Obvious Independence of Clones

Ratip Emin Berker, Sílvia Casacuberta, Christopher Ong, Isaac Robinson

April 28, 2022

Abstract

The Independence of Clones (IoC) criterion for social choice functions (voting rules) measures a social choice function's robustness to strategic nomination. However, prior literature has established empirically that individuals cannot always recognize whether or not a mechanism is strategy proof and may still perform costly, distortionary misreports even in strategy-proof settings. The intersection of these issues motivates the search for mechanisms which are *Obviously Independent of Clones (OIoC)*: where strategic nomination or strategic exiting of clones obviously have no effect on the outcome of the election. We examine three IoC ranked-choice voting mechanisms and the pre-existing proofs that they are independent of clones: Single Transferable Vote (STV), Ranked Pairs, and Schulze. We construct a formal definition of a voting system being Obviously Independent of Clones based off of a reduction to a *clocked election* by considering a *myopic agent*. Finally, we show that STV and Ranked Pairs are OIoC, whereas we prove an impossibility result for Schulze showing that this voting system is *not* OIoC.

1 Introduction

On November 6th, 1934, Oregonians took to the polls to elect their state's 28th governor. Beforehand, in a aggressively contested race, State Senator Joe E. Dun narrowly defeated Peter Zimmerman. When it came time to vote, residents faced a tough choice of candidates. The following is a summary of the votes each candidate received [oSN21]:

Charles H. Martin	Peter Zimmerman	Joe E. Dunne
116,677	95,519	86,923

After losing in the Republican primary, Peter Zimmerman, who had also previously served as a Republican senator for the state, decided to run in the general election as an independent. Peter Zimmerman and Joe E. Dunne collectively won nearly 60% of the vote, but as individuals, neither managed to edge out the Democratic candidate. The above motivates the following question: *How can we make sure that similar candidates in an election do not split the vote, leading to a candidate who is least preferred by the majority of voters winning the election*?

This so-called "spoilage" effect motivated T. N. Tideman to formulate a criterion known as *Independence of Clones* (IoC) in 1987, which ensures that the addition of a candidate with similar policy inclinations will not spoil the election [Tid87]. Intuitively, we would like to ensure the property that the winner of an election should not change due to the addition of a non-winning candidate who is very similar to the winner. Tideman then goes on to introduce a new voting rule, *Ranked Pairs voting*, that satisfies the IoC criterion. Since then, other pre-existing voting rules have been shown to be IoC, such as Single Transferable Vote (STV). Markus Schulze introduced in 2010 the so-called *Schulze voting method*, which he proved to be IoC [Sch11]. The Australian House of Representatives has been using STV since the early 20th century [BL07]. Schulze voting has been adopted for all referendums in the city of Silla, Spain as well as in dozens of organizations including the IRC Council for the Operating System Ubuntu [Sch18].

The solution to the spoilage effect then becomes to switch all voting systems to ones that are IoC. However, it is not immediately clear that the average voter or candidate would be convinced that, in switching to one of these methods, the election systems becomes impervious to spoilage effect. As such, even after switching to an IoC rule, an ill-informed or non-believing candidate may still drop out of the election early out of an abundance of caution. Moreover, even a savvy candidate may recognize that the average voter may not be willing or able to follow the proof for why a voting method is independent of clones. As a result, the savvy candidate may still drop out in fear of being blamed by their voter base for a lost election, and thus alienated and blocked from holding future elected office.

This motivates examining the *obviousness* of a mechanism: i.e., how easy it is to "expose" a certain property of a mechanism so that it is easy to understand that the property is indeed satisfied. Shengwu Li first defined the concept of *Obvious Strategy-Proofness*, which applies this idea to the property of strategy-proofness (SP) [Li17]. The paper is motivated by the fact that oftentimes, the benefits of a property of a mechanism are only accrued when agents understand said property and believe the mechanism satisfies this property. For example, if agents misreport their true rankings for the Gale-Shapley matching algorithms, then the results might be suboptimal. But this understanding often requires mathematical maturity, time investment, and/or a series of "what if" reasoning steps that are difficult for what Li defines as a *cognitively limited agent*. This is understood as an agent who is unable or unwilling to perform contingent reasoning.

Namely, in order for an agent to verify a random given mechanism is SP, they may need to reason through what the other agents would do conditioning on their own action. They would then need to reason about how they could best respond to their opponents' best response and so on. In an OSP mechanism, it is clear at each step which decision an agent should make – regardless of what the other agent will play at any point down the line.

Intuitively, the IoC criterion requires a similar type of "what if" reasoning. This time, we require agents to walk through the steps of a voting rule pretending as if there is a clone, and them comparing that to the base outcome without a clone. Inspired by Li's work, we move to the setting of voting mechanisms and introduce the original concept of *Obvious Independence of Clones (OIoC)*. For it, we borrow from Li's comparison of two mechanisms that when given private values always return the same allocation and the same payment – the second price sealed bid auctions, which are *not* OSP, and ascending clock auctions, which are OSP. We use this comparison as motivation to define a *clocked election*, which is our adaptation of the personal clock auction to the setting of voting rules based on inducing a notion of *rounds*. The definition of clocked election employs our notion of a *myopic agent*, who acts as a witness during the execution of the protocol.

1.1 Overview

Sections 2 through 4 are dedicated to establishing necessary definitions. Section 2 defines the interprofile criterion of Independence of Clones as defined in pre-existing literature and illustrates it through several examples of mechanisms which are either IoC or not. Section 3 introduces two graph-based algorithms, Ranked Pairs and Schulze, which are both IoC, which will become the objects of our proofs. Section 4 introduces Shengwu Li's definition for Obvious Strategy-Proofness, which provides the inspiration for Obvious Independence of Clones.

Section 5 introduces original definitions for a clocked election and a myopic agent. Using these definitions, we formally define a test for determining whether or not a mechanism is Obviously Independent of Clones, based on a reduction to a clocked election with a myopic agent as a witness. We then use this test to prove that STV and Ranked Pairs both fit our criterion and are Obviously Independent of Clones. Section 6 establishes an impossibility result, showing that there is no way of transforming

Schulze into a clocked election and therefore, Schulze cannot be obviously independent of clones. We conclude with implications for future research. Throughout the paper, we provide concrete examples of the different algorithms. In particular, we show with examples how the addition of clones affects the executions of STV, Ranked Pairs, and the Schulze method. The code for our simulations can be found at this GitHub repository.

2 Independence of Clones

Definition 2.1 (Set of clones [Tid87]). A proper subset of two or more candidates, K, is a *set of clones* if no voter ranks any candidate outside of K as either tied with any element of K or between any two elements of K.[1]

Intuitively, this captures the idea that the voters' rankings are consistent with the hypothesis that the candidates in K are close to one another in some shared perceptual space. For example, in the following voter profile σ, candidates B and C are clones:

2 Voters	4 Voters	7 Voters	9 Voters	13 Voters
A	A	**B**	**C**	D
B	**B**	**C**	**B**	**B**
C	**C**	D	D	**C**
D	D	A	A	A

Definition 2.2 (Independence of clones [Tid87]). A voting rule is *independent of clones (IoC)* if and only if the following two conditions are met when clones are on the ballot:

1. A candidate that is a member of a set of clones wins if and only if some member of that set of clones wins after a member of the set is eliminated from the ballot.

2. A candidate that is *not* a member of a set of clones wins if and only if that candidate wins after any clone is eliminated from the ballot.

In other words, deleting one of the clones must not increase or decrease the winning chance of any candidate not in the set of clones. This is a desirable property in voting rules, since it imposes that the winner must not change due to the addition of a non-winning candidate who is similar to a candidate already present. This prevents candidates influencing the election by nominating new candidates similar to them or their opponents.

Given this definition, there is a trivial composition that can make any non-IoC voting rule into one that is IoC. Namely, suppose we first impose a *primary* between all clone candidates. This allows us to condense a set of clones into a single clone, who will thus not be affected by clones in the general election. In fact, we see this system used in many nations that use *plurality voting*, which we will see in the next section is not IoC.

The above formulation poses several issues. First, it requires a *strict* primary process, where losing clones are disallowed from running in the general election. Moreover, it requires that primaries be run between sets of clones, which means clones need to be identified priors to voters voting.

Such a formulation also lacks a notion of "coherence", as shown by the following topological argument against primary voting as a means to insure independence of clones: the domain of anonymous and homogenous voting rule can be interpreted as the space spanned by the vectors composed of the number

[1] In this paper, we only allow voters to provide strict ranking of candidates as a part of their ballot preference profiles.

of votes for each candidate. Projecting this set of vectors onto the rational subspace of the unit simplex, gives us a topological representation of the intuitive statement that the results can be expressed as a function of the proportion of votes each candidate gets. In this domain, performing a primary where we search for clones and dominant candidates sharply divides the region into two sections: one where there is a clone and one where there is not. These ragged edges undermine the smoothness of our voting rule function, which could be interpreted as a lack of coherence in such two-part voting rules.

Next, we examine whether some of the most common voting rules satisfy IoC.

2.1 Plurality Voting

In plurality voting, the winner is given by the candidate who receives the most first-place votes. As we demonstrated with a historical example in the Introduction, we can readily see that plurality does not satisfy IoC. Consider the following voting profile for candidates A and B:

62 Voters	38 Voters
A	B
B	A

In this case, the winner of the election is A. After introducing a clone A_2 of A, we might obtain the following voting profile:

28 Voters	34 Voters	38 Voters
A	A_2	B
A_2	A	A_2
B	B	A

With the introduction of clone A_2, we see that the winner is now B, who is not part of the set of clones. Thus, the clone of A has negatively impacted A.

2.2 Borda Count

In Borda count, each voter gives $m - k$ points to the alternative placed in the k-th position, where m is the number of alternatives. We provide an example to show that Borda does not satisfy IoC either. Consider the following voting profile for candidates A and B:

62 Voters	38 Voters
A	B
B	A

In this case, candidate A receives 62 Borda points, whereas candidate B receives 38 Borda points. Thus, candidate A wins the election. Next we introduce a clone of B, obtaining the following voting profile:

62 Voters	38 Voters
A	B
B	B_2
B_2	B

Now candidate A receives 124 Borda points, B receives 138 Borda points, and clone B_2 receives 38 Borda points. Hence, candidate B now becomes the winner. Unlike the plurality example, this is a case where the clone of B has positively impacted B. Therefore, the Borda Count is not Independent of Clones. More generally, we can prove the following statement:

Lemma 2.3. *Any preference profile in which candidate A wins against candidate B using the Borda count (with an arbitarily large margin) can be turned into a preference profile in which B wins by adding clones of B, given that there is at least one voter that prefers B over A.*

Proof. See Appendix A.1. $\square$

2.3 STV

In Single Transferable Vote (STV), each voter submits a strict ranking of the candidates. At each round, if no candidate has majority, the candidate with the fewest vote is eliminated, and these votes are transferred to the candidate named second in the ranking [Tid87]. Unlike plurality and Borda count, STV *is* Independent of Clones. To show this, we adapt the argument from [Tid87].

Lemma 2.4. *STV is Independent of Clones.*

Proof. See Appendix A.2. $\square$

For examples demonstrating that STV is IoC, see Appendix B.1.

3 Graph-based algorithms for IoC

3.1 Ranked Pairs

The Ranked Pairs (RP) voting system was developed in 1987 by Nicolaus Tideman specifically to satisfy the Independence of Clones criterion [Tid87]. While Tideman's original formulation takes in an aggregate of voter strict rankings over a set of alternatives and returns a complete ranking of alternatives, for the purposes of this paper, we will be dealing with a simplified version of Ranked Pairs which takes in the same ranking and simply returns the winner of the election. On input of a voter profile σ on voters N and candidates A, the Ranked Pairs algorithm is as follows:[2]

1. Based on the voter profile, construct the so-called *majority matrix* M, whose ij component is equal to the difference between the number of voters who rank A_i ahead of A_j and the number of voters who rank A_j ahead of A_i. Observe that M is an anti-symmetric matrix, and thus $M[A_i, A_j] = -M[A_j, A_i]$.

2. Sort the positive majorities in M in decreasing order of magnitude. Let $\mathcal{L}$ be the list containing the pairs (A_i, A_j) in this order.

3. Construct a directed graph G by iterating through the pairs in $\mathcal{L}$ in order. For each pair of candidates (A_i, A_j), add a directed edge e_{ij} from node i to j if $M[A_i, A_j] > M[A_j, A_i]$, and from j to i otherwise. If e_{ij} creates a cycle in G, remove it.

4. The winner of the election is the candidate that corresponds to the source node of G.

Tideman shows that the source of the graph exists and is unique [Tid87]. To illustrate, consider an example run of the Ranked Pairs algorithm in Appendix B.2.

[2]To bound the scope of this paper, in this section and throughout this paper, we largely ignore tie-breaking. Theoretically we handle this by assuming that ties do not occur. Empirically, in our simulations, we break ties deterministically by alphabetical order.

3.2 Schulze Method

The Schulze voting system was introduced by Markus Schulze in 2010 as a new monotone, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method [Sch11]. On input a voter profile σ on voters N and candidates A, the Schulze algorithm is as follows:

1. Based on the voter profile, construct the so-called *pair-wise matrix* P, whose ij component is equal to the number of voters who rank A_i ahead of A_j.

2. Construct a graph G where node i corresponds to candidate A_i as follows. For each pair of candidates (A_i, A_j), add a directed edge e_{ij} from node i to j if $A_i > A_j$, and from j to i otherwise. The weight of the edge, denoted $w(e_{ij})$, is equal to $\max P[A_i], P[A_j]$.

3. For each path in G from candidate A_i to A_j, the *strength* of the path is defined as the minimum of the weights of all edges in the path. For each ij, we want to maximize the weight of the minimum-weight edge in the path from A_i to A_j. We can compute this, for example, with the Floyd-Warshall algorithm [Sch11].

4. After computing the strengths of the strongest paths, we construct the *strength matrix* S, whose ij component is equal to the strength of the strongest path from A_i to A_j.

5. Now we need to examine the entirety of matrix S. For each pair of candidates (A_i, A_j), compare $S[A_i, A_j]$ to $S[A_j, A_i]$. If $S[A_i, A_j] > S[A_j, A_i]$, then $A_i > A_j$ in the final ranking, and viceversa. After all pair-wise comparisons, the final ranking of candidates has been determined. The winner of Schulze method corresponds to the top-ranked candidate.

As proven by Schulze, a winner always exists and is unique. Comparing Schulze to Ranked Pairs, we remark that we can construct the pair-wise matrix from the majority matrix and vice-versa. While the original paper uses an implementation of Schulze that runs in time $O(m^3)$, where m corresponds to the number of candidates, faster implementations of the algorithm have been proposed in [SVWX21]. The bottle-neck corresponds to the runtime required to compute the strengths of the strongest paths between all candidates. We remark that this step corresponds to the *widest path problem* in graph theory.

An important aspect to remark about Schulze is the fact the final ranking is unique, *no matter the order in which we compare the candidates in the strength matrix.* This point will become relevant when we examine our proposed notion of Obvious Independence of Clones.

For an example of Schulze's algorithm, see Appendix B.3.

4 Obvious Strategy-Proofness

The origin behind our inquiry on the property of "obviousness" traces back to the introduction of *Obviously Strategy-Proof Mechanisms* (OSP) by Shengwu Li [Li17]. As outlined in the introduction, "exposing" the strategy-proofness of the mechanism is important for ensuring that agents rank truthfully, since otherwise the outputs might be suboptimal (e.g., when running the Gale-Shapley matching mechanism). Li captures this notion with the following two definitions:

Definition 4.1 (Obviously dominant [Li17])**.** A strategy is *obviously dominant* if, for any deviation, at any information set where both strategies first diverge, the best outcome under the deviation is no better than the worst outcome under the dominant strategy.

Definition 4.2 (Obviously strategy-proof (OSP) [Li17])**.** A mechanism is *obviously strategy-proof (OSP)* if it has an equilibrium in obviously dominant strategies.

Li also provides the behavioral interpretation behind this definition: a strategy is obviously dominant if and only if a cognitively limited agent can recognize it as weakly dominant. A *cognitively limited agent* is understood as someone who is *unable to engage in contingent reasoning*. This differs from our definition of *myopic agent* in Section 5, where we need to establish a formal definition of this agent to require some more restrictive properties. The notion of a cognitively limited agent is best described with an example, which is presented in [Li17]. Suppose that Agent 1 has preferences $A \succ B \succ C \succ D$. In mechanism (i), it is a weakly dominant strategy for 1 to play L, whereas in mechanism (iii) it is *not* a weakly dominant strategy for Agent 1 to play L. However, if Agent 1 cannot engage in contingent reasoning (i.e., cannot think through hypothetical scenarios), then he cannot distinguish between (i) and (ii). As coined by Li, this means that (i) and (ii) are *1-indistinguishable*.

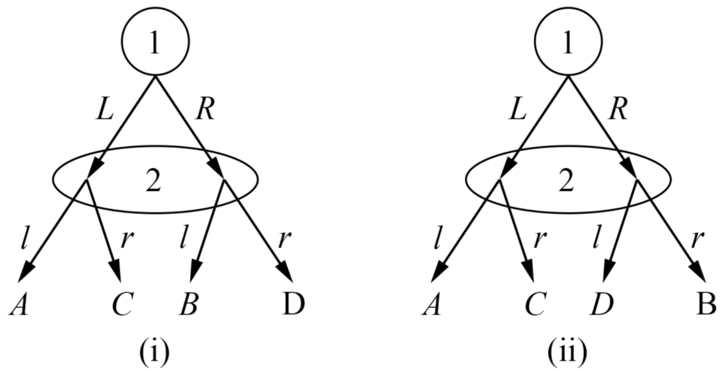


Figure 1: Agent 1 is unable to distinguish between mechanisms 1 and 2 (Fig. 1 in [Li17]).

An illustrative example of the difference between strategy-proofness and Obvious Strategy-Proofness (OSP) is in the comparative analysis of the ascending clock auction and the Second Price Sealed Bid Auction. Previous literature has established how both auctions are efficient, strategy-proof mechanisms, and how both auctions are functionally equivalent [Li17]. However, under Li's definition of Obvious Strategy-Proofness, only the ascending clock auction is OSP, while SPSB is not.

To illustrate this difference, consider the example in Appendix B.4.

5 Obvious Independence of Clones (OIoC)

Motivated by Li's formalization of the notion of *Obviously Strategy-Proof* (OSP) mechanisms, we adapt some of the key definitions and insights from his framework to the setting of voting rules and to the property of independence of clones. Li introduces three different notions that allow him to prove that certain notions are equivalent to OSP: i-indistinguishability [Li17, Def. 9, Thm. 1], bilateral commitments [Li17, Def. 12, Thm. 2], and personal-clock auctions [Li17, Def. 15, Thm. 3].

Myopic agents. As explained in Section 4, i-indistinguishability boils down to a cognitively-limited agent being incapable of engaging in contingent reasoning. For our purposes, we need to extend this definition to what we call a *myopic agent*. We first introduce the formal definition; the intuition will become clear in the explanation following Definitions 5.2 and 5.3. Because our intuitive explanation for our definition of OIoC introduces the role of a cheater (which is inspired by the role of an *adversary* in cryptographic protocols), we view the myopic agent as a *witness* during the clocked election protocol (Definition 5.2). This explains why the Definition 5.1 has a cryptographic flavor (more concretely, we take inspiration from the notion of *simulation-based security* [Lin17]). We also draw inspiration from the definition of Turing Machines in complexity theory.

Definition 5.1 (Myopic agent)**.** Let $\mathcal{F}$ be a functionality and let π be a one-party protocol for computing $\mathcal{F}$. The view of Agent i during the execution of π on input x is denoted by $\mathsf{view}_i^\pi(x)$ and it equals the *transcript* $(x, r_1^i, \ldots, r_t^i; m_1, \ldots, m_t)$, where r_j^i equals the contents of the ith party's internal computations at the j-th step of the execution, and m_j corresponds to the state of the functionality at the j-th step of its execution. Then, we say that an agent is *myopic* if it satisfies the following properties:

1. For any arbitrary agent A of possibly unbounded computational complexity who is executing π, the myopic agent M must concurrently receive the full transcript of the execution of π, including any internal side computations performed by agent A. That is, i.e., $\mathsf{view}_A^\pi(x) = \mathsf{view}_M^\pi(x)$.

2. The myopic agent M has a tape of infinite length on which it can perform `read` and `write` operations. This means that M is able to read the transcript generated during the execution of the protocol, while being unable to read any unaccessed data.

3. M is incapable of realizing any further functionality; in particular, it cannot realize *any* sort of arithmetic computation.

4. M is incapable of engaging in contingent reasoning.

For example, by Condition 3 we deduce that M is unable to perform any of the following computations: $+, \times, >, <, =, \max, \min$, pair-wise comparisons.

Clocked Election. After careful examination, we found that the idea of the personal clock auction was the most natural to adapt to our setting of voting rules. As discussed in Section 4, while the sealed second price auction and the clock auction are identical in terms of their strategy profile and outcomes, the clock auction is OSP, while the sealed second price auction is not, demonstrating that two different implementations with the same reduced normal form game can differ in whether they are OSP or not. In fact, Li further generalizes this notion by proving that a game has an OSP implementation if and only if it can be reduced to what he calls a "personal-clock auction" [Li17, Thm. 3]. The definition for "personal-clock auction" is broadly a generalization of the clock auction in which every player has a set of 'quitting' actions at each round [Li17, Def. 15]. Motivated by this theorem, we provide the following definition:

Definition 5.2 (Clocked Election)**.** A Clocked Election for a voting rule f is a protocol CE that takes as input a preference profile $\boldsymbol{\sigma}$ for voters N and candidates A and iteratively fills an ordered list of 'eliminated' candidates F (where F_i is the state of the list after i iterations and hence F_{m-1} is the final list) such that the following conditions hold:

1. For any set of clones $K \subset A$ and any $d \in K$, say F' is the list that CE would output on $\boldsymbol{\sigma} \setminus (K \setminus \{d\})$ (i.e. the preference profile with all the clones removed except for d). Then $\forall i \in \{1, ..., m-1\}$:
   (a) If $\exists g \in K$ such that $g \notin F_i$, then $F_i \setminus K = F'_j$ for some $j \in \{0, 1, ..., m-1\}$ (where F'_0 is defined as the empty set $\emptyset$).
   (b) Otherwise, let $g^* \in K$ be the clone with the largest index in F_i. Then $F_i \setminus (K \setminus \{g^*\})$ with g^* replaced with d is equal to F'_j for some $j \in \{0, 1, ..., m-1\}$.

2. A myopic agent (as defined in Definition 5.1) who has access to all the logs of the computation up to F_i for $i < m-1$ cannot deduct $a \in A$ such that $a \notin F_i$ but $a \in F_{m-1}$. In particular, the myopic agent does not have access to any entries from $\boldsymbol{\sigma}$ that the protocol has not used yet.

3. CE satisfies neutrality. In other words, permuting the rows and columns $\boldsymbol{\sigma}$ before the protocol is run should result in the same permutation on the output list F.

4. $A \setminus F_{m-1} = \{f(\boldsymbol{\sigma})\}$.

Along with the definition for myopic agent above, this definition provides us with all the pieces we will need to formalize out definition for Obviously Independent of Clones:

Definition 5.3 (Obviously Independent of Clones)**.** A social choice function f is Obviously Independent of Clones (OIoC) if it can be reduced to a clocked election.

Before applying this definition to various voting rules, it is worthwhile to elaborate on the purposes of each of the three conditions stated in Definition 5.2.

Intuition for Condition 1. Motivated by the personal clock auction definition by Li, Condition 1 imposes the notion of rounds onto the voting rule, requiring that its implementation can be reduced to steps in which one candidate is eliminated at a time, and the order in which the candidates are eliminated is stored in the list F_i at the ith iteration. Moreover, this condition ensures that at any point during the implementation, removing all but the longest-lasting clone from the list of eliminated candidates does not change the order we would have gotten if we had run the protocol with these clones removed. This provides a rather intuitive understanding of why the protocol is "obviously" independent of clones: once a clones does get eliminated, they know immediately at that round (and in following rounds) that they never really influenced the way in which the implementation of the voting rule runs, so it is as if they had never ran. Another way of looking at this is that even if we allowed the clones to forfeit and leave the election mid-way at any of the rounds, we would not have to rerun the protocol from start, as all of the lists with the forfeited clone removed would be identical to running the algorithm without the clone.

Intuition for Condition 2. Condition 2 rules out trivial reductions from the definition. Say we have an adversary with arbitrarily powerful computational capacity who is trying to reduce an independent of clones voting rule (that outputs a final ranking of the candidates) into a clock election. Using its unlimited computational capacity, the adversary could simply run the voting rule on $\boldsymbol{\sigma}$ using any implementation and get the resulting ranking as an output (e.g. say $A > B > C > D$ is the final ranking of the candidates after running the voting rule) and then start forming the eliminated candidates list by inputting the ranking from end to start (e.g. adding D and C and B and A). Since the voting rule is independent of clones, if the adversary ran the algorithm with clones, all of the clones would appear consecutively in the resulting ranking (e.g. $A > B > C > C_1 > D$ if C was cloned), and hence the list that the adversary would form would indeed have the same order when the clones are removed. This is undesirable, as it would permit all independent of clones voting rules that output a final ranking to be obviously independent of clones.

We could prevent this by limiting the complexity of the reduction to the clocked election, hence restricting the computing power of the adversary, but this would cause our definition of OIoC only depend on the complexity of the voting rules and hence prevent it from being a new metric (e.g., Schulze has a higher complexity than Ranked Pairs because of the step running the Floyd-Warshall algorithm). Instead, we introduce a myopic agent (see Definition 5.1) in order to ensure that the adversary does not "cheat." In other words, pausing the reduction at any step i, the myopic agent is not allowed to read new information from $\boldsymbol{\sigma}$ or perform new computations itself, but is allowed to read the logs of the reduction up until that point. For example, if the adversary is trying to perform the trivial reduction explained above, i.e. coming up with the final ranking and then filling the list of eliminated candidates one by one, the myopic agent will have access to this result, hence constructing $F_{i+1}, ..., F_{m-1}$ on its own, violating the definition.

Intuition for Condition 3. Condition 3 is intended to protect against any strategy in which the adversary exploit arbitrary orderings of the candidates. For instance, the adversary could first detect all the groups of clones by reading the entirety of $\boldsymbol{\sigma}$ and come up with an arbitrary ordering of candidates in which the clones appear right next to each other. If the voting method induces a certain strength or score parameter that the final ranking is based on, then the adversary could simply perform pairwise comparisons of candidates in the arbitrary order it has set (eliminating the one with the lower score), and hence ensure that the order of elimination with or without clones the same, which does not align with the formulation of a clocked election with rounds. Note that Condition 2 is not sufficient to protect against such strategies as the myopic agent is incapable of comparing magnitudes of numbers. Without Condition 3, all independent of clones methods that induces a strength or score metric would be OIoC.

Intuition for Condition 4. Condition 4 simply ensures that the clocked election that the voting rule is reduced to produces a winner which is consistent with the winner that the voting rule determines. We now look at the application of the definition of OIoC to various voting rules that are independent of clones; namely STV, Ranked Pairs, and Schulze method.

5.1 STV

In STV, the notion of rounds is already present in the natural implementation of the voting rule: in each round, the candidate with the least plurality votes is eliminated. Hence, we can easily define the following protocol:

Definition 5.4 (CE_{STV} Protocol)**.** Given input $\boldsymbol{\sigma}$ for candidates A and voters N, CE_{STV} performs the following procedure:

- **Step 1:** Read only the top row of $\boldsymbol{\sigma}$, counting the votes for each candidate. Pick the candidate with the least number of votes on the top row (say a_1), and declare $F_1 = \{a_1\}$. Delete all the entries of the top row of $\boldsymbol{\sigma}$ that has a_1 as its entry and replace it by **NULL** produce new preference profile $\boldsymbol{\sigma}_1$ (i.e. for all $i \in \{1, ..., |N|\}$, $\boldsymbol{\sigma}'_{1i} = \begin{cases} \textbf{NULL} & \text{if } \boldsymbol{\sigma}_{1i} = a_1 \\ \boldsymbol{\sigma}_{1i} & \text{otherwise} \end{cases}$
- **Step $\boldsymbol{i}$** (for $i \in \{2, ..., m-1\}$)**:** Once step $i-1$ is complete, count the votes from the top non-**NULL** entry of each column of $\boldsymbol{\sigma}_{i-1}$ (by starting from the top of each column and moving the next entry only if the entry is **NULL**). Determine the candidates with the least number of votes among these "top" votes (say a_i) and declares $F_i = F_{i-1} \cup \{a_i\}$ (where a_i is added to the end). For each of the columns that have a_i as their top non-**NULL** entry, change this entry to **NULL** to produce $\boldsymbol{\sigma}_i$ and moves to the next entry in the column. If the candidate in the next entry is in F_i, change this entry to **NULL** as well and keep moving to the next entry until either a candidate not in F_i is encountered or the column is all **NULL**.

Note that after step $m-1$, $\boldsymbol{\sigma}_{m-1}$ has only **NULL** as its entries and the F_{m-1} has all but one candidate.

Theorem 5.5. *CE_{STV} is a Clocked Election for STV.*

Proof. See Appendix A.3. □

Corollary 5.6. *STV is Obviously Independent of Clones.*

5.2 Ranked Pairs

Next, we analyze how Ranked Pairs (RP) can be reduced to a clocked election. For this voting rule, the notion of rounds can be induced using the order in which we add the edges to the graph (without creating a cycle). A key observation is that once any vertex has an incoming edge locked in the graph, the corresponding candidate no longer has the chance of being the winner (as the winner will be a source of the final graph) and hence can be added to the list F. Based on this idea, we define the following protocol:

Definition 5.7 (CE_{RP} Protocol)**.** Given input $\boldsymbol{\sigma}$ for candidates A and voters N, CE_{RP} performs the following procedure:

- **Step 0:** For each $a, b \in A$, read every column of $\boldsymbol{\sigma}$ in order to count the number of voters that prefer a over b and viceversa. Use these numbers to fill the majority matrix $M^{(0)}$ where $M^{(0)}_{ab}$ is the number of voters that prefer a over b minus the number of voters that prefer b over a. Initiate a graph $G_0(V, E_0)$ where $V = A$ and $E_0 = \emptyset$.

- **Step $\boldsymbol{i}$** (for $i \in \{1, ..., m-1\}$)**:** Once step $i-1$ is complete, the current graph is $G_{i-1}(V, E^{i-1})$. Query the current majority matrix $M^{(i-1)}$ to find the largest non-**NULL** entry (say $M^{(i-1)}_{ab}$). First, set $M^{(i)}_{ab} =$ **NULL** to produce $M^{(i)}$ from $M^{(i-1)}$. Add (a, b) to E_{i-1} to get $G_i(V, E_i) = G_i(V, E_{i-j} \cup \{(a, b)\})$ only if it does not create a cycle. Otherwise, repeat by finding the next largest entry in the majority matrix, set it to **NULL**, check if the corresponding edge creates a cycle, etc. until either an edge not creating a cycle is found and added to $G_i(V, E_i)$ or there are no more non-**NULL** positive entries in $M^{(i)}$, in which case terminate. Once an edge (a, b) is added to $G_i(V, E_i)$, check if b is in F_{i-1}. If $b \notin F_{i-1}$, add it to produce $F_i = F_{i-1} \cup \{b\}$ and move to step $i+1$ (or terminate if $i = m-1$). If $b \in F_{i-1}$, repeat this step again with the current $G_i(V, E_i)$ $M^{(i)}$ until an edge pointing at a vertex not in F_{i-1} is added to $G_i(V, E_i)$, and only then move to step $i+1$.

Note that withing a single step i, the protocol might add multiple edges to the graph $G_i(V, E_i)$, but by construction it will only eliminate one candidate per step. Hence, at step $m-1$, F_{m-1} has all but one candidate, as desired.

Theorem 5.8. *CE_{RP} is a Clocked Election for Ranked Pairs.*

Proof. See Appendix A.4. ☐

Corollary 5.9. *Ranked Pairs is Obviously Independent of Clones.*

6 The Schulze Method is Not Independent of Clones

Having shown that both STV and Ranked Pairs is obviously independent of clones, we now present an impossibility result:

Theorem 6.1. *The Schulze is not obviously independent of clones.*

Proof. See Appendix A.5. ☐

7 Conclusion

This paper bridges two separate phenomena from computational social choice and mechanism design - the *independence of clones* property of a social choice function and the game-theoretic notion of *obviousness.*

Historical example has established why the former is a valuable trait for social choice functions. Social choice functions that are independent of clones are robust to perturbations in relation to the spoiler effect. Spoiler-prone election systems weaken faith in democratic institutions, and open the door for strategic nomination by candidates, parties, and election administrators. Moreover, in order to avoid the spoiler effect, qualified and well-liked candidates will drop out of the election early. This leads to adverse secondary effects - namely, limited voter-choice and consolidation of political ideologies. These effects can then lead voters to become disgruntled, less engaged, and less likely to turn out as they are forced to settle for a candidate who more coarsely aligns with their priorities.

The game theoretic notion of *obviousness* as discussed by Li in the context of strategy-proofness, is naturally extendable to the realm of social choice functions. Namely, when dealing with voting, our constituencies are usually not agents who are willing and able to reason through a series of contingent scenarios, or to read and internalize a proof of a social choice function's properties. As such, in practice it is not always enough for a social choice function to possess some desideratum; it must possess it *obviously*. In our case, clone candidates must both understand that their presence in the election will not be putting their policy agenda at undue risk and, moreover, that voting agents will not blame them for spoiling an election should they and their associated set of clones lose.

Motivated by these two paradigms, we sought to define a new paradigm connecting the two in the context of computational social choice. This led us to formulate a new definition - that of *obvious independence of clones*. Our definition of OIoC captures a fundamental intuition of why it is harder to reason about Schulze than STV or Ranked Pairs. STV and Ranked Pairs have a fundamental temporal linear structure. Namely, agents are added or eliminated in a strictly sequential manner. Such a structure is both straightforward and robust to perturbation. Agents can clearly see why adding a clone quickly resets to the original profile without the clone, without needing to engage in multiple steps of contingent reasoning through hypotheticals. In contrast, Schulze requires multidirectional reasoning. The partial orderings produced during the running of Schulze are not fixed and can be disrupted based on the voter profile.

References

[BL07] Scott Bennett and Rob Lundie. *Australian electoral systems.* Parliament of Australia. Department of Parliamentary Services, 2007.

[Li17] Shengwu Li. Obviously strategy-proof mechanisms. *American Economic Review*, 107(11):3257–87, 2017.

[Lin17] Yehuda Lindell. How to simulate it–a tutorial on the simulation proof technique. *Tutorials on the Foundations of Cryptography*, pages 277–346, 2017.

[oSN21] Secretary of State and Andrew Needham. State of Oregon, Nov 2021.

[Sch11] Markus Schulze. A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method. *Social choice and Welfare*, 36(2):267–303, 2011.

[Sch18] Markus Schulze. The schulze method of voting. *arXiv preprint arXiv:1804.02973*, 2018.

[SVWX21] Krzysztof Sornat, Virginia Vassilevska Williams, and Yinzhan Xu. Fine-grained complexity and algorithms for the schulze voting method. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 841–859, 2021.

[Tid87] Nicolaus Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

A Proofs

A.1 Proof of Lemma 2.3

Consider N voters and 2 candidates A and B, where P voters prefer B to A and $N-P$ voters prefer A to B (with $P \geq 1$ by assumption). Let $P_1(A)$ denote the Borda points obtained by candidate A, and let $P_1(B)$ denote the Borda points obtained by candidate B, where $P_1(A) > P_1(B)$ WLOG. Next we add Q clones of B, ranking strictly less than B. Then each of the P voters who prefer B to A grants Q additional Borda points to B, for a total of QP additional points. Each of the $N-P$ voters who prefer A to B grants Q additional points to B and Q additional points to A, for a total of $Q(N-P)$ Borda points. Hence, the new total Borda points after the addition of the clones is $Q(N-P)$ for A and QN for B. Then, for any Q such that $Q > \frac{P_1(A)-P_1(B)}{P}$, we conclude that the winner has been flipped.

A.2 Proof of Lemma 2.4

Consider, an election with a set of candidates that includes a set of clones C. In each individual voter's ranking, all of the clones must be adjacent. Therefore, at any given moment in the election, if a clone is eliminated and there are other clones remaining, all of its first place votes are reallocated to another clone. Therefore, the number of first-place votes for each of the non-clone candidates is unchanged and the number of first-place votes that the set of clones has collectively is also unchanged. Furthermore, when another candidate is eliminated while some non-empty set of clones C' is still in the election, each of the eliminated candidate's first place votes is either allocated to either a member of C' or non-member of C'. If it is allocated to a non-member of C', there is no way of changing C' such that it it can change this allocation. Furthermore, if the vote is allocated to a member of C', since all of the members of C' are adjacent in the ranking, it will always be allocated to a member of C' even if C' is modified. Therefore, in both of these steps, the set of clones will always have the same number of first place votes in aggregate, no matter the composition of C. Therefore, if we compare the original election to the election with clones, by the time that all but one of the clones are eliminated, the number of first place votes that the final clone and the other candidates have will be identical to the original election.

Therefore, we conclude that STV is IoC.

A.3 Proof of Theorem 5.5

We want to show that CE_{STV} fulfills the four conditions of Definition 5.2.

Condition 1: For any set of clones $K \subset A$ and any $d \subset K$, let F' be the list that CE_{STV} would output on $\boldsymbol{\sigma}' = \boldsymbol{\sigma} \setminus (K \setminus \{d\})$. Note that $F = F'$ for $|K| = 1$, so Condition 1 is trivially satisfied. Consider $|K| > 1$. We would like to do a proof by induction on the step i that each F_i satisfies one of (a) or (b) listed under Condition 1. **Base case:** let $i = 1$. Note that $|F_1| = 1$ and $|K| > 2$, so F_1 needs to satisfy (a). First consider the case in which $F_1 = \{g\}$ for some $g \in K$. Then $F_1 \setminus K = \emptyset = F'_0$, so (a) is indeed satisfied. Now, consider the case where $F_1 = \{a\}$ for some $a \notin K$. This implies a was the candidate that appeared the least on the top row of $\boldsymbol{\sigma}$. Notice that by definition of clones the top row of $\boldsymbol{\sigma}'$ is identical to that of $\boldsymbol{\sigma}$ except each $g \in K \setminus \{d\}$ is replaced with d (since the clones on the top row transfer their votes to d once all the other clones are removed from $\boldsymbol{\sigma}$). So the number of plurality votes for each remaining candidate in $\boldsymbol{\sigma}'$ is the same as their plurality vote in $\boldsymbol{\sigma}$, except for d, whose plurality vote in $\boldsymbol{\sigma}'$ is greater or equal to its plurality vote in $\boldsymbol{\sigma}$. Since no candidate had their plurality vote decrease, a is still the candidate with the least amount of plurality votes in $\boldsymbol{\sigma}'$, implying $F'_1 = \{a\}$ and hence $F_1 = F'_1$, as desired.
Inductive case: Assume F_i satisfies Condition 1, implying it either satisfies (a) or (b). We want to show that F_{i+1} satisfies Condition 1 in either of these cases.

- If F_i satisfies (a), this implies $\exists g \in K$ such that $g \notin F_i$ and that $F_i \setminus K = F'_j$ for some $j \in \{0, 1, ..., m-1\}$ (where F'_0 is defined as the empty set $\emptyset$). First consider the case $F_{i+1} = F_i \cup \{g\}$ and $K \subset F_{i+1}$, so the last clone is eliminated. This implies that $A \setminus F_i$ includes a single clone g. Similarly, $F'_j = F_i \setminus K$, this implies $A \setminus F_i$ is the same as $A \setminus F'_j$ except with g replaced by d in the former. Hence, $\boldsymbol{\sigma}_i$ and $\boldsymbol{\sigma}'_j$ (the preference profile of remaining candidates after removing the candidates F_i or F'_j, respectively) is identical except for g is replaced by d in $\boldsymbol{\sigma}'_j$. Since g has the least plurality votes in $\boldsymbol{\sigma}_i$, then d must have the least plurality votes in $\boldsymbol{\sigma}'_j$, implying $F'_{j+1} = F'_j \cup \{d\}$. Since g is the clone with the largest index in F_{i+1}, and since $F_i \setminus K = F'_j$, then $F_{i+1} \setminus (K \setminus \{g\}) = (F_i \setminus K) \cup \{g\} = F'_j \cup \{g\}$ where we replace g with d is $F'_j \cup \{d\} = F'_{j+1}$, showing that F_{i+1} satisfies (b). Second, consider the case in which $F_{i+1} = F_i \cup \{g\}$ and $\exists g' \in K$ such that $g' \in A \setminus F_{i+1}$, so there are still clones remaining in the contest. In that case, $F_{i+1} \setminus K = F_i \cup \{g\} \setminus K = F_i \setminus K = F'_j$ by assumption, so F_{i+1} satisfies (a). Lastly, consider the case $F_{i+1} = F_i \cup \{a\}$ for some $a \notin K$. This implies a was the candidate that appeared the least on the top non-**NULL** entries of each column $\boldsymbol{\sigma}_i$. Notice that by definition of clones the top row of $\boldsymbol{\sigma}'_j$ is identical to that of $\boldsymbol{\sigma}_i$ except each $g \in K \setminus \{d\}$ is replaced with d (since the clones on the top row transfer their votes to d once all the other clones are removed from $\boldsymbol{\sigma}_i$). This follows from the fact that all the non-clone candidates eliminated to produce $\boldsymbol{\sigma}_i$ are identical to the non-clone candidates eliminated to produce $\boldsymbol{\sigma}'_j$, and there is still at least clone remaining. So the number of plurality votes for each remaining candidate in $\boldsymbol{\sigma}'_j$ is the same as their plurality vote in $\boldsymbol{\sigma}_i$, except for d, whose plurality vote in $\boldsymbol{\sigma}'_j$ is greater or equal to its plurality vote in $\boldsymbol{\sigma}_i$ (this is true even if $d \in F_i$, in which case it has it has zero plurality votes in $\boldsymbol{\sigma}_i$). Since no candidate had their plurality vote decrease, a is still the candidate with the least amount of plurality votes in $\boldsymbol{\sigma}'_j$, implying $F'_{j+1} = F'_j \cup \{a\} = F_i \cup \{a\} \setminus K = F_{i+1} \setminus K$ and hence F_{i+1} satisfies (a).

- If F_i satisfies (b), this implies $K \subset F_i$ and if given that g^* is the clone with the highest index in F_i, then $F_i \setminus (K \setminus g^*)$ with g^* replaced with d is equal to F'_j for some $j \in \{0, ..., m-1\}$. Since $K \subset F_i$ and $d \in F'_j$, all the clones are eliminated in both cases, implying $\boldsymbol{\sigma}_i$ is identical to $\boldsymbol{\sigma}'_j$ ignoring **NULL** values. Hence, the next candidate with the least plurality votes should be the same in both cases (say a). Since F'_j is equal to $F_i \setminus (K \setminus g^*)$ with g^* replaced with d , then $F'_{j+1} = F'_j \cup \{a\}$ is equal to $F_i \cup \{a\} \setminus (K \setminus g^*) = F_{i+1} \setminus (K \setminus g^*)$ with g^* replaced with d, showing that F_{i+1} satisfies (b).

Hence, in either case, F_{j+1} satisfies Condition 1, completing the inductive case.

Condition 2: This condition follows from the fact that when CE_{STV} produces F_i by looking at at most the top i entries of each column. Hence, the myopic agent is clueless about how the last $m-i$ rows of $\boldsymbol{\sigma}$ are distributed, implying that it cannot predict the remaining order in which the candidates will be eliminated (as it will not know which candidates will receive the transfered votes of the eliminated candidates in future steps). It is worth noting that if more than half of the plurality votes went to the same candidate at step i, an agent capable of contingent reasoning could deduct that this candidate will be the final winner and hence deduct that the other candidates will be eliminated. However, specified in Definition 5.1, the myopic agent is not capable of performing such a contingent reasoning, especially since the protocol never calculates the ratio of plurality votes that belong to each candidate, and instead only calculates the candidate with the least plurality votes.

Condition 3: The neutrality of this method follows from the neutrality of STV, and more specifically from the fact that at each step, the candidate that is eliminated is only dependent on the remaining plurality votes, so if we permuted the labels of the candidates, the order in which they are eliminated would also get permuted in the same way.

Condition 4: $A \setminus F_{m-1}$ does indeed give us the STV winner, which follows from the fact that CE_{STV} follows the rules of STV at each step by eliminating the voter with the least plurality votes transferring the votes of each voter that voted for this candidate to their next choice.

A.4 Proof of Theorem 5.8

We want to show that CE_{RP} fulfills the four conditions of Definition 5.2.

Condition 1: For any set of clones $K \subset A$ and any $d \subset K$, say F' is the list that CE_{STV} would output on $\boldsymbol{\sigma}' = \boldsymbol{\sigma} \setminus (K \setminus \{d\})$. Note that $F = F'$ for $|K| = 1$, so Condition 1 is trivially satisfied. Consider $|K| > 1$. Say M and M' are the majority matrices produced by the protocol on input $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$, respectively. Similarly, say that O and O' are ordered lists of edges (a, b) in decreasing order of $M_{a,b}$ and $M'_{a,b}$, respectively (which the order in which the protocol will consider these edges). Notice that $\forall a, b \in A \setminus (K \setminus \{d\})$, we have $M_{a,b} = M'_{a,b}$, so the order in which these edges appear in both O and O' are the same. The only additional edges contained in O and not in O' are the ones involving clones other than d. It follows from the definition of clones that for any $g \in K$ and any $a \in A \setminus K$, we have $M_{g,a} = M'_{d,a}$ and $M_{a,g} = M'_{a,d}$. Hence, in O, all such $M_{g,a}$ is tied with $M_{d,a}$ and all such $M_{a,g}$ is tied with $M_{a,d}$. In addition, O also contains edges (g, g') for any $g, g' \in K$, which O' do not contain. Note that such edges could be located anywhere in O, since we have no information on $M_{g,g'}$. Say $G_(V, E)$ and $G'(V, E')$ are the final graphs produced by running the protocol on majority matrices produced by the protocol on input $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$, respectively. We would like to show that any $(a, b) \in O \cap O'$ is in E if and only if it is in E' for $a, b \neq d$, and an edge (d, b) (or similarly (a, d)) in E if and only if it and all other (g, b) (or similarly (a, g)) for $g \in K$ is in E'. We do this by induction on the order in which these edges appear in O':

Base case: First element in O'. Note that the first element in O' (say (a, b)) will always be added to the graph by the $CE_{RP}(\boldsymbol{\sigma}')$, since it is the first edge and cannot create a cycle. Since elements of O' appear in the same order in O, when $CE_{RP}(\boldsymbol{\sigma})$ considers (a, b), could only be of form (g, g') for $g, g; \in K$, (a, g) for $g \in K$ (if $b = d$) and (g, b) for $g \in K$ (if $a = d$). Any cycle containing (a, b) would need an edge (g, a) and an edge (b, g') in it with $g, g' \in K$. Since we cannot have $a = b = d$, at most only one of a or b is in K, it is impossible for the protocol to have already considered (g, a) *and* (b, g'), implying that (a, b) gets added to the graph by $CE_{RP}(\boldsymbol{\sigma})$. If we have $a = d$ (or similarly $b = d$) then all (g, b) (or similarly (a, g)) for $g \in K$ would appear tied in O with (a, b). Since only one of these vertices is a clone, the same arguement about the impossibility of cycles apply, so all such (g, b) (or similarly (a, g)) also gets added to G, as desired.

Inductive case: Say the first i elements in O' satisfy the condition, and show that this holds true for the $i + 1$th element. Note that since the element of O' appear in the same order in O, the first i elements in O' have already been considered by $CE_{RP}(\boldsymbol{\sigma})$ once the $i + 1$th element of O' (say (a, b)) is considered. Say $(a, b) \notin G'$. This implies that (a, b) form a cycle with the edges in the first i elements of O' that were added to G'. But these edges are also added to the G by the inductive hypothesis, so the same cycle would form if (a, b) was added to G. Hence, (a, b) does not get added to G if it is not added to G'. Conversely, say (a, b) was added to G', implying it does not form a cycle with the edges in the first i elements of O' that were added to G'.

The only additional edges that might have already added to G once $CE_{RP}(\boldsymbol{\sigma})$ considers (a, b) are the ones involving clones. Thus, any cycle that (a, b) might cause in G must involve clones. If this cycle contained only a single clone, say g, the same cycle would have formed in G' with g replaced with d (since all the edges weights of d are the same as those of g, as shown above and an edge containing g is in the G iff the analogous edge containing d is in the G' and G, by the inductive hypothesis), which is a contradiction. If the cycle contained multiple clones, on the other hand, say g is the first clone that appears in the cycle via edge (x, g) and g' is the last (via edge (g', y)) then we could replace these edges, along with the edges that come in between, with (x, d) and (d, y) (which need to be in the graph by the inductive hypothesis) and still form the cycle, which is a contradiction. Hence, (a, b) does not form a cycle. The argument for how edges containing d are added to G iff all analogous edges containing any

$g \in G$ is analogous to that in the base case: any cycle that would form by one would form by the other, since all the edges that have been added so far fulfilled this condition by the inductive hypothesis. This concludes the inductive case.

Hence, an edge in O' appears in G iff it appears in G, and these edges get added in the same order as the elements in O' appear in the same order in O. Hence, if all the candidates $b \in F'$ that were eliminated by an edge (say (a,b)) in G' were eliminated by the same edge in G, the ordering of these candidates would be the same in both F and F'. Note that by the induction above, the only other way b could have been eliminated by $CE_{RP}(\boldsymbol{\sigma})$ is if $a = d$ and b gets eliminated by another (g,b). However, since (g,b) has the same weight as (d,b), this would not disrupt the order in which b is eliminated with respect to the other candidates. Hence, all the none-cloned candidates appear in the same order in F and F', fulfilling (a). To see that (b) is also satisfied whenever relevant, assume that the winner is not among the clones (otherwise (b) would be vacuously satisfied).

Notice that the clones in K can be eliminated by edges of form (g,g') for $g,g' \in K$ in any arbitrary step (since these edges can appear anywhere in O) but the last clone g^* must always be eliminated by an edge (a,g^*) for $a \notin K$ (since if all clones were eliminated by edges of form (g,g'), then we would have a cycle, which is a contradiction). The analogous edge (a,d) gets added to G' in the same order by the induction above, so g^* gets eliminated by $CE_{RP}(\boldsymbol{\sigma})$ by an analogous edge that eliminates d by $CE_{RP}(\boldsymbol{\sigma'})$ showing that g^* replaces d when going from F' to F, hence satisfying (b). Thus, the lists F and F' do indeed satisfy Condition 1.

Condition 2: This follows from the definition of a myopic agent, who is incapable of making comparisons of numbers or run max/min queries. Hence, after step i (when F_i is computed, $M^{(i)}$ is the current majority matrix and G_i is the current graph), there is no way for the myopic agent to query $M^{(i)}$ to get the next largest non-**NULL** entry, hence there is no way for it to know which candidates will be eliminated next. To see this more formally, for any candidate $a \notin F_i$ the next $m-1$ largest weight edges in $M^{(i)}$ could be of form (a,b) for each $b \in A \setminus \{a\}$, and all of these would be added to G_i since a currently has no incoming edges (as evidenced by $a \notin F_i$) so none of these edges could form a cycle. Adding all of these edges to G_i would ensure that m is the final winner, as no edge pointing to it can be added any longer without forming a cycle. Since this is true of all $a \notin F_i$, and since the myopic agent cannot distinguish between these cases without querying $M^{(i)}$, it is not possible for it to predict with certainty that any $a \notin F_i$ will be eliminated in the next $m-1-i$ rounds.

Condition 3: The neutrality of this method follows from the neutrality of Ranked Pairs, and more specifically from the fact that at each step, the edges that are added (and hence the candidates that are eliminated) is only dependent on the majority matrix, so if we permuted the labels of the candidates (and hence the rows/columns of M), the order in which they are eliminated would also get permuted in the same way.

Condition 4: $A \setminus F_{m-1}$ does indeed give us the Ranked Pairs winner, which follows from the fact that CE_{RP} follows the rules of Ranked Pairs at each step by forming a cycle from the edges with highest majority scores without forming cycles.

A.5 Proof of Theorem 6.1

Notice based on Definition 5.3, in order to prove this theorem, we must show that the Schulze Method cannot be reduced to a clocked election. Assume for the sake of contradiction that there exits a protocol CE_S that is a Clocked Election for the Schulze Method, i.e. it fulfills the four conditions given in Definition 5.2, which we will denote as conditions $\boxed{1}$, $\boxed{2}$, $\boxed{3}$, and $\boxed{4}$ for clarity. We also use the notation for the Strength Matrix (S) introduced in Section 3.2.

Note that by condition $\boxed{4}$, run on any preference profile $\boldsymbol{\sigma}$ with candidates A and voters N, the list F that CE_S outputs contains all the candidates except for the winner of the Schulze Method applied (denote $f_S(\boldsymbol{\sigma})$). Notice that for any $a \in A$, we have $f_S(\boldsymbol{\sigma}) = a \Leftrightarrow S[a,b] > S[b,a] \forall b \in A \setminus \{a\}$.

Then for any b that gets added to F_i on step i, since $b \neq f_S(\boldsymbol{\sigma})$, the protocol must first ensure that $\exists c \in A \setminus \{b\}$ such that $S[c,b] > S[b,c]$, which it cannot do without without computing and comparing $S[c,b]$ and $S[b,c]$ at least for one $c \in A \in A \setminus \{b\}$. Note that at any step $i < m-1$ if the protocol had compared $S[a,b]$ and $S[b,a]$ for all possible $a, b \in A$, then the myopic agent reading all the logs up to step i would be capable of deducting the full ranking that the Schulze method would produce, and hence determine which candidates would be eliminated next. This violates the assumption that CE_S satisfies condition $\boxed{2}$, hence at any step $i < m-1$, there must still be some $a, b \in A$ such that the protocol has not performed the comparison between $S[a,b]$ and $S[b,a]$ yet. This implies that the protocol must decide on a order of candidates (say $\pi : \{1, ..., m\} \to A$, injective) in which it will perform such comparisons (which cannot be based on the final ranking of the candidates, as there is no way for the protocol know ranking a priory without computing it, which would again violate condition $\boxed{2}$), such that at step i it compares $\pi(i)$ with all $\pi(j)$ such that $j < i$ (which must produce exactly one new eliminated candidate by transitivity).

Digression before the proof. First, in order to gain some intuition, we consider the following concrete example. Assume that the protocol determines π arbitrarily (i.e. using an arbitrary ordering of the candidates in A) and consider a preference profile $\boldsymbol{\sigma}_0$ with $A = \{a, b, c, d\}$ that has $a > b > c > d$ as its final Schulze ranking (constructing such an example is not difficult, as we know by transitivity that such a ranking exists for all possible $\boldsymbol{\sigma}$ with four candidates, so we will just need to label accordingly). Say that π_0 is the order in which CE_S computes candidates when given $\boldsymbol{\sigma}_0$ as an input. Consider the permutation $\tau_1 : A \to A$ that has $\tau_1(\pi_0(1)) = b$, $\tau_1(\pi_0(2)) = c$, $\tau_1(\pi_0(3)) = d$, and $\tau_1(\pi_0(4)) = a$. Say $\boldsymbol{\sigma}_1$ is the preference profile constructed by applying τ_1 to $\boldsymbol{\sigma}_0$. Since CE_S produces π_0 using an arbitrary ordering of candidates (i.e. independent of the actual votes), then given $\boldsymbol{\sigma}_1$ as an input, it must compare the candidates with ordering $\pi_1 = \{b, c, d, a\}$[3]. Then comparison in the first step will give $b > c$ ($F_1 = \{c\}$), the comparisons in the second step will give partial ranking $b > c > d$ ($F_2 = \{c, d\}$), and the comparisons in the third step will give final ranking $a > b > c > d$ ($F_3 = \{c, d, b\}$). Now consider a second permutation τ_2 that maps $\tau_2(a) = c$, $\tau_2(b) = a$, $\tau_2(c) = d$, and $\tau_2(d) = b$ and say $\boldsymbol{\sigma}_2$ is the preference profile constructed by applying τ_2 to $\boldsymbol{\sigma}_1$. Since CE_S produces an arbitrary ordering of candidates (independent of actual candidates), and since it produced order $\pi_1 = \{b, c, d, a\}$ when given $\boldsymbol{\sigma}_1$, it must produce order $\pi_2 = \{a, d, b, c\}$ when given $\boldsymbol{\sigma}_2$. Performing the comparisons in this order, the comparison in step 1 will give $a > d$ ($F'_1 = \{d\}$), the comparisons in step 2 will give $a > b > d$ ($F'_2 = \{d, b\}$), and the comparisons in step 3 will give $a > b > c > d$ ($F'_3 = \{d, b, c\}$). So going from $\boldsymbol{\sigma}_1$ to $\boldsymbol{\sigma}_2$, the list that CE_S outputs went from $F = \{c, d, b\}$ to $F' = \{d, b, c\}$, which does not correspond to the permutation τ_2 we performed on $\boldsymbol{\sigma}_1$ to produce $\boldsymbol{\sigma}_2$, which shows that any protocol that chooses the order in which the strength of the candidates are compared arbitrarily violates condition $\boxed{3}$. By assumption, CE_S satisfies $\boxed{3}$, so it must choose the order of comparisons non-arbitrarily. Since the only source of non-arbitrariness in the Schulze method is the pairwise comparison matrix P (as the strength matrix S and hence the winner is entirely determined by this matrix), this implies that CE_S implements some deterministic function h that takes S as an input and outputs an order of candidates π and performs the pairwise comparisons of strengths in this order.

Before proceeding with the proof of the theorem, we will need two lemmas:

Lemma A.1. *Assume CE_S is indeed a clocked election for Schulze Method. Say h is the deterministic function with which CE_S computes the order in which it will compare the strengths of its candidates. Then there exists some preference profile $\boldsymbol{\sigma}$ for candidates A with associated pairwise comparison matrix P and strength matrix S such that if $\pi = h(P)$ then there exists $a, b \in A$ with $\pi^{-1}(a) < \pi^{-1}(b)$ (i.e. a comes before b in the order π) but $S[b,a] > S[a,b]$ (we have $b > a$ in the final ranking).*

Proof of Lemma A.1. Assume for the sake of contradiction that h is such that for all possible $\boldsymbol{\sigma}$ (and associated matrices P and S), if $h(P) = \pi$ then $\pi^{-1}(a) < \pi^{-1}(b)$ iff $S[a,b] > S[b,a]$. This implies that

[3] Again, we abuse notation here: $\pi_1 = \{b, c, d, a\}$ denotes $\pi_1(1) = b, \pi_1(2) = c, \pi_1(3) = d$, and $\pi_1(4) = a$

for all preference profiles, the order that h outputs is the final ranking of the Schulze Method applied to this preference profile. Since h is a part of CE_S, any myopic agent that has the logs to the computation of CE_S will have access to π, which is a contradiction to assumption $\boxed{2}$ as the myopic agent will be able to list the remaining candidates that will be eliminated simply by observing the order in which CE_S plans to compare them. □

Lemma A.2. *Assume CE_S is indeed a clocked election with ordering function h. Take any $\boldsymbol{\sigma}$ (with matrices P, S) and say $\pi = h(P)$. Consider any $a, b \in A$ such that $\pi^{-1}(a) < \pi^{-1}(b)$ and $S[b,a] > S[a,b]$. Then we can produce a preference profile $\boldsymbol{\sigma}'$ (with matrices P', S'), simply by adding clones of b to $\boldsymbol{\sigma}$ such that at least one of the clones of b (say b^*) satisfies $(\pi')^{-1}(b^*) \leq (\pi')^{-1}(a)$, where $\pi' = h(P')$.*

Before proving this Lemma, let's first discuss why it actually makes sense, especially considering the restrictions imposed by the definition of clones. The key idea is that as we go from $\boldsymbol{\sigma}$ to $\boldsymbol{\sigma}'$ (and hence P to P') we have the flexibility of arbitrarily choosing $P_{b',b''}$ for any two clones of b. For example, say h simply ordered the candidates in A by the decreasing order of the largest entries in the corresponding row of P (i.e. the candidate a with the largest $P[a,b]$ for some $b \in A \setminus \{a\}$ is put first in order, and so on). Since $P[i,j]$ is at most n (and at least $-n$), and since we get to choose $P[b^*, b]$ for any clone of b that we add, for any a that comes before b in $\pi = h(P)$ we can simply ensure that b^* gets ordered before a in the new order $\pi' = h(P')$ (or at least tied) by ensuring $P[b^*, b] \geq \max_{z \in A} P[a,z]$. Hence, the lemma is satisfied for this specific example of h. We now prove it for all general h, given that CE_S is indeed a clocked election.

Proof of Lemma A.2. Take any $\boldsymbol{\sigma}$ with associated matrix P and say $\pi = h(P)$. Take any $a, b \in A$ and say $\pi^{-1}(a) < \pi^{-1}(b)$. Now, produce m clones of b to produce $\boldsymbol{\sigma}'$ and associated matrix P'. For each $c \in A$, call one clone b_c and choose the pairwise comparisons between the clones such that for any $c, d \in A$, we have $P'[b_c, b_d] = P[c,d]$ (in other words, the clone compare with each the exact same way that their corresponding non-clone candidates compare to one another) notice that P' restricted to the rows and columns corresponding to the A and P' restricted to the rows and columns corresponding to the $\{b_c | c \in A\}$ both look identical to P by construction (see Figure 2).

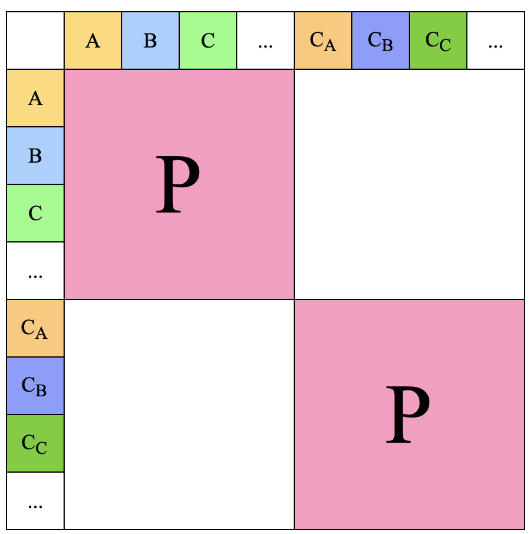


Figure 2: The new pairwise comparison matrix P' after introducing the "imposter" clones of candidate C. P is the original pairwise comparison matrix.

In a way, the clones $A_b \equiv \{b_c | c \in A\}$ are "imposters" that are locally indistinguishable from the actual candidates A. Suppose h was capable of distinguishing between the original candidates A and the imposters A_b. However, this would violate cause CE_S to violate Condition $\boxed{2}$, as any myopic agent would be capable of reading the logs of h to understand which group are the clones and which group are not, so once a single clone is eliminated in comparison to a non-cloned candidate, the myopic agent can list the remaining clones as future eliminations without any further computation. Since h cannot distinguish between A and A_b, the resulting order $h(P') = \pi'$ must be symmetric with respect to these two sets, i.e. each $c \in A$ is tied with the clone b_c in the order π'. In particular, one clone (say b^*) ties with the original candidate that appears the first in π', implying $\pi'^{-1}(b^*) \leq \pi'^{-1}(a)$, as desired.

□

Choose any A with $|A| > 2$ (note that if $|A| = 2$, Schulze reduces to plurality). By Lemma A.1, we can choose $\boldsymbol{\sigma}$ with associated matrix P such that given $\pi = h(P)$, there exist a, b such that $\pi^{-1}(a) < \pi^{-1}(b)$ (so $\pi = \{(...), a, (...), b, (...)\}$), yet $S[b, a] > S[a, b]$. Then, by Lemma A.2 (which is symmetric in its argument, so can be applied in both directions), we can clone a to produce a preference profile σ' with matrices P', S', and $\pi' = h(P')$ such that a^*, one of the clones of b, satisfies $(\pi')^{-1}(a^*) \geq (\pi')^{-1}(b)$ (so $\pi' = \{(...), a, (...), b, (...), a^*, (...)\}$). We will now conclude with the violation of Condition $\boxed{1}$. Let F and F' be the list that CE_S would output on σ and σ', respectively. Note that given in put σ, by the time CE_S compares b with the other candidates (say step i), a has already been compared. If a has not been eliminated yet (so is in the list of $i-1$ candidates that b will be compared with), comparing b with a will result in the elimination of a (and b will remain uneliminated since a compared better than the $i-2$ other candidates already on the list, hence so does b by transitivity), implying a comes before b in P. If a was already eliminated by the time b is compared with the other candidates, then again a comes before b in P, showing that this holds true in all cases. Now, consider the point (say step j) in which CE_S, given input σ', compares a^* with the $j-1$ already compared candidates, among which b is included as it comes before a^* in π'. If b is eliminated by the time a^* is compared with the other candidates, then b appears before the longest lasting clone of a in P', which violates (b) of Condition $\boxed{1}$ since a appears before b in P. If on the other hand, b is not eliminated by the time a^* we need to consider three cases. Since $|A| > 2$, there exist $c \in A \setminus \{a, b\}$, hence at least one of the following must hold true:

- **Case 1:** c appears between a and b in the original order π (denote this as $\pi \supset \{a, c, b\}$). Since b is assumed to have not eliminated by the time we consider a^* in π', we must have $S[b, c] > S[c, b]$. This implies that once b is compared, both a, c must have joined the list of eliminated candidates. Whichever among these two candidates (a and c) is eliminated earlier, we could place a clone of that candidate after b using Lemma A.2, hence switch the order in which they are eliminated, violated (b) of $\boxed{1}$.

- c appears between a and b in the original order π (denote this as $\pi \supset \{a, c, b\}$). Since b is assumed to have not eliminated by the time we consider a^* in π', we must have $S[b, c] > S[c, b]$. This implies that once b is compared, both a, c must have joined the list of eliminated candidates. Whichever among these two candidates (a and c) is eliminated earlier, we could place a clone of that candidate after b using Lemma A.2 (so $\pi' \supset \{a, c, b, a^*\}$ or $\pi'' \supset \{a, c, b, a^*\}$), hence switch the order in which they are eliminated, violated (b) of $\boxed{1}$.

- **Case 2:** c appears before a and b in the original order π ($\pi \supset \{c, a, b\}$). Since b is assumed to have not eliminated by the time we consider a^* in π', we must have $S[b, c] > S[c, b]$. This implies that this is identical to Case 1, with just a and c switched. Hence, this case too violates (b) of $\boxed{1}$ by the same argument.

- **Case 3:** c appears after a and b in the original order π ($\pi \supset \{a, b, c\}$). Note that if $S[c, b] > S[b, c]$ then $S[c, a] > S[a, c]$ by transitivity so this is identical to Case 1, and (b) of $\boxed{1}$ is violated. Otherwise, $S[c, b] < S[b, c]$, so c is eliminated as soon as its compared with the previous candidates (hence c appears after a in P^*); however, if we clone a and place it after c in the order using Lemma A.2 (hence producing $\pi''' \supset \{a, b, c, a^*\}$), then the longest lasting clones of a^* appears after c in P''', violating (b) of $\boxed{1}$.

Thus, we have seen that in all cases, $\boxed{1}$ is violated by CE_S regardless of what h function is used for the ordering. Hence, this proves that it is impossible for CE_S to satisfy all of $\boxed{1}$, $\boxed{2}$, $\boxed{3}$, and $\boxed{4}$, implying it cannot be a Clocked Election for Schulze. This is a contradiction, implying that the Schulze method cannot be reduced to a clocked election, and hence is not obviously independent of clones.

B Examples

B.1 STV Examples (with and without clones)

8 Voters	5 Voters	4 Voters	3 Voters	1 Voters
B	A	C	D	A
A	C	D	B	C
C	B	B	A	D
D	D	A	C	B

The order of elimination is as follows: [D, C, A]. The winner is B.

8 Voters	5 Voters	4 Voters	3 Voters	1 Voters
B	A	Cx	D	A
A	C	C	B	C
Cx	Cx	D	A	Cx
C	B	B	Cx	D
D	D	A	C	B

The order of elimination is as follows: [C, D, Cx, A]. The winner is B.

8 Voters	5 Voters	4 Voters	3 Voters	1 Voters
B	A	Cx	D	A
Bx	C	C	Bx	C
A	Cx	D	B	Cx
Cx	B	B	A	D
C	Bx	Bx	Cx	Bx
D	D	A	C	B

The order of elimination is as follows: [Bx, C, D, Cx, A]. The winner is B.

B.2 Example run of Ranked Pairs (with and without Clones)

Consider the following voter profile:

10 Voters	8 Voters	6 Voters	5 Voters	3 Voters	1 Voters
D	B	A	C	D	A
A	A	C	D	B	C
B	C	B	B	A	D
C	D	D	A	C	B

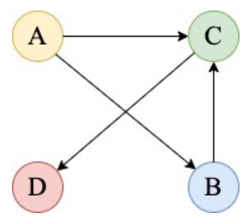
The corresponding majority matrix is:

	A	B	C	D
A	0	1	23	-3
B	-1	0	9	-5
C	-23	-9	0	7
D	3	5	-7	0

Using the majority matrix we add the nodes in the following order:

$$\{AC, BC, CD, \cancel{DB}, \cancel{DA}, AB\}$$

Where we remove directed edges DB and DA as they cause a cycle to form. This leaves us with the following graph: As node A is the source, A is declared the winner.



Furthermore, the Ranked Pairs voting algorithm is provably independent of clones; the full proof can be found in the Tideman paper [Tid87]. To illustrate some of the intuition behind this, consider the following example where C is cloned into C and Cx. This example and all other examples were generated using this simulation code.

10 Voters	8 Voters	6 Voters	5 Voters	3 Voters	1 Voters
D	B	A	Cx	D	A
A	A	C	C	B	Cx
B	C	Cx	D	A	C
Cx	Cx	B	B	Cx	D
C	D	D	A	C	B

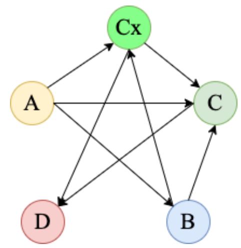
The corresponding majority matrix is:

	A	B	C	Cx	D
A	0	1	23	23	-3
B	-1	0	9	9	-5
C	-23	-9	0	-5	7
Cx	-23	-9	5	0	7
D	3	5	-7	-7	0

Using the majority matrix we add the nodes in the following order:

$$\{ACx, AC, BCx, BC, CxD, CD, \cancel{DB}, CxC, \cancel{DA}, AB\}$$

Where we remove directed edges DB and DA as they cause a cycle to form. This leaves us with the following graph: As node A is the source, A is declared the winner.



B.3 Example run of Schulze (with and without Clones)

Consider the following voter profile:

8 Voters	2 Voters	4 Voters	4 Voters	3 Voters
A	B	C	D	D
C	A	D	B	C
D	D	B	A	B
B	C	A	C	A

The corresponding pairwise matrix is:

	A	B	C	D
A	0	8	14	10
B	13	0	6	2
C	7	15	0	12
D	11	19	9	0

Which generates the strength matrix:

	A	B	C	D
A	0	(14,7)	(14,7)	(12,9)
B	(13,8)	0	(13,8)	(12,9)
C	(13,8)	(15,6)	0	(12,9)
D	(13,8)	(19,2)	(13,8)	0

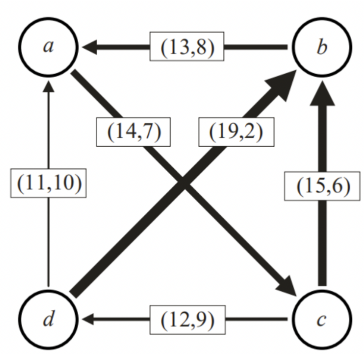


Figure 3: The graph generated by this voter profile as depicted in [Sch18]).

Therefore, as denoted by the graph, the winner is D. The Schulze voting method is also independent of clones [Sch11]. To illustrate, consider the following example where D is cloned into D and Dx.

8 Voters	2 Voters	4 Voters	4 Voters	3 Voters
A	B	C	Dx	D
C	A	Dx	D	Dx
Dx	D	D	B	C
D	Dx	B	A	B
B	C	A	C	A

The corresponding pairwise matrix is:

	A	B	C	D	Dx
A	0	8	14	10	10
B	13	0	6	2	2
C	7	15	0	12	12
D	11	19	9	0	5
Dx	11	19	9	16	0

Which generates the strength matrix:

	A	B	C	D	Dx
A	0	(14,7)	(14,7)	(12,9)	(12,9)
B	(13,8)	0	(13, 8)	(12,9)	(12, 9)
C	(13,8)	(15,6)	0	(12,9)	(12, 9)
D	(13,8)	(13,8)	(19,2)	0	(12, 9)
Dx	(13, 8)	(13, 8)	(19, 2)	(16, 5)	0

This gives us the winner D, which maintains the independence of clones.

B.4 Ascending Clock Auction vs. SPSB

This example of a 2-agent SPSB game is adapted from Li [Li17]. We will show how if the agent is cognitively limited, the agent might believe there is a useful deviation from truthful reporting. We adopt the following notation: each agent i has quasilinear utility $v_i(u_i, p_i)$ and a private value u_i that denotes their valuation for the good. Each agent's action is in the form of a bid b_i.

As illustrated in Figure 4, if agent 2 bids an amount lower than agent 1's utility (green boxes), then bidding truthfully or strategically overbidding will produce the same utility. If agent 2 bids an amount greater than agent 1's utility (orange boxes), then bidding truthfully will always lead to scenario B, with utility zero. However, strategically overbidding will either lead to scenario E, which is equivalent to scenario B, or scenario D, which is worse than scenario B. Therefore, an agent which is able to engage in contingent reasoning would decide that bidding truthfully is a dominant strategy.

However, consider a cognitively limited agent is only able to recognize a coarse partition of the scenarios, and the possible range of utilities that would result. Bidding truthfully generates the utility range $[0, u_1]$ while strategically overbidding generates the utility range $[u_1 - b_1, u_1]$. Note, that in the second case, since $b_1 > u_1$, therefore $u_1 - b_1 < 0$. Presented with these two ranges, it becomes apparent why this mechanism is not obviously strategy-proof. Since the infinum of the first range is not greater than the supremum of the second range, the first range does not obviously dominate the second range. More informally, a cognitively limited agent can envision a scenario in the partition corresponding to strategically overbidding that outperforms at least one scenario in the partition corresponding to bidding honestly. Therefore, the agent may not realize that bidding truthfully strictly dominates overbidding.

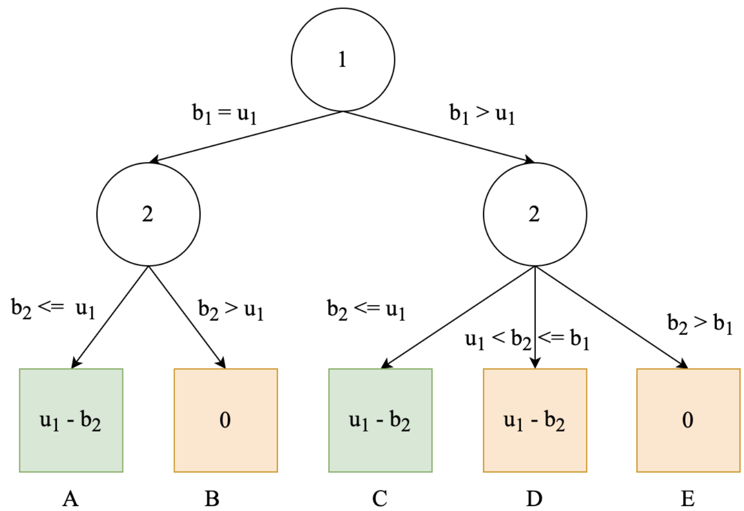


Figure 4: Second-Price Sealed Auction

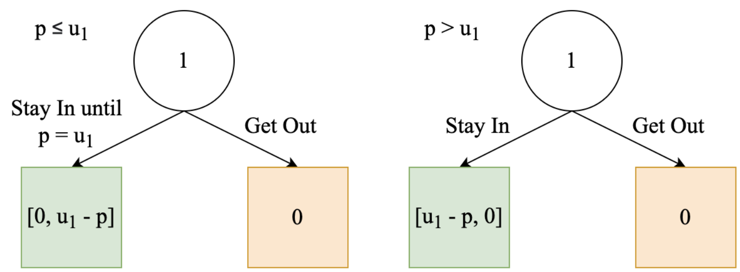


Figure 5: Agent in an Ascending Clock Auction

In contrast, the ascending clock auction *is* OSP. For an agent, there are two types of scenarios: scenarios where $p \leq u_1$ and scenarios where $p > u_1$ (Figure 5). In the first scenario, the honest report is to stay in while in the second scenario, the honest report to get out. We assume the agent begins from the honest-reporting strategy: exiting when $p = u_1$. For every deviation from this strategy, the best possible outcome is no better than the worst possible outcome from maintaining the strategy.

In the first scenario, getting out grants a utility of zero. Staying in until $p = u_1$ can grant utility no worse than 0 since the agent either loses (utility 0) or wins (utility $u_1 - p \geq 0$). Therefore, the infinum of staying in (0) is weakly greater than the supremum of getting out (0).

In the second scenario, getting out grants a utility of zero. Staying in grants at most zero utility. Since winning at this point will grant $u_1 - p$ which is always less than zero. Losing in a future round grants zero utility. Therefore, the supremum of deviating by staying in is zero, which is weakly dominated by getting out (utility 0).

Therefore, ascending clock auctions are OSP.

B.5 Simulation Code

The link to a Github containing our simulation notebook can be found here.