Efficient Allocation via Sequential Scrip Auctions

Gil Kalai (Hebrew University of Jerusalem and Microsoft Research),
Reshef Meir (Harvard University),
Moshe Tennenholtz (Technion-Israel Institute of Technology and Microsoft Research)

August 12, 2014

Abstract

We study the problem of allocating items to two agents with arbitrary valuation functions. Instead of actual money, we use sequential auctions with “scrip money”, i.e. money that has no value outside the auction.

By considering such scrip auctions as a special case of more general bidding games, we show that there exists a natural pure subgame perfect Nash equilibrium (PSPE) with the following desirable properties: (a) agents’ utility is weakly monotone in their budget; (b) a Pareto-efficient outcome is reached for any initial budget; and (c) for any Pareto-efficient outcome there is an initial budget s.t. this outcome is attained. In particular, we can assign the budget so as to implement the outcome with maximum social welfare, maximum Egalitarian welfare, etc.

We provide an efficient algorithm to compute a PSPE in bidding games, and draw implications of the above result for various games and mechanism design problems.

1 Introduction

Allocation of indivisible goods to heterogeneous agents has received much attention in the literature of both economics and computer science. There are various criteria that an allocation mechanism or protocol is expected to meet. Some of these criteria, such as efficiency, regard the final allocation. An allocation is efficient (or Pareto-efficient) if no other allocation is weakly preferred by all agents and strictly preferred by some. Other properties relate to the process by which the allocation is decided, where agents’ incentives and strategies play an important role. For example, if the allocation is based on agents reporting private information, then we would favor a mechanism that is truthful. Similarly, if the allocation is achieved via a game or a market in which agents participate, we would like a good allocation to be in equilibrium.

Often, the introduction of money solves much of the problems raised by indivisibility, and helps to align agents’ incentives via a simultaneous or sequential auction. Implementing efficient allocations without money is typically much harder.

In order to tackle this challenge, we introduce and study a variation of a sequential auction without money. Rather, each agent has an initial endowment of “scrip money”—currency that has no value outside the game.
While allocation using scrip money has been suggested both in theory and in practice (Huberman and Hogg, 1995; Zetland, 2005; Budish, 2011; Cole et al., 2013), typically the setting is quite different, with many bidders and private valuations. We focus on a variant of such an auction that is specific to two agents with public information. The only paper we are aware of in a similar setting is by Huang et al. (2012), which study a sequential auction of identical items (see Related Work).

In short, in the sequential scrip auction (SSA) mechanism items are offered for sale in some arbitrary order, and a first-price auction is held in each round. The higher bidder pays her bid to the lower bidder, and gets to decide who takes the item (if valuations are monotone, then w.l.o.g. the winner always takes the item). Note that the total amount of scrip money in the hands of the agents remains the same, emphasizing that this is a bargaining mechanism, with no notion of “revenue”.

To better understand the properties of sequential scrip auctions with heterogeneous items, we consider a more general class of games called bidding games. A bidding game is an extensive form game for two players, only instead of playing according to fixed turns, each player has an initial budget that is used to bid over turns. The winner in each such auction gets to play the current turn. It is not hard to see that any SSA with \( k \) items coincides exactly with a bidding game over a balanced binary tree of height \( k \).

Bidding games are by no means new. The most prominent example is the addition of bidding phases to recreational board games such as Tic-Tac-Toe, Chess, Go, etc. These zero-sum bidding games are known as Richman games, and were studied by Lazarus et al. (1999) and later by Develin and Payne (2010). See Related Work for more details on Richman games and other variations of bidding games. Yet, to the best of our knowledge, bidding games with general utility functions have not been previously studied. Whereas zero-sum games have a value that each player can guaranty, when extending the definition to general-sum games we can only talk about equilibrium. Since bidding games are extensive form games, a natural solution concept to look at is pure subgame-perfect equilibrium (PSPE).

Our contribution We prove that any bidding game over a binary tree has a natural PSPE that is Pareto-efficient regardless of how initial budgets are allocated. Moreover, every Pareto-efficient outcome is achieved for some initial budget allocation, and the utility of each agent is weakly increasing in her budget. By the observation above on the equivalence of bidding games and SSAs, our results immediately apply to the SSA mechanism, guarantying an efficient allocation of items to two agents. These properties hold for any valuation functions agents have over subsets of items, showing that for two rational agents with public information, the SSA mechanism implements the whole range of efficient allocations.

Our result can be applied not just to allocation problems, but also in other domains where two agents have combinatorial preferences over outcomes, including multi-issue voting (Lang and Xia, 2009) and selection of arbitrators (De Clippel et al., 2012). We show that the SSA mechanism satisfies desired criteria of ordinal fairness in these domains. Implications of our main result extend to other games such as the Nash bargaining game and Centipede games.

As a complementary result, we show that in bidding games on a non-binary tree, none of the

---

1 More precisely, we allow the winner to decide who plays next.
properties above is guaranteed, and all PSPE outcomes may be non-monotone and arbitrarily bad for both players.

Finally, we describe an algorithm that solves any binary bidding game in time that is polynomial in the number of states, and show how such an algorithm can be used so efficiently find a PSPE in SSAs with additive valuations and other succinct classes.

2 Model

For an integer $k$, we denote the set \{1, 2, \ldots, k\} by $[k]$. Unless explicitly mentioned otherwise, we assume that $N = \{1, 2\}$, i.e. that there are only two competing agents. We use the notation $-i$ (instead of $3-i$) to denote the player that is not $i$. We name player 1 the white player, and player 2 the black player.

2.1 Sequential scrip auctions

Definition 1. A sequential scrip auction (SSA) is a tuple $F = \langle N, K, (v_i)_{i \in N}, \tau \rangle$, where:

- $N = \{1, 2\}$ is the set of bidders;
- $K$ is the set of items. We denote $k = |K|$;
- $v_i : 2^K \to \mathbb{R}$ is the value function of agent $i \in N$;
- $\tau$ is a permutation over items $K$.

Intuitively, items $K$ are offered for sale according to order $\tau$, and in each turn agents bid for the current item. To complete the game we need to describe a particular auction rule, as well as the utilities of the agents given an allocation.

Typically in the auction literature items are auctioned in a first- or second-price auction where the highest bidder pays the (first or second) bid to the seller; and utilities are quasi-linear. In particular, an agent’s utility depends not just on the allocation, but also on the amount of money the agent paid for her items [Blumrosen and Nisan, 2007].

SSAs differ from these standard auctions in two ways. First, while we use the first-price auction rule, the higher bidder pays the other player, rather than paying to the seller. Second and more importantly, each player is initially endowed with a budget $B_i$ that can be used for bidding throughout the game. This budget is “scrip money” and has no value outside the game, thus the utility of $i$ from a bundle $S_i$ is $v_i(S_i)$, regardless of how much of the budget the agent spent. In case of a tie we treat agent 1 as the winner.

2.2 Bidding games

Definition 2. An extensive form game structure for two players is a tuple $G = \langle N, S, s_0, T, g_1, g_2, u_1, u_2 \rangle$, where:
• $N = \{1, 2\}$ is the set of players;
• $S$ is a set of game states;
• $s_0 \in S$ is an initial state;
• $T \subseteq S$ is a set of terminal states;
• $g_i : (S \setminus T) \rightarrow 2^S$ defines the valid moves of player $i$ in state $s \in S \setminus T$;
• $u_i : T \rightarrow \mathbb{R}$ defines the utility for player $i$ is terminal $t \in T$.

We make the following assumptions about the transition functions $g_i$. First, unless specified otherwise, we assume that there are no cycles. Formally, there is no sequence $s_1, s_2, \ldots, s_k$ where $s_k = s_1$ and $s_{j+1} \in g_1(s_j) \cup g_2(s_j)$ for all $j < k$. Second, unless specified otherwise, we assume that in every non-terminal state both players can play. Formally, that $g_i(s) \neq \emptyset$ for all $i \in N, s \in S \setminus T$. The utility function $u_i$ induces a complete preference order over terminals $T$. We denote $t \succeq i t'$ whenever $u_i(t) \geq u_i(t')$. Given a game structure $G$ and $s \in S$, we denote by $G|_s$ the subgame of $G$ rooted in $s$. The height of $s$ is the maximal distance between $s$ and a leaf $t \in T(G|_s)$. In particular, $\text{height}(G) = \text{height}(s_0)$.

The following two properties of bidding games are structural properties of the underlying game tree/DAG:

**Definition 3.** A bidding game $G$ is symmetric, if $g_1(s) = g_2(s)$ for all $s \in S$. That is, if the same set of moves is available to both players in every state.

Unless specified otherwise, all bidding games in this paper are symmetric. In symmetric games we only need to specify one transition function $g(s)$.

**Definition 4.** A bidding game $G$ is binary, if $|g(s)| \leq 2$ for all $s \in S \setminus T$.

**Playing the game** In order to complete the definition of the game, we also need a method to determine who plays at every turn. Traditionally, there is a turn function that assigns the current player for every state (e.g., alternating turns). Another way is to randomly select the current player in each turn, as in [Peres et al., 2007]. In this work we follow the bidding framework of [Lazarus et al., 1999; Develin and Pavone, 2010], where each player has an initial budget $B_i \in \mathbb{R}_+$ that is used for the bidding.

In each turn (suppose at state $s$), each player submits a bid $b_i \leq B_i$, and “next state” $s^*_i$, which is realized in case $i$ wins the round. We break ties in favor of player 1.

More formally, a strategy of $i$ in $G$ is composed of a pair $\langle s^*_i, b_i \rangle$ for every $s \in S \setminus T$, and every $B_i \in B$. We require that $b_i \leq B_i$, and that $s^*_i \in g(s)$, otherwise this is not a valid strategy.

Unless explicitly mentioned otherwise, bids can be real numbers. It is also possible to think of games where bids (and budgets) are restricted to integers, and we will consider this variation in Sections 3.1 and 4.3. We denote by $B$ the set of possible budget partitions (which are also the allowable bids). When bids are continuous, we will assume w.l.o.g. that there is one unit of budget
allocated to players, i.e. that \( B_1 + B_2 = 1 \). Since in either case the total budget is fixed, \( B_2 \) can always be inferred from \( B_1 \) and vice versa. We therefore use either \( B_1 \) or \( B_2 \) to denote a particular budget partition.

**Example—Majority** Consider the following (zero-sum) game \( G_{\text{maj}} \), depicted in Figure 1. In this game there are three turns, and the winner is the player which plays at least twice. Formally, \( S = \{0, 1, 2\}^2 \) is simply the number of times that each player played, \( s_0 = (0, 0); T = \{(s_1, s_2) \in S : s_1 = 2 \lor s_2 = 2\}; g(s_i, s_{-i}) = \{(s_i + 1, s_{-i}), (s_i, s_{-i} + 1)\} \) for all \( s \in S \setminus T; \) and \( u_i(t) = 1 \) iff \( t_i = 2 \) and 0 otherwise.

Note that if \( G_{\text{maj}} \) is played with alternating turns and without budgets or bidding, the white player must win. Consider \( G_{\text{maj}} \) with \( B_1 = B_2 = 0.5 \), i.e. where each player has an initial budget of 0.5.

The following is a possible game play:

- In turn 1, white bids 0.2 and black bids 0.15. White gains the turn, plays the round, and the new budgets are 0.3 and 0.7. \( s = (1, 0) \).
- In turn 2, white bids 0.14 and black bids 0.26. Black plays and pays 0.26. \( s = (1, 1) \), and budgets are updated to (0.56, 0.44).
- In the last turn, both players exhaust their budgets, so the player with higher remaining budget (white with 0.56 vs. black with only 0.44) plays and wins the game. \( s = (2, 1) \in T \).

Note that a higher budget makes it easier to win in this game. For example, if \( B_i > 3B_{-i} \) for some player, then player \( i \) can always gain the first two rounds.

### 2.3 SSAs are Binary Bidding Games

**Proposition 1.** Every SSA with \( k \) items has an equivalent bidding game over a binary tree of height \( k \), and vice versa.

While the full proof is in the appendix, it is not hard to see why this is true. Intuitively, we can describe any SSA as a binary tree where any internal node at depth \( k' \) corresponds to an allocation of the first \( k' \) items. Then taking the left branch means white takes the next item, whereas right branch means the item goes to black. In the other direction, we can assume w.l.o.g. that the binary tree is complete. Then we identify each of the \( 2^k \) leaves of the tree with an allocation of the \( k \) items.

For example, the game \( G_{\text{maj}} \) described above coincides with a sequential scrip auction of three identical items, where each agent assigns a value of 1 to bundles of size two or more, and 0 otherwise.
2.4 Subgame perfect equilibria

We will denote by $\gamma$ a strategy profile for the two players, where $\gamma(s, B_1)$ is the part of the profile that determines the actions taken in state $s$ under budget partition $(B_1, B_2)$. We interpret the notation $i \geq i'$ as $i \geq i'$ when $i = 1$ and as $i > i'$ when $i = 2$.

While the following definition is somewhat lengthy, it coincides with the standard definition of PSPE for the game described above. We write down explicitly all possible deviations of each player and explain them in comments.

**Definition 5 (PSPE).** A pure subgame perfect equilibrium (PSPE) in a bidding game $G$, is a mapping $\gamma$ from states to actions, s.t. each player plays a best-response strategy to the other player at any subtree, for any budget. Formally,

$$\gamma : S \times B \rightarrow N \times (B)^2 \times B \times S \times T;$$

where for $\gamma(s, B_1) = (i, b_1, b_2, B^*_i, s^*_i, t)$, the following hold (we denote by $\gamma(\cdot)_T \in T$ the terminal reached in this subtree):

- $i = 1$ if $b_1 \geq b_2$ and otherwise $i = 2$. That is, highest bidder takes the round with ties broken in favor of white.
- $B^*_i = B_i - b_i$. That is, winner pays her bid.
- $s^*_i \in \arg\max_{s' \in g(s)} u_i(\gamma(s', B^*_i)_T)$. That is, the next state is optimal for the winner (given her remaining budget).
- $t = \gamma(s^*_i, B^*_i)_T$. That is, we reach to the same terminal from this state and from the next state (consistency).
- $u_i(t) \geq \max_{b'_i > b_i} u_i(\gamma(s^*_i, B_i - b'_i)_T)$. That is, the winner $i$ cannot benefit by using a different bid and still take the turn.
- $u_i(t) \geq u_i(\gamma(s^*_{-i}, B_i + b_{-i})_T)$. That is, $i$ cannot gain by lowering her bid and dropping the turn (does not apply if $i = 1$ and $b_2 = 0$).
- For any $s'_{-i}$ and any $b'_{-i} \leq B_{-i}$ s.t. $b'_{-i} \geq b_i$, $u_{-i}(t) \geq u_{-i}(\gamma(s'_{-i}, B_i + b'_{-i})_T)$. That is, $-i$ cannot gain by increasing his bid and take the turn.

We emphasize that a PSPE determines the actions and outcome for any budget partition in any internal node, and in particular for any initial budget partition. Therefore, every PSPE $\gamma$ for $G$ induces a mapping $\mu_\gamma$ from $B$ to outcomes $T(G)$. That is, $\mu_\gamma(B_1) = \gamma(s_0, B_1)_T$. 


3 Existence of PSPE

We start by showing the following basic result.

**Theorem 2.** Every bidding game has a PSPE.

We will first prove the result for games with discrete bids, which holds regardless of our tie-breaking scheme. The proof for continuous bids is similar, but applies the assumption that ties are consistently broken in favor of the same player. Indeed, we will later see that this assumption cannot be relaxed.

3.1 The discrete case

Let a first-price game be a two-player one-shot game of the following form: each player submits a bid $b_i \in B$, and the payoffs $u_1, u_2$ are determined only by the higher bid.

**Lemma 3.** A first-price game where $B$ is finite has a pure Nash equilibrium. Moreover, any sequence of better-replies must converge.

**Proof.** We assume w.l.o.g. tie-breaking in favor of bidder 1. Assume, toward a contradiction, that there is a cycle of improvements. In particular such a cycle must contain a step where bidder 1 increases her bid from $(b_1, b_2)$ to $b_1'$. Note that $b_1' \geq b_2$, as otherwise $b_1 < b_1' \leq b_2$ and $(b_1, b_2); (b_1', b_2)$ lead to the same outcome. Any further step of bidder 1 must keep $b_1' \geq b_2$, otherwise we just go back to the outcome of $(b_1, b_2)$ (which is weakly worse for bidder 1).

Consider the next reply by bidder 2—he must also increase his bid to some $b_2' > b_1' \geq b_2$ in order to have any effect. Then bidder 1 must increase again and so on. Since no bidder ever reduces her bid (after the first increase) and budgets are bounded, the process must converge to a pure Nash equilibrium.

3.2 The continuous case

For the continuous case, we want to show that the exact budget does not matter. Informally, for games at height $k$ the interval $B = [0, 1]$ can be divided into $2^k$ “budget intervals” of equal size. All
budget partitions \((B_1, B_2)\) where \(B_1\) belongs in the same budget interval, would lead to the same outcomes. The next lemma states this formally.

**Lemma 4.** Let \(G\) be a bidding game, and denote \(k = \text{height}(G)\). Let \(\gamma\) be some strategy profile in \(G\). There is a strategy profile \(\gamma^*\) s.t. for any \(j \in \{0, 1, \ldots, 2^k - 1\}\): (a) \(\gamma^*\) and \(\gamma\) yield the same outcome under budget \(B_1 = j \cdot 2^{-k}\); (b) \(\gamma^*\) yields the same outcome under any budget \(B_1 \in [j 2^{-k}, (j + 1) 2^{-k})\). Also, \(\gamma^*\) is a PSPE if \(\gamma\) is.

For intuition, note that for \(k = 1\) it only matters if \(B_1 \geq 0.5\), since then white can force a win. For \(k = 2\), white can force two wins if \(B_1 \geq 0.75\), select whether to win in the first or second turn if \(B_1 \in [0.5, 0.75)\), etc. The full proof is in the appendix.

**Proposition 5.** There is a zero-sum game with state-specific tie breaking, that has no subgame-perfect equilibrium.

**3.3 Genericity and uniqueness**

Generic games are games where agents have strict preferences over all outcomes. In classical extensive-form games (without bidding), it is known that genericity entails the existence of a unique PSPE. However in our game there are simultaneous steps, and thus genericity may not be sufficient for uniqueness. See Appendix B.

**Enforcing generic preferences** In the general case, it is possible that a player is indifferent between two outcomes \(t, t' \in T\). In such cases, we will assume that the player has a strict preference towards the outcome that is also better for the other player. Thus every player has a strict preference order over all distinct outcomes. Recall that \(t \succ_i t'\) means that player \(i\) strictly prefers \(t\) over \(t'\). Thus in the remainder of the paper, \(t \succ_i t'\) if either \(u_i(t) > u_i(t')\), or \(u_i(t) = u_i(t')\).
and $u_{-i}(t) > u_{-i}(t')$. We highlight that this behavioral assumption only increases the number of potential deviations from a given state, and thus only narrows the set of PSPEs in a game.

**Ascending auctions and uniqueness** While a bidding game may have multiple PSPEs, we can maintain a unique equilibrium by changing the rules of the auction. Indeed, suppose that instead of a sealed-bid auction in every step, we hold an ascending auction (an English auction), where prices rise by $2^{-k-2}$ each time. Consider the proof of Theorem 2 and the related lemmas. It essentially shows that an ascending auction from any initial bids must converge, and in particular from the initial bids $(0,0)$. Moreover, we can write the ascending auction version of the game as a standard extensive-form game without simultaneous moves.

**Corollary 6.** Any generic bidding game $G$ with the ascending auction rule has a unique PSPE, which is the Lower PSPE of $G$. This also holds in non-generic games where players weakly prefer outcomes with higher social welfare.

### 4 Efficiency, Monotonicity, and Binary Trees

Having established the (rather weak) conditions for existence of a PSPE, we next turn to study the properties of this outcome, how it depends on the budget, and whether it is desired from the perspective of the players. This is the primary section of this work, where we lay out our main positive result.

A PSPE $\gamma$ is **monotone**, if $\mu_\gamma(B'_i) \geq_1 \mu_\gamma(B_1)$ whenever $B'_i > B_1$, and likewise for player 2.

Let $T_P(G) \subseteq T(G)$ be the set of Pareto-efficient outcomes in $G$. That is, $t \in T_P(G)$ if there is no $t' \in T(G)$ s.t. $t' \succ_1 t, t' \succ_2 t$. A PSPE $\gamma$ is **Pareto-optimal**, if $\mu_\gamma(B_1) \in T_P(G)$ for all $B_1 \in B$. A PSPE $\gamma$ is **Pareto-surjective**, if for all $t \in T_P(G)$, there is some budget $B_1 \in B$, s.t. $\mu_\gamma(B_1) = t$.

Intuitively, our main result is that binary games always have a highly desirable PSPE, which has all the above properties. We will first show that if the game is not binary, none of these properties is guaranteed. Violating Pareto-surjective is trivial (just consider a root with three children, whose payoffs are $(1,3), (2,2), \text{and} (3,1)$).

**Proposition 7.** There is a (non-binary) bidding game, where every PSPE violates monotonicity and Pareto-optimality.

**Proof.** Consider $G_{\text{bad}}$ (Fig. 2) under budget $B_2 = 0$. Clearly white can play at will, and thus the only PSPE leads to the outcome $(10,7)$ (marked). Note that if players reach $x$, then the outcome will be $(10,7)$ if $B_1(x) \geq 0.75$ and otherwise $(0,9)$. Now suppose that $B_1 = 0.8, B_2 = 0.2$.

---

2We thank David Parkes for this observation.
If black takes the first round, he will choose (1, 8), since by choosing $x$ we will have $B_1(x) \geq B_1 = 0.8 > 0.75$. In order to get to (10, 7), white has to get to $x$ with at least 0.75, that is to bid $b_1 \leq 0.05$. However if $b_1 < 0.2 = B_2$, then black will overbid and take the turn. Thus in every PSPE white must bid at least $b_1 = 0.2$ in $s_0$, and play (2, 1). This contradicts both monotonicity and Pareto: the utility of black dropped from 7 to 1 when his budget increased from 0 to 0.2, and (1, 2) is Pareto-dominated by (10, 7).

\[\square\]

### 4.1 Binary bidding games

Our main result is the following.

**Theorem 8.** Let $G$ be a binary bidding game. Then its Lower PSPE $\gamma$ is (A) monotone, (B) Pareto-optimal, and (C) Pareto-surjective.

The remainder of this section is dedicated to the proof of Theorem 8.

For convenience, we will denote by $\gamma|_s$ the projection of $\gamma$ on a subgame of $G$ that is rooted in $s$. Similarly, $\mu|_s = \mu|_s$. We will sometimes use the budget of player 2 as the input for $\mu$, rather than the budget of player 1. This should be clear from the notation. That is, $\mu(B_1)$ and $\mu(B_2)$ always refer to the same outcome in $T(G)$.

Denote the two children of $s_0$ by $s_t, s_r$. As the proof of all parts is in induction on the height of $G$, we assume that $\gamma$ has been computed on both subtrees $G_l = G|_{s_t}$ and $G_r = G|_{s_r}$, and denote $\gamma_l = \gamma|_{s_t}, \gamma_r = \gamma|_{s_r}$. Similarly for $\mu_l, \mu_r$.

**Lemma 9.** Algorithm computes the Lower PSPE of $G$.

**Proof.** Intuitively, $A_i$ is the strategy of $i$, where $A_i(s, B)$ will contain the action of $i$ in state $(s, B)$ (both bid and next move). $T^*(s, B) \in T$ is the outcome that will be reached from state $(s, B)$.

In every state $s$, players start from zero bids, and effectively run an ascending action with minimal increments to find the minimal stable bids (see comment after proof of Theorem 2 and Corollary 6). To show that this is the Lower PSPE, we only need to show that if there is a deviation for $i$, then increasing $b_i$ by $\epsilon$ is sufficient.

Indeed, we assume by induction that in each subtree we have a monotone PSPE $\gamma_l, \gamma_r$. Thus if a player can gain by increasing her bid, it is sufficient to raise by a minimal amount, i.e. by $\epsilon$. As in the proof of Lemma 3 players will never lower their bids. This proves that $\gamma$ is a PSPE (in particular the Lower PSPE).

We now turn to complete the proof of our main theorem.

**Proof of Theorem 8 (Monotonicity).** By the induction hypothesis, both of $\mu_l, \mu_r$ are monotone. Also, by the Pareto property (B), $\mu(B) \in T_P(G)$ for all $B$. Let $B'_1 = B_1 + \epsilon$, and let $b_1, b_2$ be the equilibrium bids in $\gamma(s_0, B'_1)$. Likewise, $b'_1, b'_2$ are the equilibrium bids in $\gamma(s_0, B'_1)$.

We prove first for the case where white takes $(b_1 = b_2)$. W.l.o.g., $\gamma(s_0, B_1)_S = s_t$, i.e. $\mu_l(B_1 - b_1) \geq 1 \mu_r(B_1 + b_2)$. If $b'_1 < b_1$, then by raising to $b_1$, white can guarantee a value of $\mu_l(B'_1 - b_1) \geq 1 \mu_l(B_1 - b_1)$. Thus its current value $\mu(B'_1)$ is at least the same. If $b'_1 = b_1 - \epsilon$, then black will not
Thus suppose \( \mu_l(B_1 - b_1) = \mu_l(B'_1 - b'_1) \) over \( \mu_r(B_1 + b_1 + \epsilon) = \mu_r(B'_1 + b'_1 + \epsilon) \). In either case, \( \mu(B'_1) \preceq_1 \mu_l(B_1 + b_1) = \mu(B_1) \).

Next, we prove for the case where black takes, in which \( b_1 = b_2 - \epsilon \). W.l.o.g., \( \gamma(s_0, B_2)_S = s_r \), \( \mu(B_2) = \mu_r(B_2 - b_2) = t_r \). Since black raised last, \( t_r = \mu_r(B_2 - b_2) \succ_2 \mu_l(B_1 - b_1) \). By bidding \( b_2 \) under \( B'_1 \), white can guarantee at least \( \mu_l(B'_1 - b_2) = \mu_l(B_1 - b_1) \succ_1 t_r \), where the last inequality is by Pareto. If \( b'_1 < b_2 \) then \( \mu(B'_1) \succeq_1 \mu_l(B'_1 - b_2) \succ_1 t_r = \mu(B_1) \) and we are done.

Suppose \( b'_1 \geq b_2 \). If black takes,

\[
\mu(B'_1) = \mu_r(B'_1 + b'_{2}) \succeq_1 \mu_r(B_1 + b_1) = t_r = \mu(B_1).
\]

If white takes, \( b'_2 = b'_1 \geq b_2 \) and thus

\[
\mu(B'_1) = \mu_l(B'_1 - b'_1) = \mu_l(B'_1 - b'_{2}) \succ_1 \mu_r(B'_1 + (b'_2 - \epsilon))
\]

\[
= \mu_r(B_1 + b'_2) \succeq_1 \mu_r(B_1 + b_2) = t_r = \mu(B_1).
\]

\( \square \)

**Proof of Theorem**

Let \( b_1, b_2 \) be equilibrium bids in \( s_0 \) reached by the above process. We use throughout the proof the fact that by monotonicity of both subgames, it only makes sense for a player to raise her bid if she selects the branch that is not currently played. Note that we may not use the fact that \( \mu \) itself is monotone, since the proof of (A) relies on the proof of (B).

Assume w.l.o.g. that \( b_1 = b_2 \) (white takes) and that under budget \( B_1 - b_1 \), white prefers \( s_l \). We argue that \( t^*_l = \mu_l(B_1 - b_1) \) is Pareto-efficient in \( G \). Assume, toward a contradiction, that there is some \( t^*_r \in T_P(G_r) \) that Pareto dominates \( t^*_l \). By induction on (3), \( t^*_r = \mu_r(B'_1) \) for some budget \( B'_1 \).

Case 1: Suppose first that \( B'_1 \succ B_1 \), i.e. that white must drop the turn in order to reach \( s_r \) with sufficient budget. Denote \( b' = B'_1 - B_1 \) (note that \( b' = B_2 - B'_2 \leq B_2 \)). We argue that \( b' \succ b_1 \). If \( b_1 = 0 \), then clearly \( b' > 0 \), thus suppose \( b_1 > 0 \). Then by construction of the equilibrium \((b_1, b_2)\), \( t^*_l \) is strictly preferred by white over the outcome of \((b_1 - \epsilon, b_2)\), which is \( \mu_r(B_1 + b_1) \) (as \( b_2 = b_1 \)). Thus,

\[
\mu_r(B_1 + b') = t^*_r \succeq_1 t^*_l \succ_1 \mu_r(B_1 + b_1).
\]

By monotonicity, we must have \( b' \succ b_1 = b_2 \). Next, observe that by generic preferences \( t^*_r \succ_2 t^*_l \). Thus black could raise his bid to \( b' > b_1 = b_2 \) and strictly gain. This shows that \((b_1, b_2)\) are not an equilibrium in \( s_0 \), in contradiction to the construction of \( \gamma \).

Case 2: \( B'_1 = B_1 - b' \) for some \( b' \geq 0 \). If \( b_1 = 0 \), then white would have preferred \( s_r \), as

\[
\mu_r(B_1) \geq \mu_r(B'_1) = t^*_r \succeq_1 t^*_l = \mu_l(B_1).
\]

Thus suppose \( b_1 > 0 \), and we will show that this will lead to a contradiction.

Since \( \mu_r(B_1 - b') = t^*_r \succeq_1 t^*_l \), we must have by monotonicity of \( \mu_r \) that \( 0 \leq b' \leq b_1 \). By the construction of the equilibrium bids, white strictly preferred \( t^*_l = \mu_l(B_1 - b_1) \).
over the previous state $\mu_r(B_1 + b_2) = \mu_r(B_1 + b_1)$ (where the bid of player 1 was strictly lower). Thus $$t_r^* \geq_1 t_r^* \geq_1 \mu_r(B_1 + b_1) \geq_1 \mu_r(B_1 - b') = t_r^*,$$
which is a contradiction.

\begin{proof}[Proof of Theorem \[8C\] (Pareto-surjective)]
We prove by induction on the height of the tree $G$. For height 0 it is obvious. For height 1 either one leaf weakly Pareto-dominates the other, in which case this leaf is reached in $\mu$ regardless of budgets, or each player has a favorite leaf in which case both leaves are Pareto-efficient.

Let $t^* \in T_P(G)$, we need to show that there is some initial budget $B \in B$ s.t. that $\mu(B) = t^*$. W.l.o.g. $t^* \in T_P(G_1)$.

By induction on property (C), let $B_1^* \in B$ s.t. $\mu_l(B_1^*) = t^*$, and let $t_r^* = \mu_r(B_1^*)$. We set $B_1^*$ s.t. by slightly increasing or decreasing the budget, the outcome will not change (i.e. it is not on the edge of its respective interval). Formally, for the $\epsilon$ used in Algorithm $[1]$, $\mu_l(B_1^* - 2\epsilon) = \mu_l(B_1^* + \epsilon) = t^*$ [also equal to $\mu_l(B_2^* - \epsilon)$]; and likewise for $\mu_r$. Since the size of budget intervals is $2^{-\text{height}(G)}$, setting $\epsilon < 2^{-\text{height}(G)}$ is sufficient.

We know that $t_r^*$ does not strictly Pareto dominate $t^*$ (as $t^* \in T_P(G)$), thus either $t_r^* \preceq_1 t^*$ or $t_r^* \preceq_2 t^*$.

Case I: Both players prefer $t^*$ over $t_r^*$ at budget $B_1^*$. We claim that the equilibrium bids are $b_1 = b_2 = 0$.

Suppose that black raises to $b' > b_2 = 0$, takes the round and goes right. then

$$\mu_r(B_2^* - b') \preceq_2 \mu_r(B_2^*) = t_r^* \preceq_2 t^*,$$

and thus black does not gain. Therefore $b_1 = b_2 = 0$, and $\mu(B_1^*) = \mu_l(B_1^*) = t^*$.

Case II: Suppose that $t_r^* \succeq_2 t^*$, $t_r^* \preceq_1 t^*$, i.e. that black strictly prefers child $s_r$ at budget $B_2^*$, whereas white strictly prefers $s_l$. Let $\hat{B}_2 \leq B_2^*$ be the lowest budget s.t. for every $\beta_2 \in [\hat{B}_2, B_2^*]$, $\mu_r(\beta_2) > \mu_r(t^*)$ (it is possible that $\hat{B}_2 = 0$). Set $t_r = \mu_r(\hat{B}_2)$. By Pareto it follows that for every $\beta_2 \in [\hat{B}_2, B_2^*]$, white prefers $s_l$ (since $\mu_r(\beta_2) \preceq_1 t^* = \mu_l(B_1^*) \preceq_1 \mu_l(\beta_1))$.

We set $B_2 = \frac{B_2^* + \hat{B}_2}{2}$, $B_1 = 1 - B_2$, and argue that $\mu(B_1) = t^*$. Let $b^* = \frac{B_2 - \frac{B_2^* + \hat{B}_2}{2}}{2}$. We will show that the equilibrium in $s_0$ is $b_1 = b_2 = b^*$.

As long as bids are strictly below $b^*$, the resulting budget is in the “conflict zone” $[\hat{B}_2, B_2^*]$, and the lowest bidder will raise to take the turn from the other bidder. Suppose now that $b_1 = b_2 = b^*$. If $\hat{B}_2 = 0$, then $b_2 = b^* = B_2^*$ and black cannot raise. Otherwise, by definition of $b^*$ if black raises his bid to $b_2' = b^* + \epsilon$, then

$$t^* \preceq_2 \mu_r(\hat{B}_2 - \epsilon) = \mu_r(B_2 - (b^* + \epsilon)) = \mu_r(B_2 - b_2'),$$

so black cannot gain by selecting $s_r$. Thus $b_1 = b_2 = b^*$ is an equilibrium, where $\mu(B_1) = \mu_l(B_1^*)$.

Case III: $t_r^* >_1 t^*$, i.e. white strictly prefers $s_r$ at budget $B_1^*$. Quite expectantly, the proof of case III is similar to case II, where white and black change roles. However, there are some fine issues due to tie-breaking, so we lay out the full proof.

\end{proof}
Let \( \hat{B}_1 \leq B_1^* \) be the lowest budget s.t. for every \( \beta_1 \in [\hat{B}_1, B_1^*] \), \( \mu_r(\beta_1) \succ_1 t^* \). Set \( \hat{t}_r = \mu_r(\hat{B}_1) \). As in case II, whenever the budget after the bid is in the range \([\hat{B}_1, B_1^*]\), white prefers \( s_r \) whereas black prefers \( s_l \). We define \( B_1 = \frac{\hat{B}_1 + B_1^*}{2} \), \( b^* = \frac{B_1^* - \hat{B}_1^*}{2} \), and argue that the equilibrium reached in \( s_0 \) is \( b_1 = b^*, b_2 = b^* + \epsilon \).

Indeed, if \( b_1 < b_2 \leq b^* \), then black selects \( s_l \) under the remaining budget \( B_2 - b_2 \geq B_2^* \), and we have

\[
\mu_r(B_1 - b_2) \succ_1 t^* = \mu_l(B_1^*) = \mu_l(B_1 - b^*) \succeq_1 \mu_l(B_1 - b_2).
\]

I.e., white will raise her bid to \( b_1' = b_2 \).

Similarly, if \( b_1 = b_2 \leq b^* \), then white selects \( s_r \), but black wants to raise to \( b_2' = b_2 + \epsilon \): since \( B_2 - b_2' \geq B_2 - b^* - \epsilon = B_2^* - \epsilon \), we have

\[
\mu_l(B_2 - b_2') \succeq_2 \mu_l(B_2^* - \epsilon) = \mu_l(B_2^*) = t^* \succ_2 \hat{t}_r = \mu_r(B_2 + b^*) \succeq_2 \mu_r(B_2 + b_1).
\]

It remains to show that \( b_2 = b_1 + \epsilon = b^* + \epsilon \) is an equilibrium, i.e. that white will not raise. This is exactly as in case II. Either white cannot raise, or \( b_1' > b^* \). In the latter case,

\[
\mu_r(B_1 - b_1') \preceq_1 \mu_r(B_1 - b^* - \epsilon) = \mu_r(\hat{B}_1 - \epsilon) \preceq_1 t^*,
\]

where the last inequality is by definition of \( \hat{B}_1 \). Thus white cannot gain by selecting \( s_r \), and clearly not by keeping \( s_l \) at a lower budget. \( \square \)

### 4.2 Monotonicity in Bidding Games

While Theorem\[^{[8]}\] shows that a monotone PSPE exists, this does not necessarily mean that an agent is always better off by having a higher budget, since the monotone PSPE may not be unique. We would like to rule out the possibility that increasing the budget of a player enables a PSPE that is strictly worse for her.

**Definition 6** (Monotonicity). A game \( G \) is monotone, if adding more budget to a player cannot make her worse off. Formally, if for all \( B_i' > B_i \), there is a PSPE that yields under \( B_i' \) an outcome that is at least as good for \( i \) as any PSPE under \( B_i \).

Note that if \( G \) is monotone, then it has at least one monotone PSPE (say, the one maximizing the utility of player 1 for every budget). However, the existence of a monotone PSPE in \( G \) does not entail that \( G \) is itself monotone.

The following theorem is valid for games with either discrete or continuous budgets.

**Theorem 10.** Any binary tree game \( G \) is monotone.

\[^{[8]}\]The equality is by our selection of \( B_2^* \) so that it is not on the edge of its respective interval. This is where the assumption of continuous/large budgets is applied.
4.3 Discrete bidding games

Consider bidding games where the bids and budgets are integers, and the total budget is some $M \in \mathbb{N}$. Which results still go through and which results change? When considering existence, we showed in Section 3.1 that a PSPE exists whether bids are continuous or discrete, and in fact the proof for the discrete case is easier.

The proof of Theorem 10 (monotonicity) works just the same with discrete bids, for any $M$. We only need to change $\epsilon$ (the minimal significant budget change) with $1$.

How about Pareto-optimality? If $M \geq 2^{\text{height}(G)+2}$, then we effectively simulate the continuous case, since the budget interval $\{0, 1, \ldots, M\}$ can be partitioned to $2^{\text{height}(G)}$ sub-intervals whose size is at least $3$, and the exact budget within each sub-interval has no effect on the outcome. Thus Theorem 8 holds in the discrete case when $M$ is sufficiently high (i.e., exponential in the height of the game tree). If $M$ is too low, this is no longer true.

Proposition 11. For any $k \in \mathbb{N}$ there is a binary bidding game $G_k$, s.t. if $G_k$ is played with a discrete budget of $M < 2^k$, it has no Pareto-optimal PSPE.

5 Implications

There are two ways we can think of bidding games. First, we can look for games that inherently have budgets and sequential bidding, and there are some examples of scenarios that arguably fit into this model, typically the incentive structure and/or the bidding structure is quite different (see Section 5.3 below, and the Related work section). A more promising way is to think of the sequential bidding process as a mechanism that is designed to increase cooperation and welfare among a pair of players in various scenarios.

5.1 Back to Scrip Auctions

Since SSAs are essentially equivalent to bidding games on binary trees, an immediate corollary of Theorem 8 and Prop. 1 is the following. For a set of items $K$, let $T_P(K)$ be the set of Pareto-efficient allocations of $K$ among two players.

Theorem 12. Let $F = \langle N, K, (v_i)_{i \in N}, \tau \rangle$ be a sequential scrip auction (with two players). Then there is a PSPE $\gamma$ in $F$ s.t. $\mu = \mu_\gamma$ is a monotone and surjective mapping from $B$ to $T_P(K)$.

As we mentioned in the introduction, the set of items $K$ need not be a collection of physical items. We next consider some applications of SSAs with “virtual items”.

5.2 Voting and Arbitration

Multi-issue voting Consider a set of voters, voting over $n$ binary issues [Winter, 1997; Lacy and Niou, 2004; Lang and Xia, 2009]. Here the voters (who are the players) have a complete (weak) preference order over the $2^n$ outcomes, but need not attribute cardinal utilities to them. It is known
that when preferences are unrestricted, either truthful or strategic voting may lead to the selection of a Pareto-dominated outcome. Voting on the issues sequentially provides a partial solution, but Pareto-dominated outcomes may still be selected [Lacy and Niou, 2000]. While our model only applies for two voters, it provides strong guaranties of Pareto-efficiency, regardless of the agenda or the structure of preferences. In particular, a Condorcet loser is never selected, and the Condorcet winner is always selected if exists.

**Arbitration** For two self-interested parties, a bidding game can be used as an arbitration mechanism that reaches an efficient outcome, and also allows a natural way to take into account parties of different importance, or weight. As a concrete example, we can think of the Democrats and the GOP bargaining over the various clauses of the Health Act, or some other reform. The initial budget of each party can be set, say, based on its number of seats. Arbitration mechanisms for two parties (not necessarily in a combinatorial setting) have been widely studied in the literature [Sprumont, 1993; Anbarci, 2006; De Clippel et al., 2012].

The minimal satisfaction test (MST) [De Clippel et al., 2012] is an ordinal fairness criterion, that is satisfied if the selected outcome is above the median outcome in every player’s preference order. That is, every player weakly prefers the selected outcome to at least half of all possible outcomes. De Clippel et al. studied several sequential mechanisms, and showed that three of them—The Alternate-Strike mechanism [Anbarci, 2006], the Voting by Alternating Voters and Vetoes mechanism [Sprumont, 1993], and the Shortlisting mechanism—each implement some Pareto-efficient outcome that satisfies MST.

**SSA as a fair arbitration mechanism** Consider the following parametrized extension of the MST requirement.

We say that an outcome $t$ satisfies the $(\alpha_1, \alpha_2)$-satisfaction test, if each player $i$ weakly prefers $t$ to at least $[\alpha_i|T|]$ outcomes.

**Proposition 13.** Let $G$ be a bidding game over a full binary tree, then any PSPE outcome satisfies the $(B_1, B_2)$-satisfaction test.

In particular, for equal budget we get that the outcome satisfies the minimal satisfaction test of De Clippel et al. [2012].

*Proof.* We will prove that there is a strategy for player 1 that guarantees an outcome weakly preferred to $[B_1|T|]$ outcomes, and likewise for player 2 and $[B_2|T|]$. Since any equilibrium outcome is at least as good as the maximin outcome, this would entail the proposition.

Given $G$ and $B_1$, we define a zero-sum game $G'$, that is equivalent to $G$ except for the utility functions. We set $u'_1(t) = 1$ if $t$ is weakly preferred to at least $[B_1|T|]$ terminals, and $-1$ otherwise; and set $u'_2 = -u'_1$. Thus each player can either ‘win’ or ‘lose’ in every terminal. Note that if $B_1|T|$ is an integer, then there are exactly $|T| - B_1|T| + 1 = (1 - B_1)|T| + 1$ winning nodes, and $B_1|T| - 1$ losing nodes. Otherwise there are $[B_1|T|]$ losing nodes, since the next one is weakly preferred to all of them plus itself, i.e. to $[B_1|T|]$ terminals. Thus the number of losing nodes is exactly $[B_1|T|] - 1$. 

15
According to Lazarus et al. [1996], the Richman value \( R(s) \) of a node in a zero-sum win/lose game, is the minimal value s.t. any \( B_1 > R(s) \) guarantees a victory for player 1. Lazarus et al. show that \( R(s) \) equals exactly to the probability that player 1 loses in the corresponding “spinner game” (the same game only turns are taken at random rather than by bidding).

This probability is easy to compute. Since \( G' \) is a zero-sum game, w.l.o.g. at every node the players prefer a distinct child. Thus if players take random turns, the essentially reach a terminal that is selected uniformly at random from \( T \) (since at every node they go right w.p. 0.5). We conclude that

\[
R(s_0) = Pr_{t \sim U(T)}(u_1(t) = -1) = \frac{|\{t : u_1(t) = -1\}|}{|T|} = \frac{[B_1|T|] - 1}{|T|} < B_1,
\]

which means that a budget of \( B_1 \) is sufficient to guarantee a victory to player 1 in \( G' \). By definition of \( G' \), player 1 can guarantee an outcome in \( G \) that is weakly preferred to \([B_1|T|] \) outcomes. The proof for player 2 is symmetric.

What if \( G \) is not a full tree? Then some terminals may be more likely then others and \((B_1, B_2)-ST \) may not hold \textit{ex post}. However if for a given set of outcomes \( T \) we construct a game tree at random, it is easy to see that the criterion holds in expectation. Further, We can always construct a tree such that all terminals are at depth \( h \) or \( h - 1 \). In such tree the probability of selecting a shallower terminal is exactly double. It can be shown that in every such instantiation, the SSA mechanism still guarantees a \((B_1, B_2)-ST \) outcome.

The SSA mechanism has the additional property of implementing the entire set of Pareto-efficient outcomes. As the proposition shows, every such outcome that is attained under a given budget allocation, satisfies the parametrized MST. Thus SSA may be a considered as a desirable mechanism for selecting arbitrators when the playing parties are asymmetric.

\textbf{Zero-sum games} In a zero-sum game, player 1 gets an outcome that beats \( B_1|T| \) outcomes, if and only if player 2 gets an outcome that beats \( B_2|T| \) outcomes. Thus by Prop [13] the set of outcomes and the utility functions define the (unique) PSPE outcome completely, regardless of the tree structure.

\textbf{Communication complexity} In combinatorial settings (such multi-item allocation or multi-issue voting) the SSA mechanism has an additional practical advantage over other mechanisms in terms of \textit{communication complexity}. The \textit{Alternate-Strike} mechanism and the \textit{Voting by Alternating Voters and Vetoes} mechanism [Sprumont, 1993; Anbarci, 2006] both require a number of rounds that is linear in the total number of outcomes. The \textit{Shortlisting} mechanism [De Clippel \textit{et al.}, 2012] only has two rounds, but requires the first player to specify some subset that includes half of the outcomes. As the number of outcomes is exponential in the number of items/issues, all three mechanisms result in exponential communication complexity.

In contrast, under the SSA mechanism, every player is only required to make a number of decisions that is linear in the number of items/issues, and each decision includes a bid and a binary choice. Thus even if agents are unable to accurately compute the equilibrium strategies, they are still
able to play large combinatorial games. Thus to the best of our knowledge, the SSA mechanism (with equal budgets) is the first mechanism that guarantees both Pareto efficiency and MST with low communication complexity. As we show on Section 3 for some classes of combinatorial games we can even compute the equilibrium efficiently.

**Lobbying** A bidding variation for a particular two-player game has been suggested by Le Breton and Zaporozhets [2010] in the context of political lobbying. There, two opposing lobbies compete by offering conditional payments (i.e., bribes) to members of a legislative body. The sequential setting is quite different, so this game does not quite fit in our model. Also, in [Breton and Zaporozhets, 2010] there are only two possible outcomes and the sum of utilities (of the lobbies) is constant. However, we could easily formulate a bidding game under our model using the same story, and allow multiple outcomes with different utilities for the two lobbies.

**Multi-agent pathfinding** A widely-studied problem in multi-agent systems, is that of finding paths that take a number of agents from their source locations to their targets [Silver, 2005; Sharon et al., 2012]. As paths may intersect, the agents affect one another, and self-interested agents may choose paths that severely delay other agents [Bnaya et al., 2013]. We can think of this problem as an allocation of space-time segments among the agents. For two agents, the SSA mechanism guarantees that the agents will settle on paths that are Pareto-efficient.

### 5.3 Other games

Consider any (symmetric) game in tree-form. If the tree is not binary, we can modify it by recursively breaking decision nodes to a subtree of binary decisions. Then we can divide a large or continuous budget among the players, and let them play the new bidding game. If players adhere to the Lower PSPE, then the initial budget partition will determine the final outcome (from $T_P(G)$). Thus we can get a good outcome even in games where playing by turns would lead to a poor outcome. In particular, with proper initial budgets we can implement the outcome with as maximum social welfare, maximum Egalitarian welfare, etc.

**Centipede games** One class of games that falls under the conditions of Theorem 8, is that of Centipede games [Rosenthal, 1981]. Such games are a notorious example to how rationality leads players to end up in poor outcomes. Under random turn there are still Centipede games where players always choose to finish early, even though staying in the game is eventually significantly better for both.

Nevertheless, in our game setting there is a PSPE where players are guaranteed to continue until they reach one of the Pareto-efficient outcomes (the last two leaves).

---

4Of course, there are many other mechanisms with low communication complexity, such as alternating turns. However they may not guarantee Pareto efficiency and/or MST.
**Nash bargaining game** The Nash bargaining game [Nash, 1950] for two players is typically described by a (convex) set of feasible outcomes \(F\) in the plane, whose boundary forms the Pareto-efficient frontier, and a status-quo point \(q\). Given \(F\) and \(q\), we can think of an extensive-form bargaining game, where players start from \(q\) and at each state (point of the grid) they can either “go right” (increase the utility of player 1) or “go up” (increase the utility of player 2). The terminal states are the outcomes along the Pareto frontier. In this case it is clear that every outcome is Pareto (as this is how we defined the terminal states), but the properties of monotonicity and Pareto-surjective are still interesting. The (unique) PSPE induces a generalized solution concept for the Nash bargaining game (a point on the Pareto frontier for any budget allocation).

A natural question is how the PSPE solution—in particular for the case of equal budgets—compares with other solution concepts such as the Nash bargaining solution and the Kalai-Smorodinsky bargaining solution [Kalai and Smorodinsky, 1975]. Since the SSA induced by a Nash bargaining game is zero-sum (always selects an outcome on the Pareto frontier), it follows from Proposition 13 that when the budget is equally partitioned, the realized outcome is reached when players make the same number of moves. In other words, if we normalize \(q\) to the origin \((0, 0)\), then both players have the same utility (or the closest approximation to same utility), which coincides with the Egalitarian, or proportional, solution [Kalai, 1977].

### 6 Computational issues

From the computer science perspective, a natural question regarding Algorithm 1 is its computational complexity. In a naïve implementation, we would write any auction with \(k\) items as a complete (balanced) binary tree. Then we would traverse the game tree once, and update the strategy tables for every possible budget.

The number of states is \(|S| = 2^{k+1} - 1\), and the required budget resolution is \(\epsilon = \Theta(1/\exp(\text{height}(G))) = \Theta(2^{-k})\). Thus the runtime of the naïve algorithm is \(\Theta(|S|\epsilon^{-1}) = \Theta(2^k)\), i.e. exponential in the number of items. In the general case this is in a sense inevitable, since the description of the valuation functions itself might be exponential in \(k\). However, many valuation functions have succinct representation, and we would like to at least be able to efficiently compute a PSPE in those cases.

We approach this computational challenge by identifying two separate problems. the first problem is to compile a given valuation function into a succinct bidding game representation. That is, to construct either a tree or a DAG with \(|S| = \text{poly}(k)\). As noted above, this is not always possible but we will show in Section 6.1 how to construct succinct DAGs for some broad classes of auctions. The second, more fundamental problem, is to efficiently compute a PSPE in time polynomial in \(|S|\), regardless of the height. Interestingly, there is a solution that works for arbitrary binary bidding games.

**Proposition 14.** Let \(G\) be a binary bidding game. The Lower PSPE \(\gamma\) can be computed in time \(|S| \cdot \text{poly}(|T|) = \text{poly}(|S|)\).

As an outline for the proof, we explain why \(\gamma\) has a polynomial size representation. In each state \(s \in S\), there is a step function, where in each interval we can write down the equilibrium
as a mapping to the next state that each player would choose, and to the outcome that would be attained. By Lemma 4 the number of intervals in this function is at most $2^{\text{height}(e)}$, which can still be exponential in $|S|$. However due to monotonicity (Theorem 8), the actual number of intervals is at most $|T| \leq |S|$ (in fact at most $|T_P(G)|$, since every outcome may occur only once. It is left to show how this step function can be efficiently computed by backward induction. See Algorithm 2 in the appendix for details.

6.1 Succinct Auction Representations

We next briefly explain how to take some succinct valuation classes, and compile them to a bidding game with a small DAG. See Appendix C for more details.

**Additive and weight-based valuations**  We first solve a simple case, where valuations are additive, i.e., $v_i(S) = \sum_{j \in S} v_i(j)$. A corresponding bidding game can be constructed using a dynamic programming technique, similar to the one used for Knapsack.

**Proposition 15.** Given an additive SSA $F$ with integer valuations, there is an equivalent bidding game with $|S| \leq (v_1(K) + 1) \times (v_2(K) + 1)$. Further, such a bidding game can be constructed efficiently.

Intuitively, we assign a state $s_{m_1,m_2}$ for every partial assignment where agent $i$ has items of total value $m_i$. We then connect states that differ by the value of a single item.

A similar technique can be applied in games where the valuations are not additive, but still based on some additive notion of weight. We say that $v_i$ is weight-based if every item has a fixed weight $w_j$, and there is a function $f_i : \mathbb{N} \to \mathbb{R}$, s.t. $v_i(S) = f_i(\sum_{j \in S} w_j)$. Note that without further constraints any function is weight-based, as we can make sure that each bundle has a different total weight. Weight-based valuations have a bidding game representation with at most $w(K)^2$ states. This idea can be further generalized to value that depends on several different quantities. As a concrete example, suppose that the items are computing machines, each with some properties like storage, memory, bandwidth, etc. The value of a set of machines to a client depends only on the total storage, total memory, and total bandwidth, regardless of how these resources are allocated across the machines.

As a corollary from Proposition 14 a PSPE in weight-based and in additive SSAs can be computed efficiently. That is, in time that is polynomial in the number of items and in the total weight/value (and exponential in the number of dimensions).

**Voting with single peaked preferences**  If voters have a strict preference order over all $2^k$ possible outcomes of the game, there cannot be a succinct representation, since we will need $2^k$ terminals. However consider a scenario, where each voter as an “ideal” point $t_i^* \in \{0,1\}^k$. The utility of any other outcome only depends on its distance from $t_i^*$, i.e. on the number of coordinates in which $t, t_i^*$ differ (possibly assigning different weights to different coordinates). This scenario is a special case of an SSA with additive valuations, where the items are the coordinates on which the two voters disagree.
The items in a pathfinding problem over a graph, are pairs \( s = \langle \text{location}, \text{time} \rangle \). In a naïve representation, all pairs would be allocated, and the value of a bundle of pairs to an agent, would be the optimal legal path from source to target that this bundle contains. If our game states would each represent a subset of such pairs, we would quickly reach an exponential blowup. However, it turns out that with a somewhat different representation, we can construct a bidding game with a most \( \text{MaxTime}^2 \times |\text{locations}|^4 \) states.

### 7 Related Work

**Richman games** Lazarus et al. [1996, 1999] were the first to systematically analyze bidding variations of zero-sum games. They coined the term “Richman games” in honor of David Ross Richman, the original inventor, and considered games with an infinitely divisible unit of budget. A Richman game is a directed graph (possibly with cycles) with two terminal nodes (say, black and white), and a full play is a path starting from some node and ending in a terminal node. The goal of each player is to end the game in her own terminal. The main focus of Lazarus et al. was on the following question: “in every node, what is the minimal fraction of the budget that will guarantee a victory for the white player?” The Richman function assigns a unique value to every node \( R(s) \), which is the average of the Richman values of its lowest and highest neighbors (normalizing the values of white and black terminals to 0 and 1, respectively). Lazarus et al. show that the Richman function exists on every graph and is also unique if the graph is finite. It turns out that \( R(s) \) marks the critical budget: if \( B_1 > R(s) \) then white can force a victory when starting from node \( s \); if \( B_1 < R(s) \) then black can force a victory. Moreover, \( R(s) \) is also the exact probability that black wins in a game where instead of bidding, the player in each turn is selected by a fair coin toss (see [Peres et al., 2007] for more details on random-turn games).

Lazarus et al. further study other variations of bidding games by applying different auction rules. In particular, in the *Poorman game* the highest bidder pays the bank rather than to the other player, so the total budget shrinks with every step.

Develin and Payne [2010] extended the theory of zero-sum bidding games in several important aspects. In most recreational games, such as Tic-Tac-Toe, Chess, and Checkers, white and black can perform different actions when in the same board state \( s \), thus games are not necessarily symmetric. Develin and Payne showed that previous results of Lazarus et al. go through even in asymmetric games. In addition, they considered discrete bids, and the implications of various tie-breaking schemes. To demonstrate their approach, Develin and Payne showed a complete solution of bidding Tic-Tac-Toe for every possible initial budget. Interestingly, in the continuous case the Richman value of the initial state in Tic-Tac-Toe is \( \frac{133}{256} \equiv 0.519 \), which means that an advantage of 4\% in the initial budget is sufficient to guarantee a victory.

**Sequential combinatorial auctions** “Standard” sequential auctions (with quasi-linear utilities rather than budgets) have been studied by several researchers [Gale and Stegeman, 2001; Rodriguez, 2009; Leme et al., 2012]. These papers focused on particular classes of value functions (unit-demand, submodular, etc.) and generally demonstrated that while pure equilibria exist, they may
be substantially inefficient for some of these classes. Inefficient outcomes occur even in 2-buyer, complete information auctions \cite{Bae et al., 2007}.

**Sequential Colonel Blotto**  In another recent paper, Powell \cite{2009} models a particular sequential game between an attacker and a defender (a sequential Colonel Blotto game). This game is essentially a specific general-sum bidding game over a degenerated tree (a path), where the utility in every match—the success probability—is determined by the invested resources of each party. Our model does not cover this particular game due to several differences. First, in our model the utility is only determined in the leaves rather than accumulated over the entire path. More importantly, the utility in every match depends exactly on the “bids” (investments) of both parties, and both parties discard their bids (an “all-pay” auction).

**Sequential auctions with budgets**  Lastly and closest to our work, Huang et al. \cite{2012} study particular bidding game, in which two agents use an initial budget to bid over identical items that are sold sequentially. This is essentially a sequential scrip auction, yet their model is not strictly a special case of ours, as they add some additional refinements. While we believe that the models are very close, the focus on a particular value function (which makes the game an almost-zero-sum game) allows Huang et al. to provide an accurate characterization of the allocations under PSPE as a function of the budgets. Our results are qualitative in nature, but apply to sequential auctions with arbitrary utility functions.

Huang et al. prove that item prices decrease over time. Curiously, we observe (empirically) that prices increase. It is possible that this is an artifact of the auction rule, as in the Poorman version the total amount of money is also decreasing in each round.

8  **Discussion**

We presented a simple and intuitive mechanism—the sequential scrip auction—that implements the full range of Pareto-efficient allocations of items to two agents with arbitrary valuation functions. Further, sequential bidding with budgets can be applied to efficiently and fairly settle other combinatorial bargaining problems that involve two parties.

8.1  **Variations**

By changing the auction rules, different types of bidding games arise. For example, an all-pay auction may better describe various real world scenarios like security games, sport matches, and

---

5This would be a minor difference if the number of outcomes was finite.

6In \cite{Huang et al., 2012} players pay to the bank rather than another (the Poorman variation), and also try to minimize spent budget as a secondary goal.

7Using an implementation of Algorithm in Matlab, we computed the PSPE prices for an SSA with k identical items. When the initial budget is the same (0.5 for each player), we witness a clear pattern of increasing prices over time. This pattern breaks when we start from different initial budget allocations.
R&D competitions. However these games typically do not have a pure equilibrium point, and are more difficult to analyze (see Related work section for specific games that have been studied).

Using a second-price rather than a first-price auction to determine the winner should not have a large effect, since in equilibrium the bids will always be very close w.l.o.g.

An important question is whether our results still hold when the “Poorman game” version is played, i.e. when the highest bidder pays the bank rather than the other player. It seems that all proofs should go through with minor variations, but we have not yet verified this. The Poorman version has one important advantage over the Richman version—the game definition naturally extends to any number of players. It is an open question whether a Pareto-optimal PSPE exists in a binary game for more than two players. We believe that the answer to this question is negative, since there are simple games with three players that are non-monotone.

8.2 Future Work

Other than studying the variations mentioned above, there are several questions that remain open. One natural direction is how changing the order in which items are sold (or the agenda in sequential voting) affects the induced equilibrium. While in the general case this effect can be substantial, it is possible that there are classes of valuations where all orders yield the same outcome in PSPE. For example, one corollary of Proposition 13 is that the order in zero-sum games does not matter.

Another important question is whether similar results regarding efficiency and monotonicity apply in games without complete information.
A Proofs and Examples

PROPOSITION 1. Any SSA with $k$ items is equivalent to a binary bidding game of height $k$. Also, any binary bidding game of height $k$ is equivalent to an SSA with $k$ items.

By “equivalent to” we mean there is a one-to-one mapping between strategy profiles in the original and induced game, that preserves utilities.

Proof. In the first direction, given $F = \{N, K, v_1, v_2, \tau\}$, let $K_d \subseteq K$ be the set of first $d$ elements of $K$ (according to $\tau$). We construct a bidding game $G = \langle N, S, S_0, T, g, u_1, u_2 \rangle$ as follows. Let $S_d$ be the set of all $2^d$ possible partitions (to two parts) of $K_d$ for $d \in \{0, 1, \ldots, k\}$, and $S = \bigcup_{d \leq k} S_d$. Note that these are partial allocations of $K$ to the two players. $s_0$ is the empty allocation (the unique member of $K_0$). Every state can be written as $s = (K^1, K^2) \in S_d$, where $K^1$ is the set of items held by player $i$ in state $s$. There are two states accessible from $s$: either player 1 or player 2 get item $\tau(d+1)$. Thus $g(s)$ contains exactly two states in $S_{d+1}$. The set of terminals $T$ coincides with $S_k$, i.e. all full partitions of $K$. Finally, for some $t = (K^1, K^2)$, we have that $u_i(t) = v_i(K^i)$. Thus $G$ is a symmetric binary tree of height $k$, which is clearly equivalent to $F$.

In the other direction, suppose we are given a binary bidding game $G$ of height $k$. W.l.o.g. $G$ is a balanced tree: if it is in DAG form, we can clone every node with several incoming edges, along with its subtree; if some branches are shorter than $k$, we extend them in the trivial way: replace each terminal $t$ at depth $k' < k$ with a balanced binary subtree of height $k - k'$, all of whose leaves are identical to $t$.

Now we have a balanced tree of height $k$. For every internal node, arbitrarily label one child as “left” and the other as “right”. We construct an SSA $F = \langle N, k, v_1, v_2, \tau \rangle$ as follows. Let $\tau$ be the identity permutation over $K = [k]$. We identify any path from the root $s_0$ to a terminal $t$ with a partition $(K^1, K^2)$ of $K$ where $K^1$ contains all levels $d$ s.t. the left child was selected, and $K^2$ contains all other levels (i.e., where the right child was selected). Note that any $K' \subseteq K$ appears in exactly one terminal $t'_1$ as $K^1$, and in exactly one terminal $t'_2$ as $K^2$. We set $v_i(K') = u_i(t'_i)$. Thus the set of allocations in $F$ coincides with the set of paths in the original game $G$.

\[\square\]

LEMMA 4. Let $G$ be a bidding game, and denote $k = \text{height}(G)$. Let $\gamma$ be some strategy profile in $G$. There is a strategy profile $\gamma^*$ s.t. for any $j \in \{0, 1, \ldots, 2^k - 1\}$: (a) $\gamma^*$ and $\gamma$ yield the same outcome under budget $B_1 = j \cdot 2^{-k}$; (b) $\gamma^*$ yields the same outcome under any budget $B_1 \in [j2^{-k}, (j+1)2^{-k})$. Also, $\gamma^*$ is a PSPE if $\gamma$ is.

Proof. We show by induction on $k$. Clearly for $k = 0$ there is just one budget interval, and no strategies. For intuition, note that for $k = 1$ it only matters if $B_1 \geq 0.5$, since then white can force a win. For $k = 2$, white can force two wins if $B_1 \geq 0.75$, select one win if $B_1 \in [0.5, 0.75)$, etc.

More generally, assume that the lemma holds for height at most $k - 1$. We replace $\gamma$ with $\gamma^*$ in all subtrees of $s$. Now we construct $\gamma^*(s)$. For any $j \in \{0, 1, \ldots, 2^k\}$, $B_1 = j2^{-k}$, set strategies in $\gamma^*(s, B_1)$ as in $\gamma$. Take some $j \in \{0, 1, \ldots, 2^k - 1\}$ and denote the bids under $j2^{-k}$ by $b_1, b_2$. For $B_1 \in [j2^{-k}, (j+1)2^{-k})$, we denote $\delta = B_1 - j2^{-k}$. We prove for the case where white wins in $(s, j2^{-k})$, i.e. that $b_1 \geq b_2$, and w.l.o.g. $b_2 = \min\{b_1, B_2\}$. The proof for the case where
black wins is symmetric. We set \( b_1^* = b_1 + \delta \) and \( b_2^* = \min\{b_1^*, B_2\} \). With budget \( j2^{-k} \), the remaining budget was \( j2^{-k} - b_1 = (2j)2^{-(k-1)} - b_1 \). With the new budget \( B_1 \), white still wins, and the remaining budget is \( B_1 - b_1 = j2^{-k} + \delta - (b_1 + \delta) = j2^{-k} - b_1 \). Thus (a) and (b) of the Lemma hold by construction. Also, if \( \gamma \) is a PSPE, then by construction (and induction) there are no deviations from \( \gamma^* \) at the budget points \( j2^{-k}, j \in \{0, 1, \ldots, 2^k\} \). We need to show that there are no deviations as \( \langle s, B_1 \rangle \), \( B_1 = j2^{-k} + \delta \) for some \( j < 2^k \) and \( \delta < 2^{-k} \).

Clearly black does not have new deviations in \( \langle s, B_1 \rangle \), since any such deviation would also be a deviation in \( \langle s, j2^{-k} \rangle \). Suppose that white has a deviation from \( b_1^* \) to \( b_1^* + \tau (\tau \neq 0) \). If \( b_2^* = b_1^* \), then \( b_2 = b_1 \) and white had a similar deviation from \( b_1 \) to \( b_1 + \tau \).

Thus suppose that \( b_2^* = B_2 < b_1^* \), and white deviates by decreasing her bid. If \( b_1' < b_2^* = B_2 \) (white drops), then after bidding black has 0 budget. Since \( b_2 + \delta \geq B_2 = 1 - j2^{-k} - \delta \) we have that at \( \langle s, j2^{-k} \rangle \), the remaining budget of black if white drops is \( 1 - j2^{-k} = b_2 \leq 1 - j2^{-k} - (1 - j2^{-k} - 2\delta) = 2\delta < 2^{-(k-1)} \). Thus by induction, the outcome under \( \langle s, j2^{-k} \rangle \) (with winning bid \( b_2^* \)) and under \( \langle s, B_1 \rangle \) (with \( b_2^* \)) is the same: both fall in the budget interval \( B_2' \in (0, 2^{-(k-1)}) \) (equivalently, \( B_1 \in [1 - 2^{-(k-1)}, 1) \)).

If \( b_1' \geq b_2^* \) (white still takes). Then we claim that \( B_1 - b_1^*, B_1 - b_1' \) belong to the same budget interval. Suppose otherwise, then since \( |b_1' - b_1'| < \delta \), \( B_1 - b_1^* \) is close to the right end of its interval, i.e., \( j2^{-k} - b_1 = B_1 - b_1^* \in [j'2^{-(k-1)} - j'2^{-(k-1)} + \delta] = ((2j - j')2^{-(k-1)}, (2j - j')2^{-(k-1)} + \delta] \). Thus we have \( b_2 \leq b_1 \leq j''2^{-(k-1)} + \delta \) for some \( j'' < 2^{-(k-1)} \). This means that

\[
B_1 + b_2 + \delta = 2j2^{-(k-1)} + b_2 + \delta \leq j''2^{-(k-1)} + 2\delta < (j' + 1)2^{-(k-1)} \leq 1,
\]

and thus \( b_2 + \delta \leq 1 - B_1 = B_2 \). This implies that \( b_1 = b_2 \) (it cannot be that \( b_2 = B_2 \) for \( \delta > 0 \), and thus \( b_2^* = b_1 + \delta \) wins in \( s \). This is a contradiction to the selection of \( b_2^* = B_2 < b_1^* \).

**Proposition 5** There is a zero-sum game with state-specific tie breaking, that has no subgame-perfect equilibrium.

**Proof.** Consider the game depicted in Fig. 3a. This is a zero-sum game where the white player is the maximizer. By the tie-breaking rule, white wins in \( x' \) if her budget is at least 0.5, and wins in \( y' \) if her budget is strictly above 0.5. In state \( x \), white needs to take at least one turn to win the game. Thus a budget of at least \( B_1(x) \geq 0.25 \) is sufficient. Similarly, in state \( y \) white has to take twice, and thus must have a budget of \( b_1(y) > 0.75 \) to win the game.

Clearly, if the initial budget partition in \( s_0 \) is \( B_1 \geq 0.5 + \epsilon \), then white can win by bidding \( b_1 = 0.25 + \epsilon/2 \): either she takes the turn and reaches \( x \) with \( 0.25 + \epsilon/2 \), or loses the turn and reaches to \( y \) with \( B_1 - b_2 \geq 0.5 + \epsilon - (0.25 + \epsilon/2) = 0.75 + \epsilon/2 > 0.75 \). Similarly, if \( B_1 \leq 0.5 - \epsilon \) then black can bid \( 0.25 + \epsilon/2 \) and force white to lose.

Now, suppose \( B_1 = B_2 = 0.5 \). We first show that there is no value in pure strategies. The outcome for any pair of deterministic bids \( (b_1, b_2) \) is displayed in Fig. 3b.
Let $b_1 \in [0, B_1]$. If $b_1 \leq 0.25$, then black can bid $b_2 = 0.25$, get to $y$ with $B_2(y) \geq 0.25$, and thus win. If $b_1 > 0.25$, then black can bid $b_2 = \frac{b_1 + 0.25}{2} > b_1$. Then we will get to $x$ with $B_1(x) < 0.25$. Thus white cannot guarantee the value 1.

On the other hand, let $b_2 \in [0, B_2]$. If $b_2 \leq 0.25$ then white can bid some $b_1 = 0.25 + \epsilon$ and get to $x$ with $B_1(x) \geq 0.25$ (and thus win). If $b_2 > 0.25$, then white can bid $b_1 < b_2$, and get to $y$ with $B_1(y) = B_1 + b_2 > 0.75$ (and win the game). Therefore black cannot guarantee a victory (i.e. a value of $-1$) either.

It is not hard to generalize the above argument to see that there is no value at all. Indeed, let $b_2 \in \Delta([0, B_2])$ be any mixed strategy of black. There must be an open interval $A = (0.25, 0.25 + \epsilon)$ s.t. $b_2$ assigns a probability of less than 0.01 to $A$. Then white can bid $b_1 = 0.25 + \epsilon/2$ and win w.p. of at least 0.99: the only bids of black that beat $b_1$ are in the range $(0.25, b_1)$, and thus white will win whether $b_2$ turns out to be at most 0.25, or at least $b_1$. Therefore the value is at least $0.99(1) + 0.01(-1) = 0.98$.

Similarly, let $b_1 \in \Delta([0, B_1])$ be some mixed strategy of white. Again there is some open interval $A' = (0.25, 0.25 + \epsilon)$ that is played by white w.p. less than 0.01. Then by playing $b_2 = 0.25 + \epsilon/2$ black can win w.p. of at least 0.99, and the value cannot be above $-0.98$. 

**Theorem 10.** Any binary tree game $G$ is monotone.

**Proof.** We prove by induction on the height of the game tree. For height 0 it is obvious. Note that for height 1 it is also easy, as there will be exactly one auction round, and the player with the higher budget can force her preferred outcome.

Let $\gamma$ be some PSPE in $\langle G, B_1, B_2 \rangle$, and suppose $B'_1 = B_1 + \Delta$ for some $\Delta > 0$. Suppose we are in the discrete model. Then w.l.o.g. it is sufficient to show for $\Delta = 1$. Similarly, in the
continuous model it is sufficient to show for $\Delta = \epsilon$, where $\epsilon < 2^{-\text{height}(G)-2}$, since for any $\epsilon' < \epsilon$, the game $\langle G, B_1 + \epsilon', B_2 - \epsilon' \rangle$ is either equivalent to $\langle G, B_1, B_2 \rangle$ or to $\langle G, B_1 + \epsilon, B_2 - \epsilon \rangle$. Thus assume $\Delta = \epsilon$ (where in the discrete case $\epsilon = 1$).

Intuitively, the proof shows that after the increase, either white can increase her winning bid by $\epsilon$ in order to discard the excessive budget, or she can lower her losing bid by $\epsilon$. In either case she can keep the same budget after the auction, and reach an outcome that is at least as good.

Denote by $s_l, s_r$ the left and right children of $s_0$, respectively. Each child is the root of another binary subgame, which we denote by $G_l, G_r$. Denote by $\gamma_l, \gamma_r$ the derived PSPEs in each subgame. We can assume w.l.o.g. that each of $\mu_l, \mu_r$ yields the highest possible utility for white under any budget. By induction, $G_l$ and $G_r$ are monotone, and thus $\gamma_l, \gamma_r$ are also monotone.

Case I: player 1 takes round $s_0$ in $\gamma$. That is, $b_1 \geq b_2$, and player 1 selects the next move. W.l.o.g. she selects $G_l$, which means that $t^* = \mu_l(B_1 - b_1) \succeq \mu_r(B_1 - b_1)$. Due to monotonicity of both sub-PSPEs, white is always weakly better off by reducing her bid down to the tie point, thus w.l.o.g. $b_1 = b_2 = b$. Since this is a PSPE in $s_0$, white cannot gain by reducing her bid and dropping the round, which either means that black would pick $s_1$ and (weakly) hurt white, or that black would also pick $s_1$ but without changing the utility of white. Thus either $t_r' \equiv \mu_r(B_2 - b) \succeq \mu_l(B_2 - b)$, and $t_r' \leq t_r$; or $t_r' = \mu_l(B_2 - b) \succeq 2 t_r$, and $t_r' = t_r$.

Case Ia: $B_2' \geq b + \epsilon$. We increase both bids in $s_0$ by $\epsilon$, so that $b_1' = b_2' = b' = b + \epsilon$. Note that white still takes the round, and selects $G_l$ with the same budget $B_2' - b' = B_1 - b - b'$. The only new deviation white has is bidding higher than $B_1$, but this is clearly pointless due to monotonicity of the subgames. Any deviation for black (i.e. bidding $b_2' > b'$) was also possible before by bidding $b_2 > b$, thus it cannot be beneficial. Note that in this case

$$
\mu'(B_1') = \mu'_l(B_1' - b_1') = \mu_l(B_1' - b_1') = \mu_l(B_1 - b_1) = t^*,
$$

so both players get the same utility as before the budget change.

Case Ib: $B_2' < b + \epsilon$. Since $b + \epsilon > B_2' + \epsilon = B_2 \geq b$, we have that $B_2 = b$, i.e., black cannot increase his bid. We set $b_1' = b_2' = B_2$. We set $b_1' = b_1 + \epsilon$, so that $B_2' - b_1' = B_1 - b_1$. Clearly black cannot deviate, and white can reach the same outcome as before the increase (or maybe to a better outcome with a different bid).

Case II: white (player 1) drops the first round in $\gamma$ under $B_1$. It is sufficient for black to take the round by $\epsilon$ (the minimal significant budget unit), so w.l.o.g. $b = b_2 = b_1 + \epsilon$. W.l.o.g. black selects $s_r$, which means $t^* = t^*_r \equiv \mu_r(B_2 - b_2) \succeq \mu_l(B_2 - b_2)$. This means that white either cannot or does not want to take the round. Now, consider $B_1' = B_1 + \epsilon$.

Case IIa: $b_1 > 0$. In this case both players can reduce their bids so that $b_2' = b_2 - \epsilon, b_1' = b_1 - \epsilon$. We get that black still wins and $B_1' + b_2' = B_1 + b_2$, so that black still selects $s_r$ and reaches $t^*_r$.

From here we may have a chain of responses where in each response either white increases her bid and selects $s_l$, or black increases and selects $s_r$ again. If eventually black wins with bid $b_2'' \geq b_2'$, then by monotonicity of $\mu_l, \mu_r(B_1' + b_2'') \succeq 1 \mu_r(B_1' + b_2'') = t^*_r$. If sequence ends with white wins ($b_1'' \geq b_2'' \geq b_2'$). Then we have that $t^* = \mu(B_1 - b_1') \succeq 1 \mu_r(B_1 + b_2') \succeq 1 t^*_r$.

This follows since if there is some PSPE better for white under a higher budget $B_1'$, the best PSPE for white under $B_1'$ must hold this as well.
Case IIb: \(b_1 = 0, b_2 = \epsilon\). In this case white cannot lower her bid. We define \(b'_1 = b'_2 = 0\), so that now white wins, and here remaining budget is \(B'_1 - b'_1 = B_1 + \epsilon = B_1 + b_2\). If there is no deviation of black, then clearly the outcome is at as good for white as before, since in particular she can select \(s_r\) and reach \(\mu_r(B_1 - b'_1) = \mu_r(B_1 + b_2) = t^*_r\). Assume, towards a contradiction, that black has a deviation. Such deviation is w.l.o.g. to bid \(b''_2 = \epsilon\), select \(s_l\). The outcome would be \(\mu_l(B'_2 - b''_2) = \mu_l(B_2 - 2\epsilon) \succ 2 \mu_r(B'_2 + b'_1) = \mu_r(B_2 - \epsilon) \succeq 2 \mu_l(B_2 - \epsilon)\),

where the last inequality is since black selects \(s_r\) with remaining budget \(B_2 - \epsilon = B_2 - b_2\). Finally, \(\mu_l(B_2 - 2\epsilon) \succ 2 \mu_l(B_2 - \epsilon)\) is a contradiction to the monotonicity of \(\mu_l\), so black does not have a deviation.

## B Uniqueness

Our next example shows that the Lower PSPE may not be unique, and that there may be other PSPEs that may not have the same properties.

**Proposition 16.** There is a generic bidding game with PSPEs that lead to different outcomes under the same budget.

**Proof.** Consider the game \(G_{\text{two}}\) in Fig. 4 with budgets \((0.5, 0.5)\). The first PSPE we describe starts with a bidding of \(b_1 = b_2 = 0\) in \(s_0\). White wins and remains with budget 0.5. With this budget she can win one more bidding phase and thus selects \(x\). The game ends in terminal \((5, 5)\) (marked with *) regardless of the bids in \(x\), since if the winner in \(x\) goes to \(y\) he will lose the next round and get 1. We note that black would bid strictly above 0, since otherwise white would bid 0 and still reach \((9, 1)\).

Next, consider a PSPE that starts with bids \(b_1 = b_2 = 0.5\). White wins the turn and remains with a budget of 0. If she selects \(x\), then black can play two turns in a row and reach \((1, 9)\). Thus white will select terminal \((2, 2)\) (marked with **). This is an equilibrium: black clearly cannot change the outcome. If white lowers her bid, then black wins and remains with a budget of 0. Then from the same consideration he would avoid selecting \(x\) (as white would then reach \((9, 1)\)). Thus white cannot strictly gain by lowering her bid below 0.5.

**Proposition 11.** For any \(k \in \mathbb{N}\) there is a binary bidding game \(G_k\), s.t. if \(G_k\) is played with a discrete budget of \(M < 2^k\), it has no Pareto-optimal PSPE.

---

Figure 4: The game \(G_{\text{two}}\). Note that utilities are generic.
Figure 5: The game $G_k$. Pareto-optimal terminals are marked with *.

Proof. If $M$ is small, a Pareto-efficient outcome may not be reachable in equilibrium under certain budget partitions. Consider the binary game $G_k$, described in Figure 5. In order to reach one of the Pareto-optimal leaves (marked with *), the same player must win $k$ times in a row. Suppose that the budget is $M < 2^k$. In any budget partition except when $B_2 = 0$, it holds for both players that $2^kB_i > B_i$. Since the budget of the loser doubles with each loss, no agent can win in all $k$ times. Thus the game must end in one of the “side” leaves, which are Pareto-dominated.

C Computational Issues

In this section we elaborate on the algorithmic considerations in computing the Lower PSPE.

Proposition 14. Let $G$ be a bidding game. Algorithm 2 computes the Lower PSPE $\gamma$ (more precisely, the mapping $\mu_\gamma$). Moreover, Algorithm 2 runs in time $|S| \cdot \text{poly}(|T|) = \text{poly}(|S|)$.

Proof. There are three differences between Algorithms 1 and 2: (a) the fast algorithm only computes the equilibrium in specific budget points ($z_{lr}$), rather than for any possible budget $B \in B$; (b) when black deviates, the fast algorithm increments the bid by a quantity larger than $\epsilon$ (and white raises by $\epsilon$ rather than 0); (c) after computing the equilibrium at all critical budget points, the fast algorithm “filters” some points out, and does not add them to the description of the PSPE.

We need to show that despite those differences we get the same equilibrium outcome, i.e., the same mapping $\mu : [0, 1] \rightarrow T(G)$.

We start by showing (b). Note that $B$ and $b_1$ are always a multiple of $2^{-\text{height}(G)}$. By Lemma 4, $B - b_2$ (which is the outcome in the slow algorithm) and $B - b_1 = B - (b_2 + \epsilon)$ belong to the same budget interval, and thus the outcome is the same.

The harder part is to show that by bidding $b'_2 = b_1 + \epsilon$ or $b_2 = \min\{B - F_{s_1}(a_1 - 1), F_{s_2}(a_2 + 1) - B\} - \epsilon$, the same equilibrium is reached. We first argue that $b_2 > b'_2 > b_1$ (so black indeed takes). Since $B - b_1 \leq F_{s_1}(a_1)$, and $F_{s_1}$ is strictly increasing, clearly $B - F_{s_1}(a_1 - 1) > b_1$ and thus $B - F_{s_1}(a_1 - 1) - \epsilon > b_1 + \epsilon = b'_2$ (since $b_1$ and every $F_{a}(a)$ are a multiple of $2^{-\text{height}(G)} > 2\epsilon$). Similarly, $F_{s_2}(a_2 + 1) > B + b_1 \geq F_{s_2}(a_2)$ by definition of $f_{s_2}$. Thus $F_{s_2}(a_2 + 1) - B - \epsilon > b_1$. We conclude that $b_2 = \min\{B - F_{s_1}(a_1 - 1), F_{s_2}(a_2 + 1) - B\} - \epsilon > b_1$.
Next, note that by bidding \( b_2' \) or \( b_2 \), black reaches the same budget interval \([F_{s_2}(a_2 - 1), F_{s_2}(a_2)]\) in \( s_2 \). The problem with just setting \( b_2 = F_{s_2}(a_2 + 1) - B - \epsilon \) is that this may prevent a further deviation by white, that would have been possible after a lower increment. That is, we may reach an equilibrium that is not the lowest-bids equilibrium. However, if white has a deviation after black, then we only care if she has a deviation that brings her to the next budget interval \( a_1 - 1 \) in \( s_1 \), otherwise black would deviate back to \( a_2 \) with another \( \epsilon \) increment. Therefore constraining \( b_2 < B - F_{s_1}(a_1 - 1) \) would prevent this problem.

We now turn to show (a). By induction, the budget between any consecutive critical points \([a_l - 1, a_l]\) yields the same outcome in \( s_l \), and likewise for \( s_r \). Fix some \( r, l \) and let \( B \in [z_{lr}, z_{lr+}] \), where \( z_{lr+} = \min\{z_{lr'}, z_{lr'} > z_{lr}\} \). We argue that whether white or black win in \( \langle s, z_{lr}\rangle \), they reach the same next state \( s' \) and with the same budget as from \( \langle s, B\rangle \). The proof is similar to the proof of Lemma\[4\] (there we always take the average between two multiples of \( 2^{-k-1} \), which yields a multiple of \( 2^{-k}\)).

It is left to show (c), i.e. that no information is lost when filtering the \( \binom{|T|}{2} \) critical points. Due to monotonicity, \( T \) is sorted, thus \( U \) contains the first index of every unique entry in \( T \). The filtering removes redundant parts of the equilibrium, as there are at most \(|T|\) points where the outcome changes. Thus we only consider these points.

Finally, we consider the complexity of the fast algorithm. After filtering, the size of the lists \( F_1, T^*_1, A_1^1, A_2^1 \) is at most \(|T|\) each. With every deviation, either \( a_1 \) or \( a_2 \) strictly decreases, and thus there can be at most \( 2|T| \) deviations. The heaviest part in computing a deviation is calculating \( f_{s_i} \). Since \( |F_{s_i}| \leq |T| \) a naïve computation is linear in \( |T| \). Therefore \( \text{Ascending - Auction}() \) runs in \( O(|T|^2) \). It is called at most \(|T|^2 \) times in every internal node, thus the total complexity is \( O(|S| \cdot |T|^4) = O(|S|^5). \)

\[C.1\] Compiling small DAGs from succinct valuations

Suppose we are given a sequential scrip auction, with some succinct representation of the value functions \( v_i \). While we can apply Proposition\[1\] to construct an equivalent bidding game, the naïve construction would yield a complete binary tree of height \( k \). In particular, the number of nodes \(|S|\) would be exponential in \( k \). In the worst case, there is not much we can do. For example, if buyers assign distinct utilities to exponentially many bundles of items, then any bidding game must contain this number of terminal states. Also, if \(|T|\) is fixed, but merely computing an optimal partition of \( K \) is NP-complete, then it must also be NP-hard to construct an equivalent succinct bidding game (if such a game even exists). This is since constructing the tree would in particular require us to efficiently generate all \(|T|\) outcomes, and one them is the optimal partition (we just check all of them).

**Additive valuations** We first solve a simple case, where valuations are additive, i.e., \( v_i(S) = \sum_{j \in S} v_i(j) \). Suppose that we sold the first 4 items out of \( k \), whose values to agent 1 are 2, 2, 4, and 3. Then the agent does not care if she currently has the first two items or just the third. If the other
agent is also indifferent between these partial partitions, then we can represent them with a single state.

We can generalize this observation using a dynamic programming technique, similar to the one used for Knapsack problems.

**Proposition 15.** Given an additive SSA $F$ with integer valuations, there is an equivalent bidding game with $|S| \leq M$ states, where $M = (v_1(K) + 1) \times (v_2(K) + 1)$. Further, such a bidding game can be constructed efficiently.

**Proof.** We assign a state $s_{m_1,m_2}$ for every pair $m_1 \in \{0, 1, \ldots, v_1(K)\}, m_2 \in \{0, 1, \ldots, v_2(K)\}$. The initial state is $s_{0,0}$. For every $j = 1, \ldots, k$, we go over all states in level $j - 1$. For connect each such state $s_{m_1,m_2}$ with two children: $s_{m_1+v_1(\tau(j)),m_2}$, and $s_{m_1,m_2+v_2(\tau(j))}$. That is, either agent 1 or agent 2 gets the $j$’th item (according to order $\tau$). We then add all new children to level $j$.

Note that any state at level $j$ corresponds to a partition of the first $j$ items, thus a state can only belong in one level. Finally, we identify the terminals $T$ with the last level $k$ (full partitions), and assign $u_i(s_{m_1,m_2}) = m_i$.

Using an implementation of Algorithm 2 in Matlab, we computed the PSPE prices for an SSA with $k$ identical items. When the initial budget is the same (0.5 for each player), we witness a clear pattern increasing prices over time. This pattern breaks when we start from different initial budget allocations.

**Weight-based games** A similar technique can be applied in games where the valuations are not additive, but still base on some additive notion of weight. We say that $v_i$ is *weight-based* if every item has a fixed weight $w_j$, and there is a function $f_i : \mathbb{N} \to \mathbb{R}$, s.t. $v_i(S) = f_i(\sum_{j \in S} w_j)$. Note that without further constraints any function is weight-based, as we can make sure that each bundle has a different total weight. However if weights are bounded then we get a succinct representation for a subclass of functions.

This idea can be further generalized. We say that $v_i$ is *multi-weight-based* if every item has a vector of fixed weights $(w^1_j, \ldots, w^q_j)$, and there is a function $f_i : \mathbb{N} \to \mathbb{R}$, s.t. $v_i(S) = f_i(\sum_{j \in S} w^1_j, \ldots, \sum_{j \in S} w^q_j)$. As a concrete example, suppose that the items are computing machines, each with some properties like storage, memory, bandwidth, etc. The value of a set of machines to a client depends only on the total storage, total memory, and total bandwidth, regardless of how these resources are allocated across among the machines.

Note that additive SSAs are a special case of multi-weight-based SSAs, where there are two dimensions. $f_1(S) = \sum_{j \in S} w^1_j, f_2(S) = \sum_{j \in S} w^2_j$. Note however that we allow $f_i$ to be a completely arbitrary function, and it does not even need to be monotone.

**Proposition 17.** Given a multi-weight-based SSA $F$, there is an equivalent bidding game with $|S| \leq M$ states, where $M = (w^1(K) + 1) \times \cdots \times (w^q(K) + 1)$. Further, such a bidding game can be constructed efficiently.

**Proof.** Construction is very similar to Proposition 15. We identify each state with a tuple $\langle m^1_i, m^2_i, \ldots, m^q_i, m_2^2 \rangle$, where $m^r_i$ now tracks the total dimension-$r$ weight of the bundle owned by agent $i$. We link each
state to its two children by adding all respective weights of item \( \tau(j) \) to the bundle owned by one of the agents.

Finally, in every terminal \( t = s(m_1^1, m_2^1, \ldots, m_1^q, m_2^q) \) (state that corresponds to a full allocation), we set \( u_1(t) = f_1(m_1^1, \ldots, m_1^q) \), \( u_2(t) = f_2(m_2^1, \ldots, m_2^q) \).

As a corollary from Propositions 14 and 17, a PSPE in multi-weight-based and in additive SSAs can be computed efficiently. That is, in time that is polynomial in the number of items and in the total weight/value.

**Voting with single peaked preferences** If voters have a strict preference order over all \( 2^k \) possible outcomes of the game, there cannot be a succinct representation, since we will need \( 2^k \) terminals. However consider the following scenario. Each voter as an “ideal” point \( t^*_i \in \{0,1\}^k \). The utility of any other outcome only depends on the Manhattan distance from \( t^*_i \), i.e. on the number of coordinates in which \( t, t^*_i \) differ. We may also assign a different weight to each coordinate, so that \( u_i(t) = -\sum_{j \leq k} w_{ij}|t^*_i(j) - t(j)| \).

This scenario has a simple mapping to an SSA of additive items: The value to \( i \) of every coordinate on which \( t^*_1, t^*_2 \) differ is \( w_{ij} \). If bidder \( i \) “buys” coordinate \( j \) she may set it to \( t^*_i(j) \). The value of every coordinate \( j \) on which \( t^*_1(j) = t^*_2(j) \) can be arbitrary. The bidders will not compete on these coordinates, as it is Pareto-optimal to set \( t(j) = t^*_1(j) \).

**Pathfinding** The items in a pathfinding problem over a graph, are pairs \( s = \langle \text{location}, \text{time} \rangle \). In a naïve representation, all pairs would be allocated, and the value of a bundle of pairs to an agent, would be the optimal legal path from source to target that this bundle contains. If our game states would each represent a subset of such pairs, we would quickly reach an exponential blowup. We modify the auction so that a state \( \langle \text{location}_1, \text{location}_2, \text{time} \rangle \) means that each agent \( i \) is in \( \text{location}_i \) at \( \text{time} \), after reaching there from the start. Thus we do not auction states (tuples) in an arbitrary order, but rather construct an alternative bidding game. We present an outline of the construction.

We start from \( s_0 = \langle \text{start}_1, \text{start}_2, 0 \rangle \), and link each state \( s = \langle \text{location}_1, \text{location}_2, \text{time} \rangle \) to two other states: the first is just a copy of \( s \), and the second is one of the legal moves from \( s \), for example \( \langle \text{location}_1', \text{location}_2', \text{time} + 1 \rangle \), where \( \text{location}_i' \) is either a neighbor of \( \text{location}_i \) or \( \text{location}_i \) itself, and it does not hold that \( \text{location}_1' = \text{location}_2' \). We keep enough copies of \( s \) to exhaust all legal moves (at most the number of locations, squared). The terminal states are those where \( \text{time} = \text{MaxTime} \), or both agents reached their targets. After the first agent reaches her target, we record her time. The negative utility to each agent in a terminal is her time.

Thus, in total we have at most \( \text{MaxTime}^2 \times |\text{locations}|^4 \) states including all copies.

**Bibliography**

Nejat Anbarci. Finite alternating-move arbitration schemes and the equal area solution. *Theory and Decision*, 61(1):21–50, 2006.
Junjik Bae, Eyal Beigman, Randall Berry, Michael L Honig, and Rakesh Vohra. Efficiency bounds for sequential resource allocation auctions. In CDC’07, pages 765–770, 2007.

Liad Blumrosen and Noam Nisan. Combinatorial auctions. In N. Nisan et al., editor, Algorithmic Game Theory. Cambridge University Press, 2007.

Zahy Bnaya, Roni Stern, Ariel Felner, Roie Zivan, and Steven Okamoto. Multi-agent path finding for self interested agents. In SoCS’13, pages 38–46, 2013.

Michel Le Breton and Vera Zaporozhets. Sequential legislative lobbying under political certainty. The Economic Journal, 120(543):281–312, 2010.

Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. Journal of Political Economy, 119(6):1061–1103, 2011.

Richard Cole, Vasilis Gkatzelis, and Gagan Goel. Mechanism design for fair division: Allocating divisible items without payments. In ACM-EC’13, pages 251–268, 2013.

Geoffroy De Clippel, Kfir Eliaz, and Brian Knight. On the selection of arbitrators. Technical report, Working Paper, Brown University, Department of Economics, 2012.

Mike Develin and Sam Payne. Discrete bidding games. the electronic journal of combinatorics, 17(1):R85, 2010.

Ian L Gale and Mark Stegeman. Sequential auctions of endogenously valued objects. Games and Economic Behavior, 36(1):74–103, 2001.

Zhiyi Huang, Nikhil R Devanur, and David Malec. Sequential auctions of identical items with budget-constrained bidders. arXiv preprint arXiv:1209.1698, 2012.

Bernardo A Huberman and Tad Hogg. Distributed computation as an economic system. The Journal of Economic Perspectives, 9(1):141–152, 1995.

Ehud Kalai and Meir Smorodinsky. Other solutions to Nash’s bargaining problem. Econometrica, pages 513–518, 1975.

Ehud Kalai. Proportional solutions to bargaining situations: interpersonal utility comparisons. Econometrica: Journal of the Econometric Society, pages 1623–1630, 1977.

Dean Lacy and Emerson MS Niou. A problem with referendums. Journal of Theoretical Politics, 12(1):5–31, 2000.

Jérôme Lang and Lirong Xia. Sequential composition of voting rules in multi-issue domains. Mathematical social sciences, 57(3):304–324, 2009.

Andrew J Lazarus, Daniel E Loeb, James G Propp, and Daniel Ullman. Richman games. Games of no chance, 29:439–449, 1996.
Andrew J Lazarus, Daniel E Loeb, James G Propp, Walter R Stromquist, and Daniel H Ullman. Combinatorial games under auction play. *Games and Economic Behavior*, 27(2):229–264, 1999.

Renato Paes Leme, Vasilis Syrgkanis, and Éva Tardos. Sequential auctions and externalities. In *SODA’12*, pages 869–886, 2012.

John F Nash. The bargaining problem. *Econometrica*, pages 155–162, 1950.

Yuval Peres, Oded Schramm, Scott Sheffield, and David B Wilson. Random-turn hex and other selection games. *American Mathematical Monthly*, 114(5):373–387, 2007.

Robert Powell. Sequential, nonzero-sum blotto: Allocating defensive resources prior to attack. *Games and Economic Behavior*, 67(2):611–615, 2009.

Gustavo E Rodriguez. Sequential auctions with multi-unit demands. *The BE Journal of Theoretical Economics*, 9(1), 2009.

Robert W Rosenthal. Games of perfect information, predatory pricing and the chain-store paradox. *Journal of Economic Theory*, 25(1):92–100, 1981.

Guni Sharon, Roni Stern, Meir Goldenberg, and Ariel Felner. The increasing cost tree search for optimal multi-agent pathfinding. *Artificial Intelligence*, 195:470–495, 2012.

David Silver. Cooperative pathfinding. In *AIIDE*, pages 117–122, 2005.

Yves Sprumont. Intermediate preferences and rawlsian arbitration rules. *Social Choice and Welfare*, 10(1):1–15, 1993.

Eyal Winter. Negotiations in multi-issue committees. *Journal of Public Economics*, 65(3):323–342, 1997.

David Zetland. An auction market for journal articles. *Available at SSRN 861067*, 2005.
Algorithm 1: Find-Lower-PSPE(G)

Set \( \epsilon \leftarrow 2^{-\text{height}(G) - 2} \);
\( \mathcal{B} \leftarrow \{ c \cdot \epsilon : c \in \{0, 1, \ldots, \lceil 1/\epsilon \rceil \} \} \);
Initialize tables \( A_1, A_2, T^* \) of size \(|S| \times |\mathcal{B}|\);
In every leaf \( t \), assign \( T^*(t, B) = t \) for all \( B \in \mathcal{B} \);
for every node \( s \) in post-order do

Let \( s_l, s_r \) be the two children of \( s \);
for every budget partition \( B_1 \in \mathcal{B} \) do

\( b_1 \leftarrow 0; b_2 \leftarrow 0 \); // Initialize both bids
repeat

\( t^* \leftarrow \text{Get-outcome}(G, T^*, s, B_1, B_2, b_1, b_2) \);
\( i \leftarrow \) player with higher bid;
// Try a deviation by the loser \(-i\):
if \( -i = 1 \) then
\( b'_1 \leftarrow \min\{b_2, B_1\} \)
else
\( b'_2 \leftarrow \min\{b_1 + \epsilon, B_2\} \)
\( t' \leftarrow \text{Get-outcome}(G, T^*, s, B_1, B_2, b'_i, b_i) \);
if \( t^* \succeq -i t' \) then
\( \text{Break; } \) // I.e., continue as long as the loser can deviate
\( b_{-i} \leftarrow b'_i \);
until;
// Write down the equilibrium strategies for \( s \):
\( \text{succ}(s) = \arg\max_{s' \in \{s_l, s_r\}} u_i(T^*(s', B_i - b_i)) \);
\( \text{other-child}(s) = \arg\max_{s' \in \{s_l, s_r\}} u_{-i}(T^*(s', B_i + b_{-i})) \);
\( T^*(s, B_1) \leftarrow T^*(\text{succ}(s), B_i - b_i) \);
\( A_i(s, B_1) \leftarrow (b_i, \text{succ}(s)) \);
\( A_{-i}(s, B_1) \leftarrow (b_{-i}, \text{other-child}(s)) \);

return \( \gamma \leftarrow (A_1, A_2) \);

Function Get-Outcome(G, T*, s, B_1, B_2, b_1, b_2)

\( i \leftarrow \) player with higher bid;
\( \text{succ}(s) = \arg\max_{s' \in \{s_l, s_r\}} u_i(T^*(s', B_i - b_i)) \);
\( t \leftarrow T^*(\text{succ}(s), B_i - b_i) \);
return \( t \)
Algorithm 2: **FIND-PSPE-FAST(G)**

1. **Initialize** \( F, T, \overline{A}_1, \overline{A}_2 \);
2. // See description of **Initialize()** for explanation of variable roles.
3. **for** every leaf \( t \) **do**
   4. // In the leaves there is only one budget interval, which is all the range \([0,1]\).
   5. \( F_t \leftarrow (0,1) \); // An array of length two
   6. \( T^*_t \leftarrow (t,t) \);
7. // Traverse nodes bottom up:
8. **for** every node \( s \) in post-order **do**
9. Let \( s_l, s_r \) be the two children of \( s \);
10. For every \( a_l \leq |F_{s_l}|, a_r \leq |F_{s_r}| \) set \( z_{lr} = \frac{F_{s_l}(a_l)+F_{s_r}(a_r)}{2} \);
11. Sort \( \{z_{lr}\}_{lr} \) in increasing order;
12. // \( z_{lr} \) are the only ’’critical points’’ of the budget in state \( s \). The equilibrium bids for any point in the interval \([z_{lr}−, z_{lr}]\) reach the same budget interval in the next state (either \( s_l \) or \( s_r \)).
13. Initialize global empty lists \( F, T, \overline{A}_1, \overline{A}_2 \);
14. **for** every \( i, r \) (in increasing order of \( z_{lr} \)) **do**
15. Set \( B = z_{lr} \); // Budget of player 1
16. // Compute the lowest equilibrium of the current step
17. for budget \( B_1 = B \)
18. \( \langle s_1, a_1 \rangle, \langle s_2, a_2 \rangle, t \) \leftarrow Ascending-Auction\( (G, s, B) \);
19. Append\( (F, B) \); Append\( (T, t) \);
20. Append\( (\overline{A}_1, \langle s_1, a_1 \rangle) \); Append\( (\overline{A}_2, \langle s_2, a_2 \rangle) \);
21. \( U \leftarrow T(1) \);
22. for \( \ell = 2, 3, \ldots, |T| \) **do**
23. if \( T(\ell) \neq T(\ell - 1) \) then
24. Append\( (U, \ell) \);
25. // Write down the equilibrium strategies in \( s \):
26. \( F_s \leftarrow Filter(T, U) \); \( T^*_s \leftarrow Filter(T, U) \);
27. \( A^1_s \leftarrow Filter(\overline{A}_1, U) \); \( A^2_s \leftarrow Filter(\overline{A}_2, U) \);
28. return \( \gamma \leftarrow \langle F, T^*, A^1, A^2 \rangle \);
**Function** Initialize\((F, T, A^1, A^2)\)

for every node \(s \in S\): do

- Initialize global empty lists \(F_s, T_s^*, A^1_s, A^2_s\);

  // Intuitively, \(F_s\) contains all budget cutoff points. For every \(a \in \{1, 2, \ldots, |F_s|\}\), all budgets \(B_1 \in [F_s(a - 1), F_s(a)]\) are equivalent, in the sense that the outcome from \(\langle s, B_1 \rangle\) will be the same. We use \(f_s : [0, 1] \rightarrow \{1, 2, \ldots, |F_s|\}\) as the inverse of \(F_s\), i.e. \(f_s(B) = \max\{a : F_s(a) \leq B\}\). \(a\) is the index of the budget interval that contains \(B\).

  // \(T_s^*(a) \in T\) is the outcome that will be reached from state \(\langle s, F_s(a) \rangle\).

  // \(A^i_s\) is the strategy of \(i\), where \(A^i_s(a) = \langle s', a' \rangle\). The equilibrium bid in \(s\) for any budget \(B\) can be derived by subtracting \(B - F^i_s(a')\), where \(s' \in g(s)\) and \(a'\) is the budget index in \(s'\).
Function Ascending-Auction($G, s, B$)

\[
\begin{align*}
    b_1, b_2 & \leftarrow 0; \quad \text{(// Initialize bids)} \\
    a_l & \leftarrow f_{s_l}(B); \\
    a_r & \leftarrow f_{s_r}(B); \\
    s_1 & \leftarrow \arg\max_{w \in \{l, r\}} u_1(T^*(s_w, a_w)); \\
    s_2 & \leftarrow \arg\max_{w \in \{l, r\}} u_2(T^*(s_w, a_w)); \\
    \text{if } s_1 == s_2 \text{ then} \\
    & \quad \text{// White wins with bid 0, and plays } s_1. \text{ Reaches budget interval indexed by } a' \text{ (either } a_l \text{ or } a_r). \\
    & \quad s' \leftarrow s_1; \\
    & \quad a' \leftarrow f_{s'}(B); \\
    & \quad t^* \leftarrow T^*_{s_1}(a_1); \\
    & \quad \text{return } \langle\langle s', a' \rangle, \langle s', a' \rangle, t^*\rangle; \\
\end{align*}
\]

repeat \(//\) Compute outcome under current bids \(b_1, b_2\): \(a_1 \leftarrow f_{s_1}(B - b_1); \\
\)
\[
    t_1 \leftarrow T^*_{s_1}(a_1); \quad \text{// current outcome} \\
    a_2 \leftarrow f_{s_2}(B + b_1); \\
    b'_1 \leftarrow b_1 + \epsilon; \quad \text{// try a deviation of black} \\
    a'_2 \leftarrow f_{s_2}(B + b'_1); t'_2 \leftarrow T^*_{s_2}(a'_2); \quad \text{// outcome if black takes} \\
\]
\[
\text{if } b'_2 > 1 - B \text{ or } t'_1 > t'_2 \text{ then} \\
\quad \text{// Black cannot or would not raise. In this budget interval (above } B \text{ and below the next significant point), white wins. The bid is set so that } B - b \text{ falls in the budget interval indexed by } a_1 \text{ (in } s_1). \\
\quad \text{return } \langle\langle s_1, a_1 \rangle, \langle s_2, a_2 \rangle, t_1\rangle; \\
\]
\[
    b_2 \leftarrow \min\{B - F_{s_1}(a_1 - 1), F_{s_2}(a_2 + 1) - B\} - \epsilon; \\
    b''_1 \leftarrow b_2 + \epsilon; \quad \text{// try a deviation of white} \\
    a''_1 \leftarrow f_{s_1}(B - b''_1); \\
    t''_1 \leftarrow T^*_{s_1}(a''_1); \\
\]
\[
\text{if } b''_1 > B \text{ or } t''_1 > t'_1 \text{ then} \\
\quad \text{// White cannot or would not raise. In this budget interval, black wins and plays } s_2. \text{ Reaches budget interval indexed by } a'_2. \\
\quad \text{return } \langle\langle s_1, a_1 \rangle, \langle s_2, a'_2 \rangle, t'_2\rangle; \\
\]
\[
    b_1 \leftarrow b''_1; \quad \text{// equals to } b_2 + \epsilon \\
\]
\text{until}:
Function Filter($L, U$)

Initialize $L$ as an empty list;

for $j = 1, \ldots, |U|$ do

    $\ell \leftarrow U(j)$;
    Append($L, T(\ell)$);

return $L$