

Implementing the Lexicographic Maxmin Bargaining Solution

Ashish Goel and Anilesh K. Krishnaswamy

{ashishg, anilesh}@stanford.edu

Stanford University

Abstract. There has been much work on exhibiting mechanisms that implement various bargaining solutions, in particular the Kalai-Smorodinsky solution [24] and the Nash Bargaining solution. Another well-known and axiomatically well-studied solution is the lexicographic maxmin solution. However, there is no mechanism known for its implementation. To fill this gap, we construct a mechanism that implements the lexicographic maxmin solution as the unique subgame perfect equilibrium outcome in the n -player setting. As is standard in the literature on implementation of bargaining solutions, we use the assumption that any player can grab the entire surplus. Our mechanism consists of a binary game tree, with each node corresponding to a subgame where the players are allowed to choose between two outcomes. We characterize novel combinatorial properties of the lexicographic maxmin solution which are crucial to the design of our mechanism.

1 Introduction

Bargaining situations are encountered everywhere. Wage negotiations between employers and trade unions, trade pacts between nations, and environmental negotiations between developed and developing countries are some examples of bargaining that we have seen over the years. Essentially, the problem is one of choosing a feasible alternative, through cooperation, by a group of entities with potentially conflicting preferences. As Kalai [18] points out, bargaining theory

“may be viewed as a theory of consensus, because when it is applied it is often assumed that a final choice can be made if and only if every member of the group supports this choice.”

As bargaining theory inherently deals with the aggregation of peoples’ preferences over a set of outcomes, it closely resembles the theory of social choice. One aspect that distinguishes the bargaining problem from social choice is the existence of a disagreement outcome which may be invoked (often by an outside mediator) when the parties involved in bargaining fail to reach consensus. We provide a formal definition of the bargaining problem in Section 3.

The first question that bargaining attempts to answer is what constitutes a good way of choosing the consensus outcome. Assuming the preferences of the agents are known to the designer, what choice should be made? The earliest answer to this question was an axiomatic one: the well known Nash Bargaining solution [25]. Although originally defined for the 2-player case, the axiomatic characterization of the Nash solution extends seamlessly to the n -player case.

Another prominent bargaining solution is the Kalai-Smorodinsky solution [19], which is obtained by replacing Nash’s independence condition by an axiom of monotonicity. The axiomatic characterization of this solution holds up only when there are two players, and fails to generalize to the case with n players. A straightforward extension of the 2-player Kalai-Smorodinsky solution to the n -player case, one which follows the form of the 2-player solution, is not guaranteed to be Pareto optimal. Henceforth, we refer to this extension as the (n -player) Kalai-Smorodinsky solution.

In this paper, we are primarily interested in the lexicographic maxmin solution (see Section 3.1 for a formal definition), as proposed by Sen [34] or Rawls [29]. It is obtained by a repeated application of the maxmin criterion: first, selecting feasible outcomes that maximize the utility of the worst-off player, then, among these outcomes, selecting those that maximize the utility of the

next worst-off player, and so on. The utility gains are measured with respect to the disagreement point. The lexicographic maxmin solution can also be characterized by a principled extension of the Kalai-Smorodinsky axioms to the n -player case [16]. Other characterizations of the lexicographic maxmin solution are also known [8, 10].

The lexicographic maxmin solution has also had a long history outside of the literature on bargaining. It corresponds directly to the notion of maxmin fairness [1, 17] which has been extensively studied in network routing and bandwidth allocation problems [13, 20, 21]. It has also been studied in relation to the resource allocation [3, 12, 28] and cake-cutting [9] problems. Allowing for randomization, these settings are special cases of the bargaining problem we study in this paper, as discussed in Section 3.

Besides defining good solutions, the other question which bargaining theory deals with is that of implementation – designing a mechanism which, with no knowledge of the agents’ preferences, still leads them to the desired solution. However, the players are assumed to have complete information of the preferences of other players. In other words, we need to construct a game for non-cooperative agents which results in the right solution in equilibrium. While many solution concepts could arguably be used in this scenario, the standard approach is to look at subgame perfect equilibria of repeated games. Bargaining inherently involves repeated interactions between players, of offers and counter offers, which makes the subgame perfect equilibrium a logical and natural choice for the solution concept to be employed [14]. Henceforth, we use the term *implementation* to refer to the problem of designing a mechanism for which the desired bargaining solution is the unique subgame perfect outcome, with each player having complete information.

The implementation of most of the aforementioned bargaining solutions has been studied. The 2-player Nash solution can be approximately implemented¹ as the unique subgame perfect equilibrium in a game of alternating offers [2]. Of particular interest to us is the approach of Moulin [24] in implementing the Kalai-Smorodinsky solution. The same approach has been generalized to other solutions such as Nash (and its n -player extension), and relative utilitarian solutions [23, 33]. In fact, this approach works for any bargaining solution that is the maximizer of some convex function of the players’ utilities. For instance, the Nash solution is equivalent to maximizing the product of utilities, the Kalai-Smorodinsky solution is equivalent to maximizing the minimum among the players’ utilities, etc. In Section 3.2, we discuss the techniques from the above approach.

To the best of our knowledge, there is no known (subgame perfect) implementation of the lexicographic maxmin solution. There is some work on a repeated arbitration procedure [4] which leads to the lexicographic maxmin solution in equilibrium. This procedure falls short of an *implementation* in that it relies on the designer’s having knowledge of the players’ utility functions – As pointed out by Bossert and Tan [4],

“The assumption of utility information being available to the decision maker clearly is quite strong. . . . For instance, it would be useful to consider an environment in which the players know each other’s utilities, but the arbitrator does not know the players’ utility functions. ”

We tackle exactly this problem, and by utilizing novel combinatorial properties of the lexicographic maxmin solution, we provide a mechanism for its implementation. We provide a quick overview of our results and technical contributions in Section 2.

While our mechanism is technically intricate, there are three aspects of it that are interesting from a practical perspective.

¹ with time discounted utilities, in the limit as the discount factor goes to 0.

1. Our subroutine mechanism (the Knockout mechanism, Section 4.2) is simpler, and interesting in its own right. When the players are given a choice between the lexicographic maxmin solution and any other outcome, the Knockout mechanism implements the former.
2. Our mechanism offers insight into the nature of offers and counter offers required to arrive at the desired solution.
3. The gameplay terminates in $\log n$ steps (n is the number of agents) in equilibrium, i.e., when well-meaning and well-informed agents propose the right solution.

Our mechanism used the standard assumption (Assumption 1) that the space of outcomes is such that in any player's best outcome, all the surplus goes to her, and every one else gets no utility. This assumption is commonplace in the literature on implementation of bargaining solutions. For example, it is employed in the work by Moulin [24], and the various extensions of his approach [23, 33]. The bargaining problem without the above assumption is also interesting. Not much work has been done in this setting, besides the implementation of the Nash Bargaining solution [15]. We leave as open the problem of implementing any of the various bargaining solutions, including the lexicographic maxmin solution, absent the assumption that a single player can grab the entire surplus, or proving that such implementations are impossible.

2 A summary of results and techniques

The problem of interest in this paper can be stated as follows: Given n players, and a convex space of outcomes, can we design a mechanism, i.e., a repeated game with complete information, for which the lexicographic maxmin solution (in reference to a disagreement point) is the unique subgame perfect outcome? Our main result is an affirmative answer to this question.

As mentioned earlier, solving for a subgame perfect equilibrium in this case is necessitated by the fact that bargaining inherently involves offers and counter offers, and thereby a repeated game. Because the designer has no knowledge of the players' preferences, the typical building blocks of implementation mechanisms are proposals and threats. Nash equilibrium is an inadequate solution concept for it opens up the possibility of non-credible threats by players, leading to many meaningless equilibria [32]. On the other hand, implementation in dominant strategies is difficult for the bargaining problem because we have neither the possibility of side payments (as in VCG mechanisms) nor the explicit goal of divvying up a finite set of resources as in cake-cutting. The cake-cutting problem is a special case of our setting, with the disagreement point corresponding to no one getting any part of the cake. We must note here that for the cake-cutting problem with piecewise uniform utilities (where the utility of each agent for a unit portion of the cake is either a global constant or zero), it can be shown that the truthful mechanism of Chen et al. [9] implements the lexicographic maxmin solution, even though they do not claim it explicitly. However, for piecewise linear utilities, there is no truthful mechanism that implements the lexicographic maxmin solution, to the best of our knowledge. We also note that our mechanism is not coalition-proof. Similar to the above result of Chen et al. [9], the strategy-proofness of the lexicographic maxmin solution has been established in a variety of domains with dichotomous preferences [3, 12, 28]. There is also very interesting work [22] generalizing, and so unifying, all of the above results.

We design our mechanism in two stages. First, we will consider a simpler problem (see Section 4) - given n players who are offered a choice between the lexicographic maxmin solution $\mathbf{u}^*$ (with the same disagreement point as before), and some other outcome $\mathbf{u}$, is there a mechanism that implements $\mathbf{u}^*$?

The mechanism from the above problem will be used as a subroutine in constructing our full mechanism. We describe this in Section 5. In short, we run a binary tree of games, where each node corresponds to a mechanism where the players to choose between just two outcomes.

This outer mechanism is inspired by the mechanism of Howard [15] which implements the Nash Bargaining solution, although the idea of using a binary tree is novel. A complete description of our game form requires a total number of rounds of the order of $n^2 \log n$ overall. Modeling our outer mechanism analogous to that of the mechanism of Howard [15] (that implements the Nash solution) would lead to a description with n^3 rounds. In fact, our idea of the outer binary tree can be used to simplify Howard’s mechanism [15] for the implementation of the Nash bargaining solution.

To design our inner subroutine, we need a property that

- distinguishes the lexicographic maxmin solution from any other solution, and
- can be exploited in a such a way that a game form based on it leads to $\mathbf{u}^*$ being necessarily chosen as the only subgame perfect outcome.

Just the property that the lexicographic maxmin solution dominates every other outcome with respect to lexicographical ordering is inadequate for the above purpose (discussed in Section 3.2). Therefore, we identify a novel relation between any two vectors called *disagreement dominance*. For any two vectors $\mathbf{u}, \mathbf{v}$, the disagreement dominance relation between the two is obtained by checking the lexicographical dominance relation between the “disagreement projection” of $\mathbf{u}$ onto $\mathbf{v}$ and vice versa; where the disagreement projection of $\mathbf{u}$ onto $\mathbf{v}$ measures the utilities in $\mathbf{u}$ just for those players that prefer $\mathbf{v}$ over $\mathbf{u}$. The disagreement dominance relation has the very interesting property that the lexicographic maxmin solution disagreement dominates every other outcome².

3 The bargaining problem: Model and Definitions

We now describe the model of the bargaining problem, closely following Moulin [24]). We are given a finite set of alternatives denoted by A^3 . These alternatives abstractly correspond to all the possible discrete outcomes. To make the overall space of outcomes convex, all lotteries over these discrete outcomes are added as possible outcomes. Let $P(A)$ be the set of probability distributions over A :

$$P(A) = \{l \in \mathbb{R}^A \mid l_a \geq 0 \forall a \in A, \sum_{a \in A} l_a = 1\}.$$

In other words, $P(A)$ is the set of all lotteries over A . An element l of $P(A)$ is attained by a lottery that randomly selects outcome a with probability l_a .

The set of agents is denoted by $\mathcal{N} = \{1, 2, \dots, n\}$. Each agent $i \in \mathcal{N}$ has Von Neumann-Morgenstern preferences over $P(A)$, which can be represented via a vector $u^i \in \mathbb{R}^A$, and her utility for any lottery l is given by:

$$U_i(l) = \sum_{a \in A} u_a^i l_a.$$

An agent i prefers a lottery l over l' if and only if $U_i(l) > U_i(l')$. The agents are also assumed to be risk-neutral.

We are also given an exogenous lottery s^* , called the status quo outcome or disagreement point. This outcome represents the final outcome in the case where cooperation between agents breaks down, and they fail to reach a consensus. The designer, or an outside mediator has power to enforce this outcome in such a situation. We now formally define what is meant by a bargaining solution.

² To the curious reader: disagreement dominance does not always agree with lexicographic ordering (see the example at the end of Section 4.1).

³ This can be easily generalized to the case of infinite sets.

Definition 1 (Bargaining Solution [11]). *Given the set of alternatives A , the set of agents $\mathcal{N}$, and a status quo point s^* , a bargaining solution is any function that maps any given utility profile $\mathbf{u} = (u^i)_{i \in \mathcal{N}}$ to a lottery $S(\mathbf{u}) \in P(A)$, such that*

- $S(\mathbf{u})$ is individually rational, i.e.,

$$\forall i \in \mathcal{N} : U_i(s^*) \leq U_i(S(\mathbf{u})),$$

- $S(\mathbf{u})$ is invariant under linear transformations of $\mathbf{u}$, i.e., for any scalars $\alpha_i > 0$, and β_i

$$v^i = \alpha_i u^i + \beta_i \mathbb{1}, \forall i \in \mathcal{N} \implies S(\mathbf{v}) = S(\mathbf{u}),$$

where $\mathbb{1} \in \mathbb{R}^A$ is a vector with all components equal to 1.

For any utility profile $\mathbf{u}$, denote by $\mathbf{I}(\mathbf{u})$ the set of all individually rational lotteries at $\mathbf{u}$:

$$\mathbf{I}(\mathbf{u}) = \{l \in P(A) \mid U_i(s^*) \leq U_i(l) \forall i \in \mathcal{N}\}. \quad (1)$$

We assume that $\mathbf{I}(\mathbf{u})$ is non-empty, and for every $\mathbf{u}$, limit the set of feasible outcomes to $\mathbf{I}(\mathbf{u})$.

We only allow those utility profiles $\mathbf{u}$ which satisfy the following assumption:

Assumption 1 *For any agent i , all her favorite outcomes $l_i^* \in \arg \max_{l \in \mathbf{I}(\mathbf{u})} U_i(l)$ are such that*

$$U_j(l_i^*) = U_j(s^*), \quad \forall j \neq i.$$

This assumption has been indispensably used in the literature on implementation of bargaining solutions [24, 33]. It usually holds for bargaining over private resources such as in fair division and other problems. We now look at some specific problem instances where the above assumption holds.

Some motivating examples:

- (A) **Cake-cutting [5, 9, 30]:** Consider a heterogeneous cake given by the interval $[0, 1]$. Each player i has a value density function $v_i : [0, 1] \rightarrow [0, \infty)$, and for any piece (a finite union of subintervals) of the cake X , her utility is given by $V_i(X) = \int_X v_i(x) dx$. The disagreement point is the outcome where no player gets any part of the cake. Moreover, if the players are *hungry* [6], i.e., the value density functions $v_i(\cdot)$'s are strictly positive, then Assumption 1 is satisfied – since the favorite outcome for each player is one where she gets the entire cake, and every other player gets nothing.
- (B) **Wireless Relay Networks [7, 27, 35]:** Consider a wireless network with many agents communicating with their destinations via a single wireless relay. The relay has a total available power P , and this has to be divided among the agents. Denote the power allocated to agent i by P_i , then its quality-of-service – given by the effective SNR (Signal to Noise Ratio) – can be described as

$$V_i(P_i) = \frac{a_i P_i}{b_i P_i + 1},$$

where a_i, b_i are agent-specific constants. We omit a discussion of the roots of the above expression (we point the curious reader to Cao et al. [7] for details).

The disagreement point is given by an allocation of zero power to each agent. The function $V_i(P_i)$ is strictly monotone in P_i , whereby the best outcome for each agent is where she gets allocated all the power (i.e. $P_i = P$) – thereby satisfying Assumption 1.

Although we described our model starting with a finite (material) outcome space (and then taking its convex hull via lotteries), our results can be easily extended to the case where we start with an infinite set of outcomes. In this case, the space of utility outcomes (see Definition 5) could be any general convex and compact space (as a subset of $\mathbb{R}^n$). Given a convex and compact space of utility outcomes, Assumption 1 is also implied indirectly by certain restrictive but natural assumptions – namely strict convexity and free disposal of utilities [23].

3.1 Lexicographic maxmin solution

To satisfy the axiom of scale-invariance that every bargaining solution must satisfy (see Definition 1), we must use relative utility gains in our definitions. Instead, for the sake of convenience, and without loss of generality, let us assume the standard normalization of VNM utilities:

$$\forall i \in \mathcal{N}, U_i(s^*) = 0, \text{ and } \max_{l \in \mathbf{I}(\mathbf{u})} U_i(l) = 1. \quad (2)$$

We now define what lexicographical ordering means in general. Given any vector $\mathbf{p} \in \mathbb{R}^n$, denote by $p_{(1)} \geq p_{(2)} \geq \dots \geq p_{(n)}$ its components arranged in some non-increasing order. We say $\mathbf{p} \succ_L \mathbf{q}$ if and only if, for some j , $p_{(j)} > q_{(j)}$, and for $\forall i > j$, we have $p_{(i)} \geq q_{(i)}$. And $\mathbf{p} \sim_L \mathbf{q}$ if and only if $\forall i$, we have $p_{(i)} = q_{(i)}$. We say $\mathbf{p} \succeq_L \mathbf{q}$ if and only if $\mathbf{p} \succ_L \mathbf{q}$ or $\mathbf{p} \sim_L \mathbf{q}$.

Definition 2. *The lexicographic maxmin solution is a lottery l^* such that $\mathbf{u}(l^*) \succeq_L \mathbf{u}(l)$ for all $l \in p(A)$.*

The existence and uniqueness of such a solution is a consequence of the fact that the space $D = \{x \in \mathbb{R}^n : \exists l \in P(A), \text{ such that } x = [U_1(l), U_2(l), \dots, U_n(l)]\}$ is convex (since $P(A)$ is convex). We state the following lemma.

Lemma 3. *There exists a unique $\mathbf{u}^* \in D$ such that $\mathbf{u}^* \succ_L \mathbf{u}$ for any $\mathbf{u} \neq \mathbf{u}^*$.*

It is easy to see that the above solution is guaranteed to be Pareto optimal. The Kalai-Smorodinsky solution in contrast is not guaranteed to be Pareto optimal.

We now illustrate the lexicographic maxmin solution with an example.

Example 4. Consider a three person bargaining problem whose disagreement point s^* is such that the utilities of the agents for it is $(0, 0, 0)$. The set of alternatives is given by s^* and five other points a, b, c, d, e for which the utilities are $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, $(0.6, 0.6, 0.7)$, $(0.6, 0.6, 0)$ respectively. Clearly the lexicographic maxmin solution is $(0.6, 0.6, 0.7)$. The Kalai-Smorodinsky solution (see Appendix A for a definition) here is one that gives utilities $(0.6, 0.6, 0.6)$ to the agents, which is Pareto dominated.

The lexicographic maxmin solution is uniquely characterized by a set of axioms generalizing those of Kalai-Smorodinsky [16]. Just for the case of two players, however, the two solutions coincide. For more than two players, they are different as seen in Example 4.

From here on forward, we will just use the convex space D of utility outcomes, as opposed to dealing with the space of lotteries. Below, we summarize the relevant definitions in terms of D .

Definition 5. *The space of utility outcomes is given by*

$$D = \{x \in \mathbb{R}^n : \exists l \in P(A), \text{ such that } x = [U_1(l), U_2(l), \dots, U_n(l)]\}.$$

Definition 6 (Lexicographic maxmin solution). *The lexicographic maxmin solution is the unique point $\mathbf{u}^* \in D$ such that $\mathbf{u}^* \succ_L \mathbf{u}$ for all $\mathbf{u} \neq \mathbf{u}^*$.*

The following restatement of Assumption 1 in terms of the quantity D will be more convenient for us in the following sections.

Assumption 2 *For any agent i , there exists a unique ideal point $\mathbf{s}^i \in D$, and moreover it satisfies*

$$\begin{aligned} s_i^i &= 1, \\ s_j^i &= 0, \quad \forall j \neq i. \end{aligned}$$

The problem of implementing the lexicographic solution can be stated as follows:

Definition 7 (Implementation). *Given n players with complete information, collectively the set $\mathcal{N}$, and the utility space D , a mechanism $M(D, \mathcal{N}, \mathbf{s}^*)$ is said to implement $\mathbf{u}^*$ if and only if $\mathbf{u}^*$ is the unique subgame perfect outcome of $M(D, \mathcal{N}, \mathbf{s}^*)$.*

In other words, a mechanism $M(D, \mathcal{N}, \mathbf{s}^*)$ is said to implement $\mathbf{u}^*$ when $M(D, \mathcal{N}, \mathbf{s}^*)$ always admits a subgame perfect equilibrium, and $\mathbf{u}^*$ is the outcome of any such equilibrium.

3.2 Comparison with previous work

The implementation of the Kalai-Smorodinsky solution is possible via a mechanism that has a single round of simultaneous bidding, followed by a series of accept/reject decisions by the players [24]. The framework of Moulin's mechanism [24] has been extended to many other bargaining solutions [23, 33]. In particular, it is known that any bargaining solution that is equivalent to maximizing a convex function of the players' utilities can be implemented as a subgame perfect equilibrium by a mechanism that does a pass over the entire list of players twice [23]. Such a mechanism works roughly as follows: In the first pass, each agent gets to propose an outcome, and a corresponding "bid" (a vector of promised utilities to all the players). These bids are arranged in descending order according to the desired convex function, and the first player is designated the winner. Each player then gets to accept the winner's proposed outcome, or reject it and implement a dictatorial threat [23]. These games are much simpler than the one we come up with in this paper: the full description of these game involves a total number of rounds of the order of n .

Because the lexicographic maxmin solution does not correspond to maximizing a convex function of utilities, the above approach fails to work. A natural question is whether using lexicographic ordering to order the bids in the above approach solves our problem. In the remainder of this section, we will describe why this is not the case. We are therefore forced to look for more complicated game forms. A similar hurdle was faced by Howard [15] in trying to design a mechanism that implements the Nash solution without the standard assumption that every player grabs all the surplus in her favorite outcome, while the others get nothing.

Without going into the details of Moulin's mechanism (that implements the Kalai-Smorodinsky solution) [24] and other related ones, we would like to illustrate, by way of a concrete example, why direct extensions of these approaches do not work for the lexicographic maxmin solution. This will shed light on the complexity of our problem.

A direct extension of Moulin's mechanism [24] to our problem would roughly be the following: Each player i declares a bid $\mathbf{p}^i \in [0, 1]^n$, and these are ordered in accordance with lexicographic ordering. A player whose bid, say $\mathbf{p}$, lexicographically dominates every other bid is declared the winner, and gets to propose an outcome $\mathbf{L}$ in D . Each other player i can either

1. accept $\mathbf{L}$, or,

2. reject to impose the (lottery) outcome $p_i s^i + (1 - p_i) s^*$, which gives her a utility of p_i , as the final outcome.

If all players accept $\mathbf{L}$ then it becomes the final outcome. Note that when some player rejects, the utility for the winner, and indeed all others is 0 by Assumption 2.

While the lexicographic maxmin is a subgame perfect equilibrium of this game, it's not the only equilibrium. The fact that a player can lie about her own utility level in her bid (since it is not checked by anyone's threat) leads to many meaningless equilibria. Let's look at an example to see why this is the case.

Example 8. Recall the outcome space in Example 4: given 3 players, let the space D be given by all points in the convex hull of the points

$$(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1), (0.6, 0.6, 0.7), (0.6, 0.6, 0).$$

The disagreement point, as usual, is $(0, 0, 0)$.

The lexicographic maxmin solution here is $(0.6, 0.6, 0.7)$.

In the above example, say player 2 bids $(0.6, 1, 0.65)$. Player 1 has no reason to challenge this bid, as she is guaranteed a utility of 0.6, and there is no point where she can get a higher utility while ensuring that player 2 gets at least 0.6. If player 3 bids above (lexicographically speaking) player 2's bid, and then proposes at outcome where she gets a utility more than 0.55, her proposed outcome is bound to get rejected by someone. We now explain why this happens.

Even if player 3 bids $\mathbf{p}$ such that $p_3 = 1^4$, then the other components in her bid have to together lexicographically dominate $(0.6, 0.65)$. There is no outcome that ensures these levels of utility for the other players. Therefore, this leads us to an equilibrium where player 2 wins the bid and can possibly propose the outcome $(0.6, 0.6, 0.65)$ which is accepted by others.

Note that the above analysis holds even when the bids are done in turn. If the bids are simultaneous as in the game of Moulin's mechanism [24], simpler examples of bad equilibria are possible.

As a result, current techniques are inadequate to design a mechanism that implements the lexicographic maxmin.

4 Implementing the lexicographic maxmin solution

Before going into the general problem of implementing the lexicographic maxmin solution, we will look at a simpler problem: Given $\mathbf{u}^*$ and another outcome $\mathbf{u}$ (with the same disagreement point as before), can we design a mechanism which implements, for any given number of players, the solution $\mathbf{u}^*$? We answer this question by designing the Knockout mechanism. This mechanism will be an important subroutine in implementing our full mechanism for the implementation of the lexicographic maxmin solution.

Let us first characterize a novel combinatorial property of the lexicographic maxmin that will be of immense use to us.

4.1 Disagreement dominance

As mentioned before, given two utility outcomes, say $\mathbf{u}$ and $\mathbf{v}$, we use the *disagreement-dominance* relation as a way of comparing the two.

⁴ There's no reason to bid less than 1 here, as doing so can only hurt the lexicographic ordering of her whole bid.

For any pair of utility outcomes, $\mathbf{u}$ and $\mathbf{v}$, we define another vector $\pi(\mathbf{u}, \mathbf{v})$ of the same size, which we call the *disagreement* projection of $\mathbf{u}$ onto $\mathbf{v}$.

$$(\pi(\mathbf{u}, \mathbf{v}))_i = \begin{cases} u_i, & \text{if } u_i < v_i, \\ 1, & \text{otherwise.} \end{cases}$$

We define the disagreement-dominance ($\mathcal{D}$ -dominance) relation as follows:

Definition 9 ($\mathcal{D}$ -dominance). $\mathbf{u}$ $\mathcal{D}$ -dominates $\mathbf{v}$ if and only if $\pi(\mathbf{u}, \mathbf{v}) \succeq_L \pi(\mathbf{v}, \mathbf{u})$. Moreover, if $\pi(\mathbf{u}, \mathbf{v}) \succ_L \pi(\mathbf{v}, \mathbf{u})$, then $\mathbf{u}$ strictly $\mathcal{D}$ -dominates $\mathbf{v}$.

The above relation is essentially comparing, in a leximin fashion, the utilities in $\mathbf{u}$ of those agents that prefer $\mathbf{v}$ over $\mathbf{u}$, and the utilities in $\mathbf{v}$ of those agents that prefer $\mathbf{u}$ over $\mathbf{v}$.

The following lemma, which characterizes the relationship between the lexicographical ordering and the $\mathcal{D}$ -dominance relation, is a major technical contribution of this paper.

Lemma 10. *The lexicographical maxmin solution $\mathbf{u}^*$ strictly $\mathcal{D}$ -dominates any other feasible utility outcome $\mathbf{u}$, where $\mathbf{u} \neq \mathbf{u}^*$.*

Proof. Assume to the contrary that $\mathbf{u}$ $\mathcal{D}$ -dominates $\mathbf{u}^*$, i.e., $\pi(\mathbf{u}, \mathbf{u}^*) \succeq_L \pi(\mathbf{u}^*, \mathbf{u})$. We will show shortly that this assumption leads to the absurdity that $\frac{1}{2}(\mathbf{u} + \mathbf{u}^*) \succ_L \mathbf{u}^*$.

Divide the players $\mathcal{N} = \{1, 2, \dots, n\}$ into three sets X, Y, Z as follows:

$$\begin{aligned} X &= \{i \in \mathcal{N} : u_i > u_i^*\}, \\ Y &= \{i \in \mathcal{N} : u_i^* > u_i\}, \\ Z &= \{i \in \mathcal{N} : u_i = u_i^*\}. \end{aligned}$$

Without loss of generality we can assume that both X and Y are non-empty. Of course, both can't be empty simultaneously, for then $\mathbf{u} = \mathbf{u}^*$. If only Y is empty, then $\mathbf{u} \succ_L \mathbf{u}^*$, which is an absurdity. If only X is empty, then $\mathbf{u}^*$ strictly $\mathcal{D}$ -dominates $\mathbf{u}$ trivially.

Let $u_{(1)}^* \leq u_{(2)}^* \leq \dots \leq u_{(n)}^*$ denote the components of $\mathbf{u}^*$ arranged in non-decreasing order, and let $\sigma(k)$ be the agent corresponding to $u_{(k)}^*$.

Let j be the smallest value such that $\sigma(j) \in X \cup Y$, and let $u_{\sigma(j)}^* = \alpha$. Since ties can be broken arbitrarily in the above ordering of the components, assume without loss of generality that

$$\forall j' \geq j, u_{\sigma(j')}^* > \alpha \text{ whenever } j' \in Z. \quad (3)$$

By definition, for $j' < j$, we have $u_{\sigma(j')}^* = u_{\sigma(j)}$. In addition, if we show that for $j' \geq j$, $\frac{1}{2}(u_{\sigma(j')}^* + u_{\sigma(j)}) > \alpha$, then $\frac{1}{2}(\mathbf{u}^* + \mathbf{u}) \succ_L \mathbf{u}^*$ which is a contradiction according to Lemma 3.

Let $\gamma = \min_{i \in X} u_i^*$, and $\delta = \min_{i \in Y} u_i$. Since $\alpha = \min_{i \in X \cup Y} u_i^*$, we have $\gamma \geq \alpha$. Since $\pi(\mathbf{u}, \mathbf{u}^*) \succeq_L \pi(\mathbf{u}^*, \mathbf{u})$, we have $\min_{i \in Y} u_i \geq \min_{i \in X} u_i^*$, i.e., $\delta \geq \gamma$. Combining the above two arguments, we have $\delta \geq \alpha$.

In other words, we have

$$\begin{aligned} \forall i \in X, u_i^* &\geq \alpha, \\ \forall i \in Y, u_i &\geq \alpha. \end{aligned}$$

Now consider any $j' \geq j$. There are three possible cases here:

(A) $\sigma(j') \in X$: In this case, $u_{\sigma(j')} > u_{\sigma(j')}^*$, and $u_{\sigma(j')}^* \geq \alpha$. Therefore $\frac{1}{2}(u_{\sigma(j')}^* + u_{\sigma(j)}) > \alpha$.

(B) $\sigma(j') \in Y$: In this case, $u_{\sigma(j')}^* > u_{\sigma(j')}$, and $u_{\sigma(j')} \geq \alpha$. Therefore $\frac{1}{2}(u_{\sigma(j')}^* + u_{\sigma(j')}) > \alpha$.

(C) $\sigma(j') \in Z$: By our assumption to simplify accounting (equation 3 above), we already have $u_{\sigma(j')}^* > \alpha$, which automatically means $\frac{1}{2}(u_{\sigma(j')}^* + u_{\sigma(j')}) > \alpha$, since $u_{\sigma(j')}^* = u_{\sigma(j')}$. $\square$

We note here that lexicographic dominance does not imply $\mathcal{D}$ -dominance. For example, take $\mathbf{u} = (0.1, 0.2, 0.3, 0.4, 0.5)$ and $\mathbf{v} = (0.3, 0.3, 0.4, 0.1, 0.2)$. Here $\mathbf{u} \succ_L \mathbf{v}$ but $\mathbf{v}$ strictly $\mathcal{D}$ -dominates $\mathbf{u}$.

4.2 The Knockout mechanism

Armed with Lemma 10, we first look for a mechanism where the players are given a choice between $\mathbf{u}^*$ and some other outcome $\mathbf{u}$, with the same disagreement point as before, i.e., $\mathbf{s}^*$. Before we describe this game, we will need to define the mechanism $D(i, \mathbf{p})$, agent i 's dictatorial mechanism [24], which comes into force when i rejects a proposal in the overall mechanism.

Definition 11 ($D(i, \mathbf{p})$). *Agent i proposes an outcome $\mathbf{L}^i$ for unanimous approval by other agents. If at least one agent rejects $\mathbf{L}^i$, the referee enforces the status quo $\mathbf{s}^*$. If everybody accepts $\mathbf{L}^i$, the referee enforces the lottery $p_i \mathbf{L}^i + (1 - p_i) \mathbf{s}^*$.*

A straightforward consequence of Assumption 2 is that the outcome $\mathbf{s}^i$, i.e., i 's ideal outcome, is a subgame perfect proposal of i in $D(i, \mathbf{p})$. One might wonder why we could not just have a provision to directly implement $p_i \mathbf{L}^i + (1 - p_i) \mathbf{s}^*$ based on player i 's choice of outcome. The point is that $\mathbf{L}^i$ could be Pareto-dominated by $\mathbf{s}^*$, and we need a device to ensure individual rationality, with respect to $\mathbf{s}^*$ being normalized to $(0, 0, \dots, 0)$, when some agent rejects a proposed lottery.

We now define the Knockout mechanism $K(X, Y, \mathcal{N})$ on a pair of outcomes X, Y , and a set of players $\mathcal{N}$ of size n . The game will be defined recursively.

Knockout Mechanism $K(X, Y, \mathcal{N})$ ⁵

Round 1:

- (A) Each players i in turn chooses her preferred outcome $\mathbf{L}^i$ to be either X or Y .
- (B) Each player i bids a vector $\mathbf{p}^i \in [0, 1]^n$ in turn⁶.
- (C) The players are re-ordered, with all ties broken lexicographically, such that

$$\mathbf{p}^1 \succeq_L \mathbf{p}^i, \forall i \neq 1, \text{ and } p_2^1 \leq p_3^1 \leq \dots p_n^1,$$

where p_j^i is the component corresponding to player j in the bid $\mathbf{p}^i$ made by player i .

Round j ($j \in \{2, 3, \dots, n\}$): Player j has to make a choice between two options which vary based on the value of her value in player 1's winning bid, i.e., p_j^1 :

- (A) If $p_j^1 < 1$, then player k can either accept $\mathbf{L}^1$, or reject it. In the latter case, i.e., a rejection by player j , the game immediately reduces to $D(j, \mathbf{p}^1)$.
- (B) If $p_j^1 = 1$, the player k can either accept $\mathbf{L}^1$, or reject it. In the latter case, i.e., a rejection by player j , the game immediately reduces to $K(X, Y, \mathcal{N} - \{1\})$ (with player 1 ejected).

If player j accepts, then we move on to the next player $j + 1$, and so on, to player n in round n . If all players accept and we reach the end of round n , then $\mathbf{L}^1$ is the final outcome.

⁵ Recall that X and Y , like all feasible solutions, satisfy individual rationality with respect to $\mathbf{s}^*$ (see equation 1), i.e., no player strictly prefers $\mathbf{s}^*$ over either X or Y .

⁶ Assuming i prefers X over Y , her bid $\mathbf{p}^i$ here can be thought of as the levels of utility she promises to players who prefer Y over X . Also, as mentioned in the proof of Lemma 13, simultaneous bidding here leads to the existence of bad equilibria.

4.3 Analyzing the Knockout mechanism

The above mechanism implements the lexicographic maxmin solution when it is pitted against any other outcome. More formally, when one of the outcomes is $\mathbf{u}^*$, the above game admits a subgame perfect equilibrium, and that any subgame perfect equilibrium of this game results in a payoff of u^* to the agents. In fact, we will prove a much stronger property of the Knockout mechanism in terms of the $\mathcal{D}$ -dominance relation.

Theorem 12. *If X strictly $\mathcal{D}$ -dominates Y , then $K(X, Y, \mathcal{N})$ implements X .*

We first prove the above theorem for the case of $n = 2$, which will serve as the base case for an inductive proof of the theorem in all generality. Given two players $\{1, 2\}$, and two outcomes X, Y ⁷, let $X^{\{1,2\}} = (u_1, u_2)$ and $Y^{\{1,2\}} = (v_1, v_2)$.

Lemma 13. *If $X^{\{1,2\}}$ strictly $\mathcal{D}$ -dominates $Y^{\{1,2\}}$, then $K(X, Y, \{1, 2\})$ implements X .*

Proof. Assume that ties are broken in favor of player 2. In Round 1, let's say player 1 chooses $\mathbf{L}_1 = A$ and player 2 chooses $\mathbf{L}_2 = B$. The most pertinent case is when 1 strictly prefers A , and 2 strictly prefers B . All other cases follow analogously.

Case (1) A strictly $\mathcal{D}$ -dominates B : Now player 1 can bid $\mathbf{p}^1 = \pi(A^{\{1,2\}}, B^{\{1,2\}}) = (1, A_2)$. To outbid player 1, player 2 has to bid $\mathbf{p}^2 = (t, 1)$ where $t \geq A_2$. Since $A^{\{1,2\}}$ strictly $\mathcal{D}$ -dominates $B^{\{1,2\}}$, we know that $\pi(A^{\{1,2\}}, B^{\{1,2\}}) \succ_L \pi(B^{\{1,2\}}, A^{\{1,2\}}) \iff (1, A_2) \succ_L (B_1, 1) \iff A_2 > B_1$. Therefore, player 2's bid must be such that $t > B_1$, in which case, player 1 is better off by rejecting, and player 2 gets 0 utility in this case. Consequently, A is the outcome in this case.

Case (2) B $\mathcal{D}$ -dominates A : Now player 1 has to specify some bid $\mathbf{p}^1 = (1, s)$ such that $s \leq A_2$, for otherwise, player 2 will reject and player 1 will get a utility of 0. Recall that ties are broken in favor of player 2. So player 2 can now outbid player 1, by specifying a bid $\mathbf{p}^2 = (t, 1)$ such that $B_1 \geq t \geq A_2 \geq s$. This is because we know that in this case, B strictly $\mathcal{D}$ -dominates $A \implies B_1 \geq A_2$. And in this case, B is the outcome, since player 1 will accept.

Putting these together, we see that in any subgame perfect equilibrium of the game from Round 1 on, if A strictly $\mathcal{D}$ -dominates B , then A is the outcome (and if not, B is the outcome).

As a result, since X strictly $\mathcal{D}$ -dominates Y , we know that at least one of the players strictly prefers X over Y . Consequently, X is the only possible subgame perfect outcome.

To show that a subgame perfect equilibrium exists, we show that a finite backwards reduction is possible. Let's say we are given the choice of $\mathbf{L}^1$ and $\mathbf{L}^2$ in Round 1(A). While infinite choices are available for the bids in the round 1(B), each player would not make a bid that gets rejected in favor of the dictatorial mechanism⁸. In effect, player 1, for example, is in effect making a choice between his own choice $\mathbf{L}_1$, player 2's choice $\mathbf{L}_2$, and the outcome of the subgame $L(X, Y, \{2\})$ (which by definition is equal to player 2's preferred outcome between X and Y). As the number of choices of outcome in each step is effectively finite, the game can be reduced backwards, thereby guaranteeing the existence of a subgame perfect equilibrium. $\square$

In the case when X $\mathcal{D}$ -dominates Y and Y $\mathcal{D}$ -dominates X , the outcome depends on how ties are broken between agents in the bidding phase of the game, and which outcome the winning bidder prefers.

We now extend the above claim via an inductive argument to prove Theorem 12.

⁷ Even though we have only 2 players, we let $X, Y \in [0, 1]^n$ to denote outcomes.

⁸ This might not hold with simultaneous bidding in the case of more than two players. The equilibrium bids of the two players could be such that if one lowers his bid the other wins, and a third player rejects regardless of who wins.

Proof of Theorem 12. This proof is essentially similar to that of Lemma 13, with some additional complexities. In what follows, we highlight these differences, and discuss how we can take care of them. As mentioned before, the proof is an inductive one: an induction on the size of $\mathcal{N}$. Let the above statement be true for all sets of agents smaller than $\mathcal{N}$. The base case when $|\mathcal{N}| = 2$ holds by Lemma 13.

Now consider $K(X, Y, \mathcal{N})$. Let A be the set of agents that strictly prefer X , and B be the set of agents that strictly prefer Y , and C be the set of agents that are indifferent. X strictly $\mathcal{D}$ -dominates Y necessitates that A is non-empty. If B is empty, then the argument below holds vacuously. So assume that B is non-empty.

Consider an agent i in A . This agent can choose $\mathbf{L}^i = X$ and bid $\mathbf{p}^i = \pi(X, Y)$. If this bid wins, then any player k in A cannot do better than accepting. Recall that, by way of the labeling in Round 1(C), we go through the players in increasing order of their components in the winner's bid vector. Therefore, only players in A follow k . Also, since $p_j^i = X_j$ for all $j \in B$, any agent in B cannot do better than accepting either. This follows from the fact that $X_j = p_j^i < 1$ for all $j \in B$, and by rejecting, and thereby reducing the game to $D(\mathbf{p}^i, j)$ (recall Round $j(A)$, for any $j \geq 2$, in the definition of the Knockout mechanism), any such agent j can get at most X_j .

Any agent j in B does not benefit by bidding any $\mathbf{p}$ such that $\mathbf{p} \succ_L \pi(Y, X)$, because in this case, there exists a player $k \in A$ such that $p_k > X_k$. Here we have to look at two cases when j wins:

- (A) If $p_k = 1$ then the player k will reject j 's proposal and the game will be reduced to $K(X, Y, \mathcal{N} - \{j\})$. The outcome of this game by the induction assumption is X , since $X_j < Y_j$, and removing j from the set of players still ensures $X^{\mathcal{N}-\{j\}}$ strictly $\mathcal{D}$ -dominates $Y^{\mathcal{N}-\{j\}}$ ⁹.
- (B) If $p_k < 1$, then player k will reject and the game reduces to $D(\mathbf{p}, k)$, the outcome of which has 0 utility for player j .

Thus, in either case, the outcome for player $j \in B$ is no better than X . Hence, j would rather accept i 's proposal of X .

Combining the arguments above, we have that any subgame perfect equilibrium will lead to the outcome X .

The existence of a subgame perfect equilibrium follows from the fact that a backwards reduction is possible, using an argument similar to that for Lemma 13. $\square$

A corollary of the above result, by way of Lemma 10, is that when one of the two allowed outcomes is $\mathbf{u}^*$, the Knockout mechanism implements $\mathbf{u}^*$.

Corollary 14. *For any outcome $\mathbf{u} \neq \mathbf{u}^*$, the mechanism $K(\mathbf{u}^*, \mathbf{u}, \mathcal{N})$ implements $\mathbf{u}^*$.*

5 The outer binary tree mechanism: a recursive system of Knockout games

In this section, we develop a complete mechanism – a *binary game tree system with recursive Knockout games* – that implements the lexicographic maxmin solution. On a sidenote, a modified version of our binary game tree can be used to simplify Howard's mechanism [15] that implements the Nash Bargaining solution.

Before describing our outer binary tree mechanism, we extend the definition of the Knockout mechanism to a game over two subgames, instead of two outcomes. We had defined $K(X, Y, \mathcal{N})$ as the Knockout mechanism over two outcomes X and Y . Similarly given any two bargaining games A and B that each produce some outcome, we define the Knockout mechanism $K(A, B, \mathcal{N})$

⁹ X^S denotes the projection of the n -dimensional X onto the subset of coordinates corresponding to the set of players S .

analogously: by replacing the outcomes X and Y in the game tree of $K(X, Y, \mathcal{N})$ by the pending outcomes of subgames A and B . In other words, instead of bargaining over the outcomes, the players bargain over which of the subgames A or B they prefer to play next. We now make the following simple observation:

Lemma 15. *If α and β are the unique subgame perfect outcomes of games A and B , and α strictly $\mathcal{D}$ -dominates β , then α is the unique subgame perfect outcome of $K(A, B, \mathcal{N})$.*

The above lemma is a simple consequence of the backward induction property (see, e.g. Proposition 99.2 in Osborne and Rubinstein [26]) of a subgame perfect equilibrium. We include a proof in Appendix B for the sake of completeness.

5.1 Defining the outer binary tree mechanism

The outer binary tree mechanism can be defined (see Figure 1) as follows:

- First, each player i in turn chooses an outcome of her choice P_i .
- For the sake of simplicity, assume that n is a power of 2. The game tree^{10,11} is obtained as follows: create a source node $S_{1,2,\dots,n}$, then its child nodes $S_{1,2,\dots,n/2}$ and $S_{n/2+1,\dots,n}$, and so on, till we reach the leaf nodes $S_1, S_2, \dots, S_n$.
- Each leaf node S_i is identified with the outcome P_i .
- Given any node X in the above tree, and its child nodes A and B , the game at X is defined recursively by the Knockout mechanism $K(A, B, \mathcal{N})$.
- The game-play starts at $S_{1,2,\dots,n}$, the game $K(S_{1,2,\dots,n/2}, S_{n/2+1,\dots,n}, \mathcal{N})$ is played, and if¹² and when one of $S_{1,2,\dots,n/2}$ or $S_{n/2+1,\dots,n}$ is the chosen outcome, the game moves to the corresponding node, and so on.

Note that all players are involved at the start of the game-play at every node. The subscripts of the nodes indicate which of the S_i 's are reachable from each of them, and not the players involved. However, some agents might be temporarily not involved, till the game reaches the next node, or a player's dictatorial mechanism (recall Definition 11).

Before analyzing the above mechanism, we make an important note on the intuition behind its working. *Our mechanism, in its entirety, is a recursive system that works like a knockout tournament in reverse. By a repeated application of Lemma 15, doing a backward induction on the game tree above is equivalent to running a standard knockout tournament (à la a tennis grand slam) on the set $\{P_i : i \in \mathcal{N}\}$, where each pairwise game is decided according to the $\mathcal{D}$ -dominance relation between the outcomes involved.*

5.2 Analyzing the binary tree mechanism

Our main result can be stated as follows:

Theorem 16. *The binary tree mechanism above implements the lexicographic maxmin solution.*

Proof. We prove the above theorem in two parts:

(A) the solution $\mathbf{u}^*$ is a subgame perfect outcome, and

¹⁰ The game tree can be modified very easily if n is not a power of 2. Details are deferred to a full version.

¹¹ Also, we are describing not the extensive form of the game tree, but just a scaffold – with Knockout games placed on each of the nodes of the tree. On the whole, we have a system of recursive Knockout games.

¹² The game could also go into a player's dictatorial mechanism (recall Definition 11) and end there.

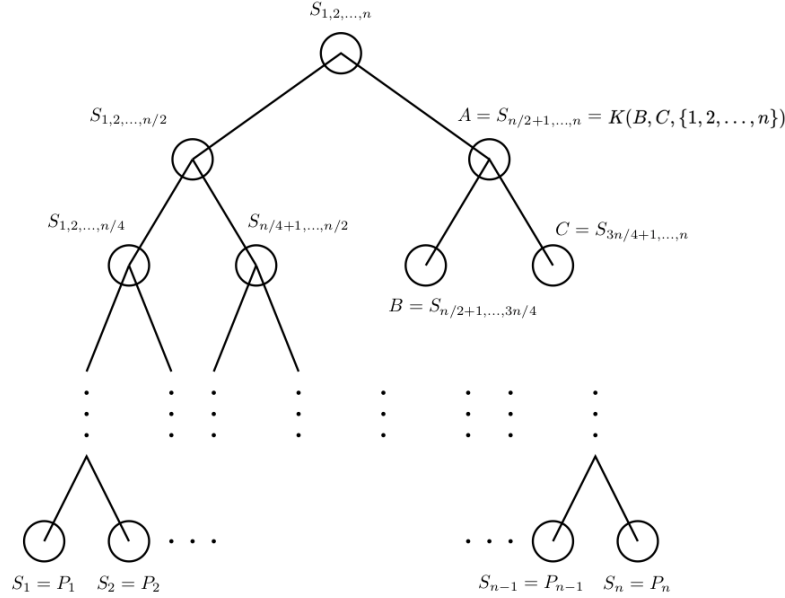


Fig. 1. The outer binary tree mechanism

(B) $\mathbf{u}^*$ is the only possible subgame perfect outcome.

Part (A): Assume all players choose $\mathbf{u}^*$, except player 1 who chooses some other outcome $\mathbf{u}$. As a consequence of Corollary 14, the outcome $\mathbf{u}^*$ is implemented in the knockout game $K(S_1, S_2, \mathcal{N})$ that is played at the parent node of P_1 , where $S_1 = P_1 = \mathbf{u}$ and $S_2 = P_2 = \mathbf{u}^*$. Since the outcome at every other leaf node, besides S_1 and S_2 , is $\mathbf{u}^*$, a repeated application of Lemma 15 tells us that the outcome at $S_{1,2,...,n}$ is $\mathbf{u}^*$.

Part (B): Assume we have a subgame perfect equilibrium where not all players choose $\mathbf{u}^*$ in the first round. In any such equilibrium, any player i is guaranteed a utility of at least u_i^* – for if not, player i can choose $\mathbf{u}^*$, and by a repeated application of Corollary 14 and Lemma 15, we have that $\mathbf{u}^*$ is the outcome implemented in every subgame node that is reachable from P_i : ultimately at $S_{1,2,...,n}$. Since every player i is guaranteed at least u_i^* in a subgame perfect equilibrium, the only possible outcome is $\mathbf{u}^*$ (by Lemma 3). $\square$

6 Discussion

In this paper, we look at a widely studied and well characterized bargaining solution, namely the lexicographic maxmin solution. Although a lot of work has been done on the implementation of other major bargaining solutions, there is no known mechanism for the implementation of the lexicographic maxmin solution. Interestingly, we are able to design such a mechanism by uncovering and utilizing novel combinatorial properties of the lexicographic maxmin solution. In doing so, we also make use of the standard assumption in bargaining implementation theory that the space of outcomes is such that it is possible for each player to grab all the surplus. As we pointed out earlier, an interesting open problem is to implement any of the various bargaining solutions, including the lexicographic maxmin solution, without the above assumption, or to prove that such implementations are impossible – our intuition aligns more with the latter (details deferred to a full version). It would also be interesting to look at the case where there is uncertainty in the knowledge that each player has about the utility valuations of other players. We hope this paper fuels more research in the abovementioned directions.

References

1. Dimitri P Bertsekas, Robert G Gallager, and Pierre Humblet. *Data networks*, volume 2. Prentice-Hall International New Jersey, 1992.
2. Ken Binmore, Ariel Rubinstein, and Asher Wolinsky. The nash bargaining solution in economic modelling. *The RAND Journal of Economics*, pages 176–188, 1986.
3. Anna Bogomolnaia and Hervé Moulin. Random matching under dichotomous preferences. *Econometrica*, 72(1):257–279, 2004.
4. Walter Bossert and Guofu Tan. An arbitration game and the egalitarian bargaining solution. *Social Choice and Welfare*, 12(1):29–41, 1995.
5. Steven J Brams and Alan D Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.
6. Simina Brânzei and Peter Bro Miltersen. A dictatorship theorem for cake cutting. In *IJCAI*, pages 482–488, 2015.
7. Qian Cao, H Vicky Zhao, and Yindi Jing. Power allocation and pricing in multiuser relay networks using stackelberg and bargaining games. *IEEE Transactions on Vehicular Technology*, 61(7):3177–3190, 2012.
8. Chih Chang and Yan-An Hwang. A characterization of the leximin solution of the bargaining problem. *Mathematical Methods of Operations Research*, 49(3):395–400, 1999.
9. Yiling Chen, John K Lai, David C Parkes, and Ariel D Procaccia. Truth, justice, and cake cutting. *Games and Economic Behavior*, 77(1):284–297, 2013.
10. Youngsub Chun. Lexicographic egalitarian solution and uncertainty in the disagreement point. *Mathematical Methods of Operations Research*, 33(4):259–266, 1989.
11. Wulf Gaertner. *A primer in social choice theory: Revised edition*. Oxford University Press, 2009.
12. Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *Nsdi*, volume 11, pages 24–24, 2011.
13. Ashish Goel, Adam Meyerson, and Serge Plotkin. Combining fairness with throughput: Online routing with multiple objectives. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 670–679. ACM, 2000.
14. Maria Jose Herrero and Sanjay Srivastava. Implementation via backward induction. *Journal of Economic Theory*, 56(1):70–88, 1992.
15. JV Howard. A social choice rule and its implementation in perfect equilibrium. *Journal of Economic Theory*, 56(1):142–159, 1992.
16. Haruo Imai. Individual monotonicity and lexicographic maxmin solution. *Econometrica: Journal of the Econometric Society*, pages 389–401, 1983.
17. Jeffrey Jaffe. Bottleneck flow control. *IEEE Transactions on Communications*, 29(7):954–962, 1981.
18. Ehud Kalai. *Solutions to the Bargaining Problem*, pages 77–105. Cambridge University Press, 1985.
19. Ehud Kalai and Meir Smorodinsky. Other solutions to nash’s bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 513–518, 1975.
20. Jon Kleinberg, Yuval Rabani, and Éva Tardos. Fairness in routing and load balancing. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 568–578. IEEE, 1999.
21. Amit Kumar and Jon Kleinberg. Fairness measures for resource allocation. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 75–85. IEEE, 2000.
22. David Kurokawa, Ariel D Procaccia, and Nisarg Shah. Leximin allocations in the real world. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 345–362. ACM, 2015.
23. Eiichi Miyagawa. Subgame-perfect implementation of bargaining solutions. *Games and Economic Behavior*, 41(2):292–308, 2002.
24. Herve Moulin. Implementing the kalai-smorodinsky bargaining solution. *Journal of Economic Theory*, 33(1):32–45, 1984.
25. John F Nash. The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162, 1950.
26. Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994.
27. Hyunggon Park and Mihaela van der Schaar. Bargaining strategies for networked multimedia resource management. *IEEE Transactions on Signal Processing*, 55(7):3496–3511, 2007.
28. David C Parkes, Ariel D Procaccia, and Nisarg Shah. Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. *ACM Transactions on Economics and Computation (TEAC)*, 3(1):3, 2015.
29. John Rawls. *A Theory of Justice: First edition*. Harvard University Press, 1971.
30. Jack Robertson and William Webb. *Cake-cutting algorithms: Be fair if you can*. AK Peters/CRC Press, 1998.
31. Alvin E Roth. An impossibility result concerningn-person bargaining games. *International Journal of Game Theory*, 8(3):129–132, 1979.
32. Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, pages 97–109, 1982.

33. Yusuke Samejima. A note on implementation of bargaining solutions. *Theory and decision*, 59(3):175–191, 2005.
34. Amartya Sen. *Collective Choice and Social Welfare*. Holden Day, San Francisco, 1970.
35. Guopeng Zhang, Hailin Zhang, Liqiang Zhao, Wei Wang, and Li Cong. Fair resource sharing for cooperative relay networks using nash bargaining solutions. *IEEE Communications Letters*, 13(6), 2009.

A The Kalai-Smorodinsky Solution

The Kalai-Smorodinsky (KS for short) solution is defined as follows:

Definition 17 (Kalai-Smorodinsky solution). *Given a utility profile $\mathbf{u}$, the KS solution is given by the lottery $l \in P(A)$ that satisfies:*

$$\forall j \in N, \frac{U_j(l) - U_j(s^*)}{\max_{x \in \mathbf{I}(\mathbf{u})} U_j(x) - U_j(s^*)} = \max_{l' \in I(\mathbf{u})} \min_{i \in N} \frac{U_i(l') - U_i(s^*)}{\max_{x \in \mathbf{I}(\mathbf{u})} U_i(x) - U_i(s^*)}.$$

In other words, the KS solution is the maximal point that results in the equalization of the relative utility gains among the agents. The KS solution is obviously individually rational, and it is easy to see that it is invariant under affine transformations.

The KS solution is not guaranteed to be Pareto optimal. We illustrate this point with an example.

Example 18. Consider a three person bargaining problem whose disagreement point s^* is such that the utilities of the agents for it is $(0, 0, 0)$. The set of alternatives is given by s^* and five other points a, b, c, d, e for which the utilities are $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, $(1, 1, 0)$, $(1, 0, 1)$ respectively.¹³ Clearly the set of Pareto optimal lotteries are those that mix between d and e , and any such must necessarily give utility 1 to the third agent. It is easy to see that the KS solution here is one that gives utilities $(1/2, 1/2, 1/2)$ to the agents, which is Pareto dominated.

In fact, it is known that for bargaining problems with 3 or more agents, no solution exists that satisfies the axioms of Pareto efficiency and symmetry together with monotonicity [31].

B Subgame perfect reduction: Proof of Lemma 15

Proof. Given that α strictly $\mathcal{D}$ -dominates β , we know that α is the unique subgame perfect outcome of $K(\alpha, \beta, \mathcal{N})$.

Consider the extensive form game tree of the perfect information game $K(\alpha, \beta, \mathcal{N})$. Some of the terminal nodes, say the set $\mathcal{A}$, give the utility outcome α , while some, say the set $\mathcal{B}$, others give the utility outcome β . Given any two bargaining games A and B that each produce some outcome, the game tree of the Knockout mechanism $K(A, B, \mathcal{N})$ by appending the game trees of A and B to the nodes in $\mathcal{A}$ and $\mathcal{B}$, respectively.

Now consider a node $a \in \mathcal{A}$ in the game tree of $K(A, B, \mathcal{N})$. Since α is the unique subgame perfect outcome of game A , we know that a backward induction on the subtree below a , upto a , yields the outcome α . Similarly, for any node $b \in \mathcal{B}$, the same reduction yields the outcome β .

Since the reduced tree obtained from the above step is identical to that of $K(\alpha, \beta, \mathcal{N})$, and α is the unique subgame perfect equilibrium of $K(\alpha, \beta, \mathcal{N})$, a backward induction on this reduced tree yields α .

As a result, it must be that a backward induction on the full game tree of $K(A, B, \mathcal{N})$ also yields α , whence the claim is proved. $\square$

¹³ This example violates Assumption 2. We use it nevertheless for ease of exposition.