Price oracle specifications

Denote:

- f - total arbitration fees (for all jurors combined) for the first round of a single dispute

Parameters to be set at deployment (that are not necessarily present in other applications of Kleros):

- D - submitter deposit; must have $D > f$.
- α - tuning parameter that controls how much precise, correct responses should be rewarded compared to less precise, yet correct responses. Require $\alpha > 1$.

Algorithm 1. *Input: Each respondent $\mathcal{USR}_i$ submits two distinct values - a lower bound $l_i \in \mathbb{R}$ and an upper bound $u_i \in \mathbb{R}$, $l_i < u_i$, giving an interval (l_i, u_i) in which this respondent believes the true value of the question is located.*

- *Sort the lower bound responses into a list $\mathcal{L}$ and the upper bound responses into a list $\mathcal{U}$, where in each case identical values are considered single elements.*
- *Compute the lists*

  $$\mathcal{L}_0 = \{l_i \in \mathcal{L} : \exists u_j \in \mathcal{U} \textit{ with } u_j \leq l_i, \;\; \nexists\, l_k \in [u_j, l_i) \cap \mathcal{L}\}$$

  and

  $$\mathcal{U}_0 = \{u_i \in \mathcal{U} : \exists l_j \in \mathcal{L} \textit{ with } l_j \geq u_i, \;\; \nexists\, u_k \in (u_i, l_j] \cap \mathcal{U}\}\,.$$

- *Compute*

  $\mathcal{C}_0 = \{\textit{median}(l_i, u_j) : l_i \in \mathcal{L}_0,\, u_j \in \mathcal{U}_0 \textit{ with } u_j \leq l_i, \;\; \nexists\, l_k \in [u_j, l_i) \cap \mathcal{L}, \;\; \nexists\, u_k \in (u_j, l_i] \cap \mathcal{U}\}$

  (So, if we considered $\mathcal{L}$ and$\mathcal{U}$ in the same line, essentially $\mathcal{L}_0$ would consist of lower bounds which have an upper bound to their immediate left and $\mathcal{U}_0$ would consist of upper bounds that have a lower bound to their immediate right. Then $\mathcal{C}$ consists of the midpoints between each of these pairs.)

- *If $\mathcal{C}_0 \neq \emptyset$*

- *For each $z \in \mathcal{C}_0$ perform the following in parallel:*
  - *Ask Kleros jurors if*

    $$\textit{desired value} \leq z$$

    or

    $$\textit{desired value} > z.$$

  - *Allow appeals of their decision as necessary following the standard pattern of number of jurors doubling, crowdsourceable fees, etc. Again, here the two sides of the dispute are as follows:*

    $$\textit{desired value} \leq z$$

    or

    $$\textit{desired value} > z$$

- *Take $\mathcal{C}_1 = \mathcal{C}_0$.*
- *While $\mathcal{C}_1 \neq \emptyset$*
  - *If $\#\mathcal{C}_1$ is odd, calculate $m = median(\mathcal{C}_1) \in \mathcal{C}_1$. If $\#\mathcal{C}_1$ is even and positive, choose one of the two middle-most values of $\mathcal{C}_1$ as m in some predictable way (such as by always taking the value on the left).*
  - *Eliminate all l_i and u_i that are on the wrong side of what the jurors decided with respect to m from $\mathcal{L}$ and $\mathcal{U}$.*
  - *Add m to $\mathcal{L}$ if the jurors have ruled that the true value is higher than m, and add m to $\mathcal{U}$ otherwise.*
  - *(Re)calculate*

    $$\mathcal{C}_1 = \{median(l_i, u_j) : l_i \in \mathcal{L}_0,\ u_j \in \mathcal{U}_0 \text{ with } u_j \leq l_i,\ \ \nexists\, l_k \in [u_j, l_i) \cap \mathcal{L},\ \ \nexists\, u_k \in (u_j, l_i] \cap \mathcal{U}\}$$

    based on the updated/recalculated $\mathcal{L}_0$ and $\mathcal{U}_0$.

Output the average of the largest remaining element of $\mathcal{L}$ and the smallest remaining element of $\mathcal{U}$.

After the output is decided, determine payouts to submitters as follows:

For each user $\mathcal{USR}_i$ who submits an interval $I_i = (l_i, u_i)$ take $length(I_i) = u_i - l_i$. If the ultimate response from the oracle is not in I_i, the user loses his deposit D. If the response is in I_i, the user gets his deposit back plus an additional

$$\frac{\#\ \textit{incorrect responses} \cdot D - \textit{total cost of first round arbitration}}{\sum_{j \textit{ such that } \mathcal{USR}_j \textit{ correct}} \alpha^{-length(I_j)}} \cdot \alpha^{-length(I_i)}.$$

Remark 1. *At the expense of additional gas, after each appeal round in algorithm 1, one can test whether the required disputes for the while loop to terminate have been finalized, i.e. have not been appealed. The appealed disputes should still be arbitrated to allow for the correct redistribution of PNK to participating jurors, but they need not unnecessarily delay the finalization of the result of the oracle.*