mathcal{F}}) - \mathrm{Eval}(\tau, \mathcal{F}) < \sum_{a \in \mathcal{F} - \hat{\mathcal{F}}} \pi_a$.

Let $\mathcal{A} = \mathcal{F} - \hat{\mathcal{F}}$. From the algorithm, we know that the set $\mathcal{F}$ satisfies $\mathrm{Eval}(\tau, \mathcal{F}) = \mathrm{MinEval}(\tau, \mathcal{C})$. Since $\hat{\mathcal{F}} \subseteq \mathcal{C} - \mathcal{A}$, the definition of MinEval implies that $\mathrm{MinEval}(\tau, \mathcal{C} - \mathcal{A}) \leq \mathrm{Eval}(\tau, \hat{\mathcal{F}})$. Thus, we have $\mathrm{MinEval}(\tau, \mathcal{C} - \mathcal{A}) - \mathrm{MinEval}(\tau, \mathcal{C}) \leq \mathrm{Eval}(\tau, \hat{\mathcal{F}}) - \mathrm{Eval}(\tau, \mathcal{F}) < \sum_{a \in \mathcal{A}} \pi_a$, which shows that the inequality output by the algorithm is violated.

Lemma 3. *If π violates some inequality in Equations (1), (2), and (3), then the separation algorithm run on input π returns an inequality.*

Proof. If the violation is in Equations (1) or (3), the violated inequality will be output by the direct check in step 1. If some inequality is output by step 2, we are done. Otherwise, since steps 1 and 2 output no inequality, we know that $\pi_a = 0$ for all $a \in \mathcal{C} - \mathcal{F}$, where $\mathcal{F}$ is as computed in the algorithm.

Now, suppose the inequality from Equation (2) for set $\mathcal{A} \subseteq \mathcal{C}$ is violated. In other words, we have, $\sum_{a\in\mathcal{A}} \pi_a > \text{MinEval}(\tau, \mathcal{C} - \mathcal{A}) - \text{MinEval}(\tau, \mathcal{C})$. Let $\mathcal{B}$ be such that $\text{Eval}(\tau, \mathcal{B}) = \text{MinEval}(\tau, \mathcal{C} - \mathcal{A})$.

Thus, we have $\sum_{a\in\mathcal{A}} \pi_a > \text{MinEval}(\tau, \mathcal{C} - \mathcal{A}) - \text{MinEval}(\tau, \mathcal{C}) = \text{Eval}(\tau, \mathcal{B}) - \text{Eval}(\tau, \mathcal{F})$.

Since $\pi_a = 0$ for all $a \in \mathcal{C} - \mathcal{F}$, we have $\text{Eval}(\tau, \mathcal{F}) + \sum_{a\in\mathcal{F}\cap\mathcal{A}} \pi_a > \text{Eval}(\tau, \mathcal{B})$.

Adding $\sum_{a\in\mathcal{F}-\mathcal{A}} \pi_a$ to both sides of the above inequality and substituting the definition of Eval, we have $\text{Bcost}(\mathcal{F}) + \sum_{a\in\mathcal{F}} \tau(a) + \sum_{a\in\mathcal{F}} \pi_a > \text{Bcost}(\mathcal{B}) + \sum_{a\in\mathcal{B}} \tau(a) + \sum_{a\in\mathcal{F}-\mathcal{A}} \pi_a$.

Since $\pi_a = 0$ for all $a \in \mathcal{C} - \mathcal{F}$ and $\mathcal{B} \subseteq \mathcal{C} - \mathcal{A}$, we can alter the right hand side of the above inequality to get $\text{Bcost}(\mathcal{F}) + \sum_{a\in\mathcal{F}} \tau(a) + \sum_{a\in\mathcal{F}} \pi_a > \text{Bcost}(\mathcal{B}) + \sum_{a\in\mathcal{B}} \tau(a) + \sum_{a\in\mathcal{B}} \pi_a + \sum_{a\in\mathcal{F}-\mathcal{A}-\mathcal{B}} \pi_a$.

By applying the definition of $\hat{\tau}$ and Eval, we have $\text{Eval}(\hat{\tau}, \mathcal{F}) > \text{Eval}(\hat{\tau}, \mathcal{B}) + \sum_{a\in\mathcal{F}-\mathcal{A}-\mathcal{B}} \pi_a$. We know that $\pi_a \geq 0$ for all $a \in \mathcal{P}$ since the algorithm does not output anything in step 1. Thus, $\text{Eval}(\hat{\tau}, \mathcal{F}) > \text{Eval}(\hat{\tau}, \mathcal{B}) \geq \text{MinEval}(\hat{\tau}, \mathcal{C}) = \text{Eval}(\hat{\tau}, \hat{\mathcal{F}})$, where $\hat{\mathcal{F}}$ is as computed in the algorithm. So, step 3 outputs an inequality.

The following lemma illustrates a key difference between Garg et al. and this work.

Lemma 4. *If suppliers are substitutes, then all but the $|\mathcal{C}|$ singleton equations of Equation (2) are not constraining. Thus, if suppliers are substitutes, optimization over the core of the buyer-supplier game is reduced to solving a polynomially sized linear program.*

Proof. Suppose that the suppliers are substitutes. By the definition of suppliers are substitutes, we have that $\tilde{V}(\mathcal{P}) - \tilde{V}(\mathcal{P} - \mathcal{A}) \geq \sum_{a\in\mathcal{A}}[\tilde{V}(\mathcal{P}) - \tilde{V}(\mathcal{P} - \{a\})]$ for all $\mathcal{A} \subseteq \mathcal{C}$. By the definition of $\tilde{V}$, we have $\text{MinEval}(\tau, \text{Bcost}, \mathcal{C} - \mathcal{A}) - \text{MinEval}(\tau, \text{Bcost}, \mathcal{C}) \geq \sum_{a\in\mathcal{A}}[\text{MinEval}(\tau, \text{Bcost}, \mathcal{C} - \{a\}) - \text{MinEval}(\tau, \text{Bcost}, \mathcal{C})]$ for all $\mathcal{A} \subseteq \mathcal{C}$. This implies that if the singleton equations in Equation (2) are satisfied, then so are all equations in Equation (2). Thus, if suppliers are substitutes, we may drop all non-singleton equations from Equation (2) and reduce the number of inequalities to a polynomial in the number of players.

5 Inapproximability of Optimization Over Core Solutions

Consider a buyer-supplier game defined by $(\mathcal{C}, \tau, \text{Bcost})$. We introduced the concept of the focus point price in the introduction. The concept leads us to ask the natural question: What is the difference between the best and worst core outcome for the buyer? In other words, the value of interest is the solution to the linear program: maximize $\sum_{a\in\mathcal{C}} \pi_a$ subject to Equations (1), (2), and (3).

This natural question leads us to define the focus point price (FFP) problem as follows: on input $(\mathcal{C}, \tau, \mathrm{Bcost})$, output the optimal value of the afore mentioned linear program.

Define the Necessary Element (NEL) problem as follows. Given parameters $(\mathcal{C}, \tau, \mathrm{Bcost})$ return TRUE if there exist an element $a \in \mathcal{C}$ such that for all $\mathcal{F} \subseteq \mathcal{C}$ satisfying $\mathrm{Eval}(\tau, \mathrm{Bcost}, \mathcal{F}) = \mathrm{MinEval}(\tau, \mathrm{Bcost}, \mathcal{C})$ we have $a \in \mathcal{F}$. Otherwise, return FALSE.

Define the OPT-SET problem as follows. Given parameters $(\mathcal{C}, \tau, \mathrm{Bcost})$, return $\mathcal{F}$ such that $\mathrm{Eval}(\tau, \mathrm{Bcost}, \mathcal{F}) = \mathrm{MinEval}(\tau, \mathrm{Bcost}, \mathcal{C})$.

In this section, we will show that the FPP problem, the OPT-SET problem and the NEL problem are polynomial time equivalent. Again, because of space considerations we choose to present some intuition and a proof sketch. For the fully detailed proofs, see the companion technical report [7].

For a fixed tuple $(\mathcal{C}, \tau, \mathrm{Bcost})$ we say we extend the tuple to contain a *shadow element* for an element $a \subseteq \mathcal{C}$ by creating the extended tuple $(\hat{\mathcal{C}}, \hat{\tau}, \mathrm{Bcost}^*)$, where $\hat{\mathcal{C}} = \mathcal{C} \cup b$ with $b \notin \mathcal{C}$; $\hat{\tau}$ is the same as τ with the addition that $\hat{\tau}(b) = \tau(a)$; and for $\mathcal{A} \subseteq \hat{\mathcal{C}}$, if $b \notin \mathcal{A}$, then $\mathrm{Bcost}^*(\mathcal{A}) = \mathrm{Bcost}(\mathcal{A})$, otherwise $\mathrm{Bcost}^*(\mathcal{A}) = \mathrm{Bcost}((\mathcal{A} - \{b\}) \cup \{a\})$. We call b the *shadow element* corresponding to a.

The *full shadow extension* of $(\mathcal{C}, \tau, \mathrm{Bcost})$ is the tuple $(\hat{\mathcal{C}}, \hat{\tau}, \mathrm{Bcost}^*)$ resulting from extending $(\mathcal{C}, \tau, \mathrm{Bcost})$ to contain a shadow element for each element in $\mathcal{C}$.

First, we reduce OPT-SET to NEL. To show the result, we analyze the following algorithm, which we call the shadow algorithm.

On input $(\mathcal{C}, \tau, \mathrm{Bcost})$,

1 Let $(\hat{\mathcal{C}}, \hat{\tau}, \mathrm{Bcost}^*)$ be the full shadow extension of $(\mathcal{C}, \tau, \mathrm{Bcost})$.
2 For each $a \in \mathcal{C}$
  - Remove a's corresponding shadow element from $\hat{\mathcal{C}}$.
  - Run NEL on $(\hat{\mathcal{C}}, \hat{\tau}, \mathrm{Bcost}^*)$.
  - If the return value is TRUE, then add the shadow element back to $\hat{\mathcal{C}}$.
  - If the return value is FALSE, then remove a from $\hat{\mathcal{C}}$.
3 Return $\hat{\mathcal{C}} \cap \mathcal{C}$. In other words, we return all elements from $\mathcal{C}$ remaining in $\hat{\mathcal{C}}$, disregarding any shadow elements.

Lemma 5. *Let* $(\mathcal{C}, \tau, \mathrm{Bcost})$ *be the input to the shadow algorithm. Also, let* $(\hat{\mathcal{C}}, \hat{\tau}, \mathrm{Bcost}^*)$ *be the full shadow extension of* $(\mathcal{C}, \tau, \mathrm{Bcost})$. *If for all* $\mathcal{A} \subseteq \hat{\mathcal{C}}$ *the NEL problem on input* $(\mathcal{A}, \hat{\tau}, \mathrm{Bcost}^*)$ *is solvable in polynomial time, then the OPT-SET problem on input* $(\mathcal{C}, \tau, \mathrm{Bcost})$ *is solvable in polynomial time.*

Given the lemma assumptions, a simple analysis shows that the shadow algorithm runs in polynomial time. The rest of the proof comes in two steps. First, the shadow algorithm maintains the invariant $\mathrm{MinEval}(\tau, \mathrm{Bcost}, \mathcal{C}) = \mathrm{MinEval}(\hat{\tau}, \mathrm{Bcost}^*, \hat{\mathcal{C}})$. This is true because we only remove an element from $\hat{\mathcal{C}}$ if there is an optimal set that does not contain the element. Second, if a remains in $\hat{\mathcal{C}}$ at the end of the iteration associated with a, then it can be shown that a is contained in all subsets of $\hat{\mathcal{C}} \cap \mathcal{C}$ that are solutions to the OPT-SET problem on input $(\mathcal{C}, \tau, \mathrm{Bcost})$.

The following lemma captures the relationship between the FPP problem and the NEL problem.

Lemma 6. *The solution to the FPP problem on input* $(\mathcal{C}, \tau, \text{Bcost})$ *is 0 if and only if the solution to the NEL problem on input* $(\mathcal{C}, \tau, \text{Bcost})$ *is FALSE. Thus, if it is possible to approximate the the FPP problem on input* $(\mathcal{C}, \tau, \text{Bcost})$ *within any multiplicative factor in polynomial time, then the NEL problem on input* $(\mathcal{C}, \tau, \text{Bcost})$ *is solvable in polynomial time.*

The intuition behind this lemma is that if the solution to NEL is TRUE, then there is some element a that is in all OPT-SET solutions on input $(\mathcal{C}, \tau, \text{Bcost})$. In this case, the solution to the FPP problem is at least the difference between the value of an OPT-SET solution on input $(\mathcal{C}, \tau, \text{Bcost})$ and the value of an OPT-SET solution on input $(\mathcal{C} - \{a\}, \tau, \text{Bcost})$. On the other hand, if the solution to NEL is FALSE, then the right hand sides of all singleton equations from Equation (2) are zero, and thus the FPP problem solution is also zero.

A set of $(\mathcal{C}, \tau, \text{Bcost})$ instances is *proper* if the following conditions hold:

– Given that $(\mathcal{C}, \tau, \text{Bcost})$ is in the set, then so is $(\mathcal{C}, \hat{\tau}, \text{Bcost})$, where $\hat{\tau}(a) = \tau(a) + \pi_a$ for a vector $\pi \in \Re_+^{|\mathcal{C}|}$.
– Given that $(\mathcal{C}, \tau, \text{Bcost})$ is in the set, then so is $(\mathcal{A}, \hat{\tau}, \text{Bcost}^*)$, where $\mathcal{A}$ is a subset of $\hat{\mathcal{C}}$ and $(\hat{\mathcal{C}}, \hat{\tau}, \text{Bcost}^*)$ is the full shadow extension of $(\mathcal{C}, \tau, \text{Bcost})$.

The definition of proper instances has a natural interpretation when applied to the transformations of combinatorial minimization problems to buyer-supplier games. For example, for the shortest path problem, the first condition implies that the set of instances is closed with respect to lengthening the edges of the graph. On the other hand, the second condition implies that the set of instances is closed with respect to adding parallel edges or removing a subset of the edges.

The results of Section 4 and the relationships we have given in this section lead us to the following theorem.

Theorem 3. *On a proper set of instances, the separation problem over Equations (1), (2), and (3), the NEL problem and the OPT-SET problem are polynomial time equivalent.*

Lemma 6 in combination with Theorem 3 gives us the following inapproximability result.

Lemma 7. *On a proper set of instances, if it is not possible to solve the OPT-SET problem in polynomial time, it is not possible to approximate the solution to the FPP problem to within any multiplicative factor in polynomial time.*

6 A Complementary Combinatorial Algorithm

In this section, we present an efficient combinatorial algorithm for solving the FPP problem for the buyer-supplier minimum spanning tree (MST) game.

Let a graph $G = (\mathcal{V}, \mathcal{E})$ and edge weights $w : \mathcal{E} \to \Re_+$ be given. Let MSTVal : $2^{\mathcal{E}} \to \Re_+$ be a function that takes as input a set of the edges $\mathcal{A} \subseteq \mathcal{E}$ and returns the weight of the minimum spanning tree of the graph induced by the edges of $\mathcal{A}$. If no spanning tree exists, MSTVal returns ∞.

By the transformation in Section 1 and Lemma 1 in the buyer-supplier minimum spanning tree game, we have $\mathcal{C} = \mathcal{E}$, $\tau(a) = w(a)$, and $\text{Bcost}(\mathcal{A}) = M$ if $\mathcal{A}$ does not connect all nodes in $\mathcal{V}$, or 0 otherwise. We omit the parameters τ and Bcost from MinEval, since they are fixed by the game.

Call the linear program from the FPP problem for the given game LP1, and let its optimal value be O_1. Consider the linear program: maximize $\sum_{b \in \mathcal{C}} \pi_b$ subject to $\sum_{b \in \mathcal{A}} \pi_b \leq \text{MinProb}(\mathcal{C} - \mathcal{A}) - \text{MinProb}(\mathcal{C})$ for all $\mathcal{A} \subseteq \mathcal{C}$ and $\pi_b \geq 0$ for all $b \in \mathcal{C}$. Call the linear program from the previous sentence LP2, and let its optimal value be O_2.

We are able to prove the following relationship between LP1 and LP2. If $\text{MinProb}(\mathcal{C}) \geq M$, then $O_1 = 0$. If $\text{MinProb}(\mathcal{C}) < M$ and $O_2 \leq M - \text{MinProb}(\mathcal{C})$, then $O_1 = O_2$. If $\text{MinProb}(\mathcal{C}) < M$ and $O_2 > M - \text{MinProb}(\mathcal{C})$, then $O_1 = M - \text{MinProb}(\mathcal{C})$. When considering the FPP problem arising from the buyer-supplier game for a specific minimization problem, it may often be helpful to consider LP2 instead of LP1. In fact, the combinatorial algorithm we present finds the optimal value for LP2.

The key insight behind the combinatorial algorithm for the FPP problem for the buyer-supplier MST game is the following. Let T be an MST of G. Suppose edges e_1 and e_2 are edges in T. Suppose the removal of the individual edge e_1 (e_2) increases the MST cost by λ_1 (λ_2). Then, the removal of both edges increases the MST cost by at least $\lambda_1 + \lambda_2$. This insight leads Bikhchandani et al. to show that for the buyer-supplier MST game, suppliers are substitutes [2]. Their result along with Lemma 4 shows that the singleton inequalities of LP2 are an optimal basis. Thus, all our combinatorial algorithm must calculate is the increase in the MST cost associated with the removal of each edge in T.

We give a modified Kruskal Algorithm which can be used to compute the optimal value of LP2. The modifications are as follows. Throughout the algorithm's execution we will keep an auxiliary set of edges, $\mathcal{A}$, which is initially empty. When edge e is added to the minimum spanning forest, also add e to the set $\mathcal{A}$. Suppose edge e is rejected from addition to the minimum spanning forest because it creates a cycle. Let the cycle created be $H = (\mathcal{V}', \mathcal{E}')$. For each edge $a \in \mathcal{E}' - \{e\}$, if $a \in \mathcal{A}$, label a with $w(e) - w(a)$ and remove a from $\mathcal{A}$. The labels computed by the algorithm are the required increases in the MST cost.

References

1. A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 482–491, Oct. 2001.
2. S. Bikhchandani, S. de Vries, J. Schummer, and R. Vohra. Linear programming and vickrey auctions. In *Mathematics of the Internet: E-Auction and Markets*, pages 75–115. Springer-Verlag, New York, NY, 2002.
3. S. Bikhchandani and J. Ostroy. The package assignment model. *Journal of Economic Theory*, 107:377–406, 2002.

4. A. Caprara and A. N. Letchford. Computing good allocations for combinatorial optimization games. Available from http://www.lancs.ac.uk/staff/letchfoa/pubs.htm, 2006.
5. X. Deng, T. Ibaraki, and H. Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24:751–766, 1999.
6. X. Deng, C. Papadimitriou, and Safra S. On the complexity of equilibria. In *Proceedings of the 34th annual ACM symposium on Theory of Computing*, pages 67–71, Montreal, Quebec, 2002.
7. N. B. Dimitrov and C. G. Plaxton. Buyer-supplier games: Core characterization and computation. Technical Report TR–06–19, Department of Computer Science, University of Texas at Austin, April 2006.
8. F. Y. Edgeworth. *Mathematical psychics, an essay on the application of mathematics to the moral sciences*. A. M. Kelley, New York, NY, 1961.
9. U. Faigle and W. Kern. On the core of ordered submodular cost games. *Mathematical Programming*, 87:483–499, 2000.
10. R. Garg, V. Kumar, A. Rudra, and A. Verma. Coalitional games on graphs: core structure, substitutes and frugality. In *Proceedings of the 4th ACM conference on Electronic Commerce*, pages 248–249, San Diego, CA, 2003.
11. D. B. Gillies. *Some Theorems on n-Person Games*. PhD thesis, Princeton University, 1953.
12. M. X. Goemans and M. Skutella. Cooperative facility location games. *Journal of Algorithms*, 50:194–214, 2004.
13. A. R. Karlin, D. Kempe, and T. Tamir. Beyond VCG: frugality of truthful mechanisms. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 615–624, Oct. 2005.
14. G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley and Sons, New York, NY, 1988.
15. M. Pál and E. Tardos. Group strategy proof mechanisms via primal-dual algorithms. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 584–593, Oct. 2003.
16. C. H. Papadimitriou. Algorithms, games, and the internet. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 749–753, July 2001.
17. C. H. Papadimitriou and T. Roughgarden. Computing equilibria in multi-player games. In *Proceedings of the 16th Annual ACM-SIAM symposium on Discrete algorithms*, pages 82–91, January 2005.
18. T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, Cambridge, MA, 2005.
19. T. Roughgarden and E. Tardos. How bad is selfish routing? *J. ACM*, 49:236–259, 2002.
20. T. Roughgarden and E. Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behaviour*, 47:389–403, 2004.
21. L. S. Shapley. Notes on the n-person game III: Some variants of the von Neumann-Morgenstern definition of solution. Research memorandum, RAND Corporation, Santa Monica, CA, 1952.
22. L. S. Shapley and M. Shubik. The assignment game i: The core. *International Journal of Game Theory*, 1:111–130, 1972.
23. M. Shubik. *Game Theory In The Social Sciences*. MIT Press, Cambridge, Massachussetts, 1984.
24. J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1953.