Notes: updated version of tradeoffs on voting and incentive systems

March 11, 2021

0.1 Voting

0.1.1 Voting Process

Jurors assess evidence that has been submitted, typically by the parties to the dispute, and are provided with court policies, comparable to juror instructions[1], on how they should reason based on that evidence. Then jurors commit [4] their vote to one of the options. They submit $\text{hash}(\text{vote}, \text{salt}, \text{address})$[2]. The salt is a random value generated locally in order to add entropy to prevent the use of rainbow tables[3]. When the vote is over, they reveal $\{\text{vote}, \text{salt}\}$, and a Kleros smart contract verifies that it matches the commitment. Jurors failing to reveal their vote are penalized, see Section 0.2.

After a juror has made a commitment, her vote cannot be changed. However, it is still not visible to other jurors or to the parties. This prevents the vote of a juror from influencing the votes of others.

Jurors can still declare that they voted in a certain way, but it is challenging for them to provide other jurors a reason to think that what they say is true. This is an important feature for the Schelling Point to arise. If jurors knew the votes of other jurors, they could vote like them instead of voting for the Schelling Point[4].

As this two step processes of committing and then revealing one's vote requires additional user interactions, in some low stakes courts, one might want

[1] These policies vary by court, see Section **??** and can be changed by the governance procedure, see Section **??**.

[2] Throughout this paper we use hash referring to a cryptographic hash function, in Ethereum the one used is keccak256.

[3] As currently implemented in the existing Kleros interface, the salt is generated by having the user use the Ethereum private key stored in Metamask to sign a block of text that includes an identifier for the specific dispute. Hence, the signatures on this text with different keys will be different and are computationally infeasible to obtain without the user's private key under the assumption of the security of the signature algorithm. Thus, a juror is prevented from copying the commitment of others. Nevertheless, if a user deletes her salt from her local memory, she can regenerate it as long as she retains access to her private key.

[4] See the discussion on future work in Section **??** for a further discussion of these ideas.

votes to be issued publicly to simplify the user experience[5]. Which system is used is determined via a court parameter, see Section **??** on governance.

0.1.2 Vote Aggregation

After all jurors have voted (or after the time to vote is over), votes are revealed by jurors. Jurors that fail to reveal their vote are penalized. Finally, votes are aggregated according to a predetermined voting rule resulting in an option that is considered the winner.

Current Voting System When jurors are presented with a binary choice, it is natural to use the Plurality, or "first-past-the-post", voting system[6]. Currently, Kleros employs the Plurality system even when there are more than two choices. (Specifically, we adopt the choice that wins a plurality of the vote in the last appeal, see Section **??**).

When there are more than two options, under a Plurality voting system, the following may occur:

- If there are many very similar honest options (or "clones"), they will divide the votes of jurors that are attempting to vote honestly, decreasing the probability that any one of them wins. Anticipating this effect, jurors might instead vote for a distinguished but dishonest choice. For example, imagine that in our contractor use-case above there were several different options that gave Bob another week, another eight days, or another nine days respectively. Then the collective odds of any of these options being chosen would likely fall below the odds of a single "give Bob more time" option, see Figure 1.
- To the degree that no single option is likely to receive more than 50% of the votes, this lowers the bar for the number of votes that attackers need to corrupt to pass a dishonest result, see Figure 2.

Considering these issues, one might expect Plurality voting to still produce generally "honest" results when one single choice has a very clear, winning case, which essentially binarizes the choice. However, Kleros should be able to cope with nuanced cases involving many choices.

Social Choice Theory and Future Voting System(s) In this section, we consider a number of desirable properties for a vote aggregation rule in a system

[5]Note that as Kleros uses an appeal system, even if a majority of votes in a voting round have already been cast for a given choice, voting for that choice does not guarantee coherence with the ultimate result used for token redistribution, see Section 0.2. This limits the effectiveness of a vote copying strategy. Hence public votes might be acceptable in some cases.

[6]Plurality, or "first-past-the-post", is the voting system that in which voters express a vote for only one candidate, and then the candidate that receives the largest number of votes is selected, even if this candidate does not receive a majority of the total votes due to there being more than two candidates.

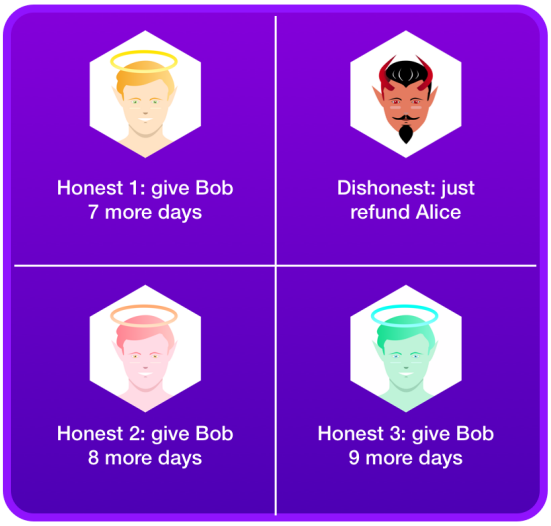

Figure 1: In the Plurality voting system, jurors can only vote for one option. Particularly, a voter cannot cast a vote such as "take one of the more time options, it doesn't matter which". Then if jurors are presented with a collection of similar, honest choices along with a single dishonest choice, the dishonest choice may seem distinguished and become the Schelling point.

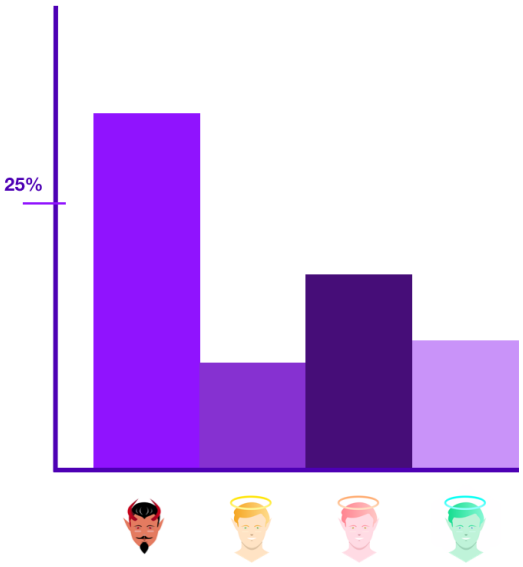

Figure 2: If the vote is split between several “honest” options, the attacker does not need to corrupt 50% of the vote in a Plurality system to have a malicious option adopted.

such as Kleros, remarking which rules satisfy or do not satisfy these properties. Note that for some of the properties we consider, an analysis of how a given vote aggregation rule performs will depend on the choice of incentive system that we discuss in Section 0.2. Indeed, the choices of voting aggregation rule and incentive system are deeply interwoven and should be considered together. Also note that, while some of the properties we consider have standard definitions, others will be somewhat subjective.

Ideally, the vote aggregation system should have the following properties:

- Clone independent - From the point of view of a voting system, this is a standard property considered in social choice theory that informally means that having a "clone set" of multiple similar options does not increase or decrease the winning chances for other options outside of the clone set, see for example [15] for a formal definition. Using a clone independent aggregation rule allows parties to produce arbitrable contracts without having to worry that presenting two or more similar options to the jurors might lead to vote splitting in or against their interests. Examples of clone independent voting systems are Instant-Runoff, Ranked Pairs, and Schulze. The results in the clone independence column of the table below are all presented in [15] and [14]. Note, however, that in the situation we are considering, a more robust notion of clone independence would demand that not only the voting system be clone independent, but that also the addition of a clone not affect the expected payoffs that voters receive, or at least not affect which votes provide the optimal payoffs, up to reordering of alternatives within a clone set. It is not yet clear what the appropriate formulation of this property should be; indeed demanding that

  $$\text{payoff}(v' \in \mathcal{L}(A \cup \{\text{clone}\})) = \text{payoff}(v \in \mathcal{L}(A))$$

  whenever v' reduces to v upon condensing a clone set is problematic as the values given by most of the payoff functions we consider (see Section 0.2.2) are discrete and would be altered by the addition of any new alternative. Nonetheless, we have analyzed these systems heuristically/numerically and we have noticed that IRV type systems, while being clone independent as voting systems, introduce a bias in the rewards of clones for the types of payoff systems we consider. Indeed, note that if a voter highly ranks the clone set, then under an IRV type system which clone the voter ranks first has more of an influence on which clone wins than if the voter ranks the clone set lower. Consequently, as our incentive systems reward or penalize voters based on where they place the ultimate winner relative to other choices, voters in these systems that rank the clones highly capture a slightly larger percentage of rewards than they do before the addition of a clone. We have, at least heuristically, not observed such a bias for voting systems such as Ranked Pairs and Schulze, where the aggregation rule is based on the graph of the relative strengths of pairwise duels.

- Satisfy the Condorcet criterion - A voting rule is said to be a Condorcet

method if whenever there is an option w such that more voters rank w higher than a for all other options a, then w wins, see [3]. Note, however that such a "Condorcet winner" does not necessarily exist for all sets of votes expressed by jurors. The idea that, if there is an option that wins "head-to-head" against all others, then that option should be selected, is fairly intuitive. Hence, if Kleros cases often have Condorcet winners, then they are particularly straightforward to mentally simulate for jurors. Moreover, the Condorcet criterion, namely to prefer options that have a consensus of the population against each other option, corresponds to certain notions of "fairness". Moreover, we see that the Condorcet property has positive effects on attack resistance when combined with the types of incentive systems that we consider in Section 0.2.2. Indeed, an attack that attempts (and fails) to dislodge a Condorcet winner by reversing a duel between that alternative and another is something that is well captured, and hence penalized, by these incentive systems. Ranked Pairs and Schulze are examples of Condorcet methods. WoodSIRV is essentially a version of IRV that is Condorcet-ized by checking at each voting round whether there is a Condorcet winner and selecting it if available.

- Resistant to attacks - Note that attack resistance is essentially economic. For example, one wants the number of votes that would need to be changed to result in a "dishonest" outcome winning to be high, with the rationale that this increases the cost of attacks, such as bribes that would be necessary to change those votes. Hence, attack resistance touches on certain standard voting systems properties that have been studied in social choice theory, such as the "later no help" and the "participation" criteria [3], in combination with analysis of the penalties and rewards laid out in the incentive system, see Section 0.2. While one can exhibit attacks on individual systems, it is challenging to produce proofs that no attacks exist[7] The discussion in Section 0.1.2 already shows that the bar for a 51% attack is lowered on the Plurality system in cases where no single (honest) option receives more than 50% of the votes. In the Borda system, if the "honest" answer is a and an attacker wants the option b to win, she can submit many votes where a is ranked first and b is ranked second, and a small number of votes where b is ranked first and a is ranked last. Using the incentive system that will be described in Section 0.2, this attack has limited cost and risk for the attacker whereas for appropriate choices of parameters, Instant-Runoff and Ranked Pairs seem to be more resistant[8].

[7] Hence the question marks on the claims of better attack resistance for Instant-Runoff, WoodSIRV, Ranked Pairs, and Schulze in the table below.

[8] All of our claims on attack resistance should be viewed as based on using the incentive system of Section 0.2. It is possible with another incentive system our conclusions would be different. Also note the role of β discussed in Section 0.2; for $\beta = 0$ in weighting systems 2 and 3, the economic cost in lost deposits of the attack described on Borda is comparable to that of an attack on Instant-Runoff where the attacker bribes many voters to place b first and a second, though the attack on Instant-Runoff requires convincing a large number of jurors to accept a small in-protocol penalty, and the attack on Borda involves convincing a small

It is also worth keeping in mind which types of manipulative behaviours the incentive systems we consider in Section 0.2.2 are capable of detecting and penalizing so as to avoid choosing an aggregation function that is vulnerable to manipulation that the incentive system does not penalize. We have already noted above that an attack that prevents a would-be Condorcet winner from winning is well detected by the incentives we consider. On the other hand, according to Arrow's Impossibility Theorem, for all non-trivial voting systems, there will be situations where the rankings of "irrelevant alternatives" will affect the results. Manipulations involving such irrelevant alternatives are not well captured by the incentive systems we consider, which is forced upon us in exchange for the winner first, maximum payout property of the incentive system that we discuss below. However, note that as Instant-Runoff satisfies Later No Harm and Later No Help, such manipulations are only relevant to the degree that they involve alternatives that are already ranked higher than the alternative that would otherwise win. Then, in this system, one would expect that in order to produce a dishonest answer, the attacker would need to corrupt many votes willing to place a dishonest answer over the "honest" answer. Thus, we argue that voting systems such as WoodSIRV, which are Condorcet, but resolve Condorcet paradoxes with an IRV like step, seem to offer the best available attack resistance as they combine these layers of defense, requiring an attacker to bear the cost of preventing a Condorcet winner if one exists, and then to the degree that a winner is manipulated once/when finds oneself in a Condorcet paradox, only manipulations by voters who already receive a significant penalty are useful, limiting effective attacks to something heuristically resembling the coalition required for a 51% attack[9]

number of jurors to accept a large penalty. However, with other choices of β, this attack on Instant-Runoff becomes more expensive while the attack on Borda does not.

[9]We briefly attempt to illustrate the differences in the attack space between WoodSIRV and Ranked Pairs, making use of examples. As both of these systems are Condorcet, if there is an "honest alternative" w, in both cases to change the result one must first manipulate enough votes to reverse at least one duel involving w. If reversing this duel results in a new Condorcet winner being selected, this same manipulation has the same result in the two systems. Or course, there are other possibilities for the profile of votes where one or the other voting system requires more manipulated votes to change the result. As an example of a situation where more manipulation is required to change the result in Ranked Pairs, consider the voting profile given by 2 votes for $\{a > b > w\}$, 5 votes for $\{a > w > b\}$, 8 votes for $\{b > w > a\}$, and 6 votes for $\{w > a > b\}$. Then w is the Condorcet winner, but already a single manipulation changing a $\{a > w > b\}$ vote to $\{a > b > w\}$ results in a Condorcet paradox where w actually receives the fewest first place votes and is eliminated under WoodSIRV. In order to change the result under Ranked Pairs, the attacker could take a few different strategies: 1) she could bribe/try to convince more votes to switch from $\{a > w > b\}$ to $\{a > b > w\}$ to strengthen the pair $b > w$, 2) she could bribe/try to convince more votes to switch for example from $\{w > a > b\}$ to $\{w > a > b\}$ to weaken the pair $a > b$, or 3) some mixture of these strategies. While the changes in the first strategy result, to the degree that they do not successfully change the result from w, result in their voters incurring losses, the voters who follow the second strategy and vote $\{w > a > b\}$ still get the maximum payoff if w remains the winner. As such, it might be more viable for an attacker to bribe or convince voters who would normally vote $\{w > b > a\}$ to "micro-cheat" by voting $\{w > a > b\}$ than it would to convince voters to manipulate their

- Not require too much code complexity nor gas - Even on a scaled version of Ethereum, gas costs would likely remain a usability concern. Moreover, in any smart contract platform, reducing code complexity is beneficial in avoiding the opportunities for bugs. For all of the voting systems that we consider here, the winner can be calculated in polynomial time. However, voting systems that use an IRV type, round based structure would seem to be easier to implement than those that require graph algorithms such as Ranked Pairs and Schulze.

- Resistant to having too many "Schelling-dishonest scenarios"/Monotonic - We have seen in our work in [7] that, for any reasonable voting system and incentive system, there will always be some rare situations where a voter is incentivized to deviate from what we call "Schelling-honesty", namely she is incentivized to cast a vote that does not actually reflect an the order in which she thinks alternatives are likely to win. This work develops upon classic impossibility theorems in social choice theory such as Arrow's Impossibility Theorem and the Gibbard-Satterthwaite Theorem. However, in that work we saw that the situations in which different voting systems failed Schelling-honesty were more contrived and seemingly less likely to occur in practice for some voting and incentive system pairs than for others. Notably, when a voting system is monotonic, the types of failures of Schelling-honesty are more limited.

The following table summarizes how selected voting systems perform according to these criteria:

---

votes in other situations. On the other hand, consider the voting profile given by 5 votes for $\{a > b > w\}$, 2 votes for $\{a > w > b\}$, 5 votes for $\{b > w > a\}$, 7 votes for $\{w > a > b\}$, and 2 votes for $\{w > b > a\}$. Then an attacker could change the outcome under Ranked Pairs from voting for w as a Condorcet winner to selecting b in a Condorcet paradox by convincing 2 $\{w > a > b\}$ votes to vote for $\{b > w > a\}$. In this case, WoodSIRV would still select w as the winner. Indeed, an attacker that wanted to change the outcome under WoodSIRV would have several available strategies: 1) bribe/convince enough votes to switch their rankings of a versus b such that b becomes a Condorcet winner, 2) Convince/bribe some of the votes who place a first to place w or b first so that a is eliminated in the IRV step, and b wins the duel with w. For example, the attacker might bribe an additional voter to change a vote from $\{a > b > w\}$ to $\{b > a > w\}$. Both of these strategies might involve transposing pairs that are not weighted by the incentive system if w wins (indeed according to Arrow's Impossibility theorem this is inevitable for an incentive system that has the winner first, maximum payout property, see below). If the attacker can successfully follow the first strategy, she could change the result of the vote under any Condorcet system. However, following the second strategy either requires transposing pairs that involve w, and hence taking penalties if w ultimately wins, or transposing pairs of alternatives that are already ranked both ahead of w. Hence, this limits the ability of the attacker to corrupt voters to make such transpositions to the limited set of voters that already bury (the "honest choice") w and hence can already be thought of as being part of an attack coalition anyway.

	Clone Independent	Complexity/ Gas	Condorcet	Monotonic	Attack Resistance
Plurality	No	Low	No	Yes	Bad
Borda	No	Low	No	Yes	Not great
Instant-Runoff	Yes as voting system Bias in incentive system	Medium	No	No	Better?
WoodSIRV	Yes as voting system Bias in incentive system	Medium+	Yes	No	Better+?
Ranked Pairs	Yes as voting system No known bias in incentives	High	Yes	Yes	Better?
Schulze	Yes as voting system No known bias in incentives	High	Yes	Yes	Better?

When jurors are presented with a binary choice, as is the case of most current applications of Kleros[10], these voting systems are equivalent. Indeed, such voting between two options satisfies all of the "good" properties in this table. Hence these comparisons are only relevant when there are at least three possible outcomes.

Finally, note that while the design choices of Kleros are motivated by the particular challenges of being attack resistant in a setting without trusted authorities, the properties considered above are potentially relevant beyond blockchain applications to other crowdsourced platforms that, like Amazon's Mechanical Turk [1], require an aggregation of user feedback with users who may provide incorrect or spam answers to minimize the effort required of them. Hence some of these ideas could potentially be used to improve the design of such systems in the spirit of [10].

0.2 Incentive System

Users are incentivized to become Kleros jurors as this gives them the opportunity to receive a portion of the arbitration fees paid for the dispute, as discussed in Section **??**. As part of the incentive system that encourages jurors to provide honest rulings, in addition to the potential to gain arbitration fees, jurors can also lose some of their PNK stake for rulings that are out-of-line with those of the other jurors. These lost PNK stakes are then redistributed to other, more coherent, jurors as will be described below. Thus jurors are participating in a Schelling game similar to those described in Section **??**.

In order to be drawn in a given court, users are required to stake a minimum amount of tokens, denoted by min_stake. Then, regardless of how a juror votes on a case, the number of tokens that she can lose from her stake per vote is limited to a fixed percentage of this minimum stake. This percentage will be

[10]In fact, all Kleros disputes also have the possible outcome that jurors vote "refuse to arbitrate", see Section **??**. However, it is expected that this option will rarely be voted for, so a case with two other possible outcomes are de facto binary with only those two outcomes having plausible chances of being voted for.

denoted by α. However, was observed in Section **??** in the discussion of the "weight", a single juror can be drawn multiple times for a given case, giving her more votes on this case. Then the maximum amount that the juror can lose as a result of her vote increases corresponding to this weight. Namely, the maximum amount of tokens than can be lost on a given case per juror is:

$$D = \alpha \cdot \text{min\_stake} \cdot \text{weight}.$$

Both the α and the min_stake parameters are defined by the governance mechanism and can vary from one court to another.

0.2.1 Current Token Redistribution Model

Currently, in parallel with the first past the post voting system that we described in Section 0.1.2 (in which particularly jurors do not provide ranking of the options other than a single vote), any juror that does not select the outcome w that wins the last appeal round loses her deposit D. Then jurors that do vote for w receive a payment of:

$$\frac{\text{ETH fees and lost deposits}}{\text{\# jurors that vote for } w}.$$

These calculations (i.e. how many deposits were lost and how many jurors voted for w) are done on a round-by-round basis.

0.2.2 Future Token Redistribution Model

Take w to be the option that wins via the vote aggregation methods described in Section 0.1.2. We speak about jurors voting "coherently" if they agree with the ultimate vote outcome; while being coherent is an all or nothing property for binary decisions, for non-binary decisions jurors' votes can be more or less "coherent". The goal is to incentivize users to place outcomes they believe to be "honest" high in their lists following the motivations of Section 0.1.2. Conversely, one also wants to strongly penalize a juror who has placed the winning choice w far down on her list. One option would be to have jurors lose:

$$\frac{\text{\# options ranked above } w}{\text{\# total options} - 1} D \tag{1}$$

to be redistributed between other jurors based on their coherence. However, in this framework, an attacker that ranks a malicious choice first and w second will risk relatively little of her deposit.

In equation 1, one rewards the jurors for the number of options a_i that they correctly place below the winner w with each a_i given the same weight.

Alternatively, one can give extra weight for rewards and penalties for options a_i for which the margin of pairwise votes between w and a_i is particularly close. This is in the spirit that *narrowly* failed attacks should be particularly expensive, which is a common goal in the design of blockchain-based platforms [5]. If an attacker is attempting to commit a bribing attack so that a would-be Condorcet winner w^* no longer wins, this requires a sufficient number of bribes so that at least some a_i defeats w^*. Hence at least one pair must pass from the honest winner winning to not, so in narrowly failed attack this pair will be weighted heavily.

Namely, one might take weights $w = (i) = w(a_i) \in [0, 1]$ for all $a_i \neq w$, such that $\sum_{a_i \neq w} w(i) = 1$.

Now we have voters lose:

$$D \sum_{a_j \neq w} \mathbf{1}_{\text{voter voted } a_j > w} \cdot w(j)$$

from their deposit D and receive redistributions of the form:

$$\frac{\text{ETH fees and lost deposits}}{\sum_{\text{voter}_k \in \mathcal{V}} \sum_{a_j \neq w} \mathbf{1}_{\text{voter}_k \text{ voted } a_j < w} \cdot w(j)} \sum_{a_j \neq w} \mathbf{1}_{\mathcal{USR} \text{ voted } a_j < w} \cdot w(j), \tag{2}$$

where $\mathcal{V}$ is the set of voters in the same voting round as $\mathcal{USR}$.

Under this payoff mechanism, regardless of which weight function $w(i)$ is chosen, the lower jurors place the winning outcome, the larger the portion of the deposit they lose and the less arbitration fees they receive. Indeed, if a juror places the ultimate winning outcome last, she receives no arbitration fees and loses her entire deposit, which is split between other jurors in accordance with how high they ranked w[11]. The above formulas do not involve a division by zero unless all voters in a given round place the winning outcome behind all other (non-zero weighted) alternatives, in which case all voters lose their deposit and do not receive a payout[12].

For the choice of weight function, we have a series of tradeoffs and criteria similar to that of Section 0.1.2 for voting systems. Indeed, while we have divided this discussion into two sections for the sake of exposition, one for the voting system and one for the incentive system, the two choices can in some cases have interactions and should be considered together.

To illustrate the tradeoffs around a choice of weight function, we present a few of the weight functions that we have considered. In all cases, we will have

[11] The redistribution mechanism is inspired by the SchellingCoin, see Section **??**, where jurors gain or lose tokens depending on whether their vote was consistent with the others jurors. Note that token redistribution mechanisms are still being actively researched and may further evolve in future versions.

[12] If at one level no one voted coherently, what to do with the amounts from that level can be determined by the governance procedure. See the descriptions on future work in Section **??** for further discussion on this point.

an adjustable parameter $\beta \geq 0$, chosen via the governance mechanism that will control how concentrated the weights are on the closest pairs[13][14]

- Weight function 1 (constant weights):

$$w(i) = \frac{1}{\#\{a_i \in A : a_i \neq w\}},$$

- Weight function 2:

$$w(i) = \frac{\left(\frac{1}{|\text{margin of } a_i \text{ against } w|+1}\right)^{\beta}}{\sum_{a_j \neq w}\left(\frac{1}{|\text{margin of } a_j \text{ against } w|+1}\right)^{\beta}},$$

- Weight function 3:

$$w(i) = \frac{\left(1 - \frac{|\text{margin of } a_i \text{ against } w|}{\text{total number of votes}}\right)^{\beta}}{\sum_{a_j \neq w}\left(1 - \frac{|\text{margin of } a_j \text{ against } w|}{\text{total number of votes}}\right)^{\beta}}$$

- Weight function 4:

$$w(i) = \frac{1 - \left(\frac{|\text{margin of } a_i \text{ against } w|}{\text{total number of votes}}\right)^{\beta}}{\sum_{a_j \neq w} 1 - \left(\frac{|\text{margin of } a_j \text{ against } w|}{\text{total number of votes}}\right)^{\beta}}$$

	Winner First Max Payout	Unanimous No Weight	Resistance to Manipulability	Concentration on Close Pairs
Weight function 1	Yes	No	Yes	No
Weight function 2	Yes	No	-	Not great
Weight function 3	Yes	Yes	Worse	Better
Weight function 4	Yes	Yes	Better	Better

- Winner first gives maximum payout - Note that all of our weight functions have this property, indeed it follows from the redistribution structure given in equation 2. However, one could imagine payoff structures that

[13] Note that one might wish to choose different β in a way that depends on the dispute round as in early rounds, the number of jurors is small enough that which a_j are narrowly decided and hence which are weighted heavily is more variable, and hence to some degree arbitrary. For example, it can be helpful to take margins in the weight functions below drawn from the results of the last appeal round, when they will be the most representative of community sentiment.

[14] Further note that none of the weight functions considered below depend explicitly on which voting system we chose. In future work, one might incorporate into a weight function further information from the vote aggregation process, such as which round an alternative is eliminated in for an IRV type system.

depend not only on how voters rank the winning choice w compared to other choices, but also on the relative positions that voters give to two non-winning alternatives, i.e. whether a voter ranks $a > b$ or $b > a$ for some $a, b \neq w$. From a user experience perspective, we consider it important that users not be obliged to rank what they consider to be irrelevant choices, hence in all of our candidate systems here the relative rankings of such choices do not have weight. However, not that this choice fundamentally limits the ways in which a voting system+incentive system can be attack resistant. Indeed, by Arrow's Impossibility Theorem, for any non-trivial voting system there will be situations where the relative rankings of "irrelevant alternatives" will affect the result. For any incentive system that satisfies the winner first gives maximum payout property, a voter's ranking of these "irrelevant alternatives" does not affect her payout, so the threshold to "micro-corruption" attacks, for example bribing such a juror to invert these seemingly less relevant alternatives, may be lower. See our comments in Section 0.3 on why we think this effect is mitigated, while nonetheless present, in the voting and incentive system we propose.

- Unanimous pairs given no weight - Generally, the weight functions above (with the exception of the constant weight function) are such that a_i is given more weight than a_j if the duel between w and a_i is closer than that between w and a_j. Taken to the extreme, one might hope that duels that are unanimous are given no weight, to not dilute the effect of the payoff system on incentivizing voters to make honest rankings on pairs that are more likely to matter. Indeed, for weight functions that have this property, if a non-binary case is "defacto binary", i.e. there are two alternatives a and b such that all voters rank a and b first and second in some order, then (assuming that the voting system actually selects a or b as the winner, which would be the case for any of the voting systems considered in 0.1.2) only the duel between a and b would receive weight in the incentive system and this situation reduces to the incentives we have analyzed in binary cases.

- Resistant to manipulability for fixed winner - By the impossibility theorems that we have seen in [7], we know that for any non-trivial voting and incentive system, there will be situations where a voter can improve her payout by manipulating her vote so that it no longer reflects her expectations of which alternatives are more or less likely to win. Nonetheless, one might hope that one could choose a payoff system so that, conditional on not changing the winner determined by the aggregation rule, a voter can never improve her payoff by ranking this winner lower. Indeed, the result we have in Proposition 1 gives a result in this sense for the constant weights. However, we have so far not found any weighting function that satisfies such a non-manipuability property where also unanimous decisions are given no weight[15] Nonetheless, we have made some numer-

[15]Indeed, we have some preliminary work that shows that, while not necessarily being

ical and heuristic observations, of which the results are shown in Figure 3, that indicate that weight function 4 tends to offer fewer situations in which such manipulation would be profitable compared to other weight functions, notably compared to weight function 3. This is particularly true for choices of β that emphasize the closes pairs.

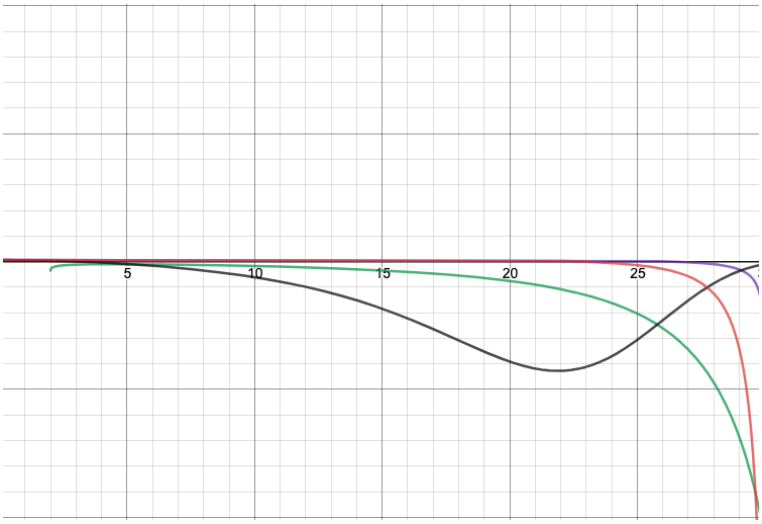

Figure 3: Lower bounds on the loss that an attacker suffers by transposing the winning alternative w with another alternative a while not changing the result. The x-axis represents the margin of the duel between w and a between 0 and M. Negative lower bounds correspond to situations under which such an attack might be profitable. The black and purple lines correspond to weight function 3 with $\beta = 3$ and $\beta = .33$ respectively. The red and green lines correspond to weight function 4 with $\beta = 3$ and $\beta = .33$ respectively. Hence, we see that weight function 3 demonstrates more extreme phenomena, performing quite well for choices of β that provide relatively little concentration on the closest pairs, but offering substantially larger potential for attack for choices of β that provide more concentration.

- Concentration of weight on closes pairs - As previous discussed, one wants to weight closer pairs more to increase attack resistance. However, one still wants to give some weight to other pairs to encourage voters to take their entire vote (or at least the rankings between all alternatives that they think have a meaningful chance of winning) seriously. For now at least our evaluations of different weights on this criterion are largely heuristic: for different values of β and common vote splits (e.g. 2-1, 5-2, or 12-3) how much weight is given to each alternative, and how the shape of these

impossible that there exist weighting functions that satisfy both of these properties, any weighting function that did must be significantly constrained in its form.

weight functions varying in β influences the flexibility of the governance process to adjust.

In the following proposition, one sees that this payoff system can have good properties with respect to incentivizing jurors to rank candidates who are likely to win higher, corresponding to the objectives laid out in Section 0.1.2.

Proposition 1. *Consider the incentive system above with the constant weights, i.e. weighting function 1. Suppose that a given voter has a probabilistic prior for the outcome of the dispute resolution process, i.e. she estimates probabilities for the votes of other jurors and for the probabilities of each outcome to win possibly after appeals, such that:*

- *she believes that the votes of other jurors in her voting round are independent of her vote,*
- *she believes that the outcome is independent of her vote and the votes of other jurors in her voting round,*
- *she assigns to the possible outcomes $a_1, \ldots, a_n$ probabilities $prob(a_1), \ldots prob(a_n)$ of ultimately winning.*

Then a weakly dominant strategy for this juror is to rank the outcomes a_j from highest to lowest by their chance of winning $prob(a_j)$.

See Appendix **??** for a proof of this result. Note that the perspective of a juror believing that her vote will not change the ultimate outcome can be justified in our setting if jurors believe that incorrect outcomes are likely to be appealed.

After Kleros has reached a decision, tokens are unfrozen and redistributed among jurors. An example of token redistribution is shown in Figure 4. Note that jurors could fail to reveal their vote. To disincentivize this behaviour, the penalty for not revealing one?s vote is at least as large as the penalty for voting incoherently. This incentivizes jurors to always reveal their vote. In case of appeal, arbitration fees and tokens are redistributed at each level according to the result of the final appeal.

When there is no attack, parties are incentivized to vote what they think, other parties think, other parties think? is honest and fair. In Kleros, the Schelling Point is honesty and fairness. One could argue that those decisions being subjective (for example, compared to a SchellingCoin mechanism for a prediction market), no Schelling Point would arise. In [13], the informal experiments run by Thomas Schelling showed that in most situations a Schelling Point plebiscited by all parties does not exist. But Schelling found that some options were more likely to be chosen than others. Therefore even if a particularly obvious option does not exist, some options will be perceived as more likely to be chosen by others parties and will effectively be chosen. We cannot expect jurors to be right 100% of the time. No dispute resolution procedure could ever achieve that. Some times, honest jurors will lose coins. But as long as overall

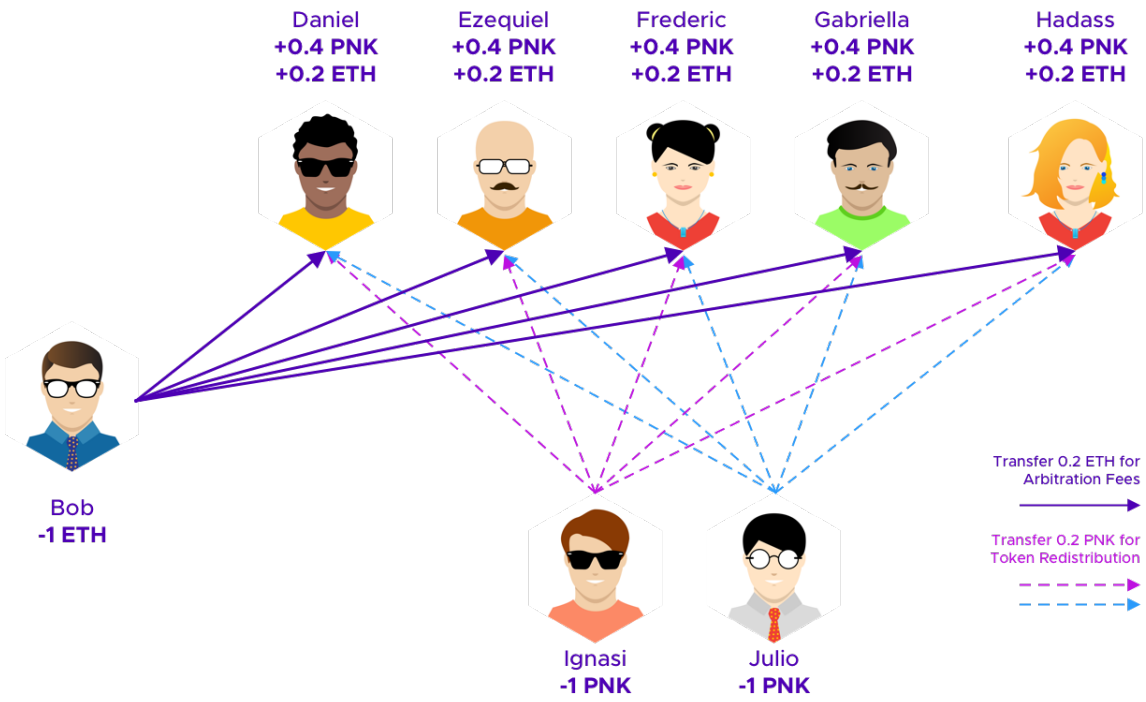

Figure 4: Seven jurors have a binary choice between ruling on behalf of Alice or on behalf of Bob. Tokens are redistributed from jurors who voted incoherently to jurors who voted coherently. Bob lost the dispute and pays the arbitration fees. The other deposits are refunded.

they lose less value than what they win as arbitration fees and as coins for other incoherent parties, the system will work[16].

Remark 1. *Above we saw that the redistribution of arbitration fees and lost deposits is handled by round. Note that if a given voter then knows or suspects that other voters in her round have voted "incorrectly", this gives her more of an incentive to vote honestly. In the extreme, a single juror that agrees with the final outcome in a round where every other juror disagreed would receive all the arbitration fees and lost deposits for that round. We call this phenomenon the "lone voice of reason" effect. We will note further implications of this effect below.*

[16]Indeed, note that, so far, this idea has largely worked as expected in experiments such as those considered in [6] as well as in practical applications such as those of [9] and [11].

0.3 Proposal for New Voting+Incentive System

Weighing these considerations, I propose we use:

- the voting system WoodSIRV (i.e. one uses the rankings provided to simulate a series of voting rounds, eliminating the alternative with the fewest first place votes in each round, as in IRV, but before each round one checks if there is a Condorcet winner among the remaining alternatives), and
- weight function 4, namely

$$w(i) = \frac{1 - \left(\frac{|\text{margin of } a_i \text{ against } w|}{\text{total number of votes}}\right)^{\beta}}{\sum_{a_j \neq w} 1 - \left(\frac{|\text{margin of } a_j \text{ against } w|}{\text{total number of votes}}\right)^{\beta}},$$

  where β is an adjustable parameter (that ideally could take an arbitrary positive value). Particularly, the weight function should ideally be calculated based on the margins from the last voting round (in order to be based on the largest, most statistically significant sample) when calculating the rewards for voters in earlier rounds.

Note that there are a number of subtle implementation points that one needs to consider. We briefly discuss these here.

- Ties in voting aggregation: In any voting system one needs a way to break ties. Here the simplest thing to do would be
  - When checking whether an alternative is a Condorcet winner, it needs to *strictly* win all of its duels, otherwise one continues as if one is in a Condorcet paradox.
  - If there is a tie for the last alternative, i.e. the alternative to be eliminated in an IRV step, all tied alternatives are eliminated.
  - If ever all alternatives are eliminated, then one returns refuse to arbitrate/a tie between the last alternates to be eliminated, similar to the existing possibility for the current version of Kleros to result in ties.

  Strictly speaking, this tie-break system invalidates a few of the properties that WoodSIRV is marked as having in the tables above, at least in edge cases (particularly its Clone Independence as a voting system). However, this is probably an acceptable sacrifice to avoid having a more complicated tie-break system. In some sense, from the perspective of the voting system having good properties, the ideal thing to do is to use a random number generator to break any ties that might occur. If whatever randomness solution we use makes this viable, it might be a worthwhile approach to consider.

- Effect of ties on payoffs: Note that the payoff system is very centered around margins of duels between the winning alternative and other alternatives. In the event of a tie, one might want to average the payoffs that would be given if each of the tied alternatives had won. Note that this would require $O(\#\text{tied alternatives})$ gas. Other approaches to this edge case might also be viable.

- Acceptable variations on WoodSIRV: There are a number of slight variations on this idea of an IRV system that makes checks for Condorcet winners that have been studied in the social choice theory literature. For example, other alternatives are Smith IRV or Tildeman's Alternative, which involve calculating the Smith set at various points in the algorithm. Another possibility is just to apply a single Condorcet winner check at the beginning and not between each IRV round. These voting systems have largely similar properties and if we decide that one is significantly better than the others from a code complexity/gas perspective, we could use that.

- Griefs based on large numbers of alternatives and optimistic checks: One asks what happens if a hostile party attempts to grief Kleros by creating a dispute with a very large number of alternatives. Let M be the number of voters and n be the number of alternatives. Then the step that checks whether there is a Condorcet winner, for example, would take $O(Mn^2)$ resources. However, if one is worried about gas consumption when there are large numbers of alternatives one could incorporate optimistic elements in the execution step that aggregates votes, e.g. whenever one needs to check if there is a Condorcet winner, one can

  - (spend the full amount of gas to) call a function in the smart contract that checks whether there is a Condorcet winner directly, or
  - submit oneself a candidate Condorcet winner that the contract checks taking $O(Mn)$ resources, if this candidate turns out to actually be a Condorcet winner, the contract updates accordingly, otherwise it reverts.
  - If enough time passes in this stage without a Condorcet winner being found, one way of another, then one can submit a call that proceeds the vote aggregation to the next step, based on the assumption that there is no Condorcet winner.

  Similar approaches could also reduce the resource consumption of the IRV steps. Alternatively, though somewhat limiting, one could simply impose a limit on the number of alternatives that Kleros would consider.

References

[1] Amazon mechanical turk. `https://www.mturk.com/`.

[2] Kenneth Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58, 1950.

[3] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, New York, NY, USA, 1st edition, 2016.

[4] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, October 1988.

[5] Vitalik Buterin. The p + epsilon attack. `https://blog.ethereum.org/2015/01/28/p-epsilon-attack/`, 2015.

[6] William George. Doges on trial curated list observations part 2 - deep dive edition. `https://blog.kleros.io/cryptoeconomic-deep-dive-doges-on-trial/`, 2018.

[7] William George. Voting systems for multiple choice Schelling games. `https://github.com/kleros/research-docs/blob/master/multiplechoiceschelling/multiplechoiceschelling3.pdf`, 2020.

[8] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973.

[9] Stuart James. Kleros TCR - a deep dive explainer. `https://blog.kleros.io/kleros-ethfinex-tcr-an-explainer/`, 2019.

[10] Andrew Mao, Ariel D. Procaccia, and Yiling Chen. Better human computation through principled voting. In *AAAI*, June 2013.

[11] Jack Peterson and Joseph Krug. Augur: a decentralized, open-source platform for prediction markets. `http://bravenewcoin.com/assets/Whitepapers/Augur-A-Decentralized-Open-Source-Platform-for-Prediction-Markets.pdf/`, 2015.

[12] Mark Allen Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[13] T. C. Schelling. *The strategy of conflict*. Oxford University Press, 1960.

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

[15] T.N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4, 1987.