Byzantine Preferential Voting

Darya Melnyk, Yuyi Wang, and Roger Wattenhofer

ETH Zurich, Zurich, Switzerland
{dmelnyk,yuwang,wattenhofer}@ethz.ch

Abstract. In the Byzantine agreement problem, n nodes with possibly different input values aim to reach agreement on a common value in the presence of $t < n/3$ Byzantine nodes which represent arbitrary failures in the system. This paper introduces a generalization of Byzantine agreement, where the input values of the nodes are preference rankings of three or more candidates. We show that consensus on preferences, which is an important question in social choice theory, complements already known results from Byzantine agreement. In addition, preferential voting raises new questions about how to approximate consensus vectors. We propose a deterministic algorithm to solve Byzantine agreement on rankings under a generalized validity condition, which we call Pareto-Validity. These results are then extended by considering a special voting rule which chooses the Kemeny median as the consensus vector. For this rule, we derive a lower bound on the approximation ratio of the Kemeny median that can be guaranteed by any deterministic algorithm. We then provide an algorithm matching this lower bound. To our knowledge, this is the first non-trivial generalization of multi-valued Byzantine agreement to multiple dimensions which can tolerate a constant fraction of Byzantine nodes.

Keywords: social choice · Byzantine agreement · Pareto-Validity · distributed voting · multivalued

1 Introduction

In distributed machine learning, different data is often collected and owned by different parties, each of which will locally train its own machine learning model. If a new data item needs to be judged, the parties could collaborate in order to make a collective decision. As an example, a hospital may be authorized to use its own collected patient data to train an image recognition model, but not to share that data with other hospitals because of patient privacy limitations. For some critical cases the hospitals would still want to collaborate and decide on the correct diagnosis together.

In order to obtain a *robust* collective decision, we need to take the following two aspects into account. On the one hand, it is possible that some of the involved parties experience hardware or software difficulties, or simply play dirty. Our decision will be robust if we can withstand even *Byzantine* parties, who are controlled by a single omnipotent adversary trying to maliciously disturb

the process. On the other hand, non-Byzantine parties should use all available information to come up with the best possible decision. In standard multi-valued Byzantine agreement algorithms, each party will provide only one input, however, machine learning algorithms usually provide information about the second-best and third-best guess. For example, when doing image recognition in medicine the result can be a *ranking* of possible diagnoses: glioblastoma $\succ$ metastasis $\succ \ldots \succ$ inflammatory. Such rankings convey much more information than just the top ranked alternative (glioblastoma). While the different honest parties might completely disagree on the top alternative, the second alternative might serve as a tie breaker, and we can therefore hope to receive more meaningful results from the voting process by considering rankings.

In this paper we use social choice theory in order to investigate the most fair choice among a set of rankings to solve Byzantine agreement on rankings. In particular, we want to study how robust preferential voting is in a Byzantine environment. In Section 2, we first focus on some basic properties for voting rules, and see that not all of them can be satisfied if the parties should reach an agreement. This is because Byzantine voters are manipulators that modify the result to make it more favorable to themselves. In the main part of the paper (Section 5) we then study how well the voting result intended by the correct (non-Byzantine) voters can be approximated. For this purpose we consider the Kemeny rule which picks the most central ranking as the voting result. We will provide an algorithm that approximates the solution of the Kemeny rule in the presence of Byzantine voters and prove that this algorithm computes the best possible approximation. We believe our paper will contribute a deeper understanding of both fault-tolerant distributed systems as well as social choice theory.

2 Background and Motivation

In search of a *fair* rule to elect candidates, philosophers and mathematicians started developing various voting mechanisms and rules already in the beginning of the 18th century. In the middle of the 20th century, Kenneth Arrow [2, 3] was one of the first to formalize existing voting rules and analyze possibility and impossibility results in an axiomatic fashion, thereby introducing the field of Computational Social Choice. In this section we will show how well Byzantine agreement connects to voting theory.

We start by considering the special case of n voters voting on only two candidates c_1 and c_2. In this setting, each voter (node) ranks the two candidates such that its preferred candidate (input value) is ranked first. A vote for a candidate c_1 means that the voter strictly prefers c_1 to c_2, which we here denote $c_1 \succ c_2$. A central authority then applies a *social choice function (SCF)* to a given preference profile in order to determine the winner (decision value), or set of winners in case of a tie. An SCF should typically strive to satisfy anonymity, neutrality and positive responsiveness. May's theorem [27] shows

that the majority rule is the only voting rule on two candidates that satisfies all three properties.

Interestingly, most known algorithms for binary Byzantine agreement indirectly exploit the properties of May's theorem: Some of them make use of leaders who suggest their decision value to all nodes, e.g., the King and the Queen algorithms [8, 9]. The leader in these algorithms temporarily plays what is known as a dictator in voting theory. Another type of algorithm, e.g., the shared coin algorithm in [39], is biased towards one of the outcomes and thus violates neutrality. In general we can say that most of the proposed algorithms try to use the majority value as the decision value if a majority exists, or an arbitrary input value otherwise, see for example [7, 11]. Such settings may satisfy anonymity and neutrality, but in cases where the correct nodes are undecided, i.e., there is a tie between the two input values, Byzantine nodes have a large influence on the majority value. Thus, if a correct node decides to swap two candidates in its ranking in order to make one of the candidates win, a Byzantine node can perform an opposite swap in its own ranking and return the profile to the previous state. This shows that positive responsiveness cannot be satisfied for these algorithms in the presence of Byzantine nodes.

May's theorem does however not apply to the general case with more than two candidates. Moreover, a lot of information is lost when a single winner is sought. When it comes to preferential voting, social choice theory often wants not only the input to be rankings, but also the output. This is satisfied by *social welfare functions (SWF)* that map a preference profile to a set of consensus rankings. For an SWF, g, the following three properties are usually considered:

- g is *dictatorial* if there is one distinguished voter whose input ranking is chosen as the single consensus ranking
- g is *independent of irrelevant alternatives (IIA)* if the consensus ranking of two candidates c_i and c_j only depends on the relative preference of these candidates in each voter's ranking, and not on the ranking of some third candidate c_k
- g is *weakly Paretian* if it satisfies the weak Pareto condition [31]: for two candidates c_i and c_j which are ranked $c_i \succ c_j$ by all voters, consensus ranking has to rank $c_i \succ c_j$ as well

Unfortunately, Arrow's impossibility theorem[2] shows that every SWF on three or more alternatives that is weakly Paretian and IIA must be dictatorial. From the viewpoint of Byzantine agreement, an SWF should not be dictatorial since one does not want a dictator to be a Byzantine node. Consequently, any reasonable Byzantine agreement protocol must either violate IIA or weak Pareto. We say that IIA or weak Pareto are satisfied in the Byzantine setting if they are satisfied with respect to the input rankings of the correct nodes only. Under this assumption, the IIA condition implies that the consensus ranking should remain the same if the input of every correct node does not change, no matter what the Byzantine nodes do. However, a Byzantine node can pretend to be a correct node but change its ranking in two executions in which the correct nodes

have the same inputs. This change may lead to a different consensus ranking and thus violate IIA. For the weak Pareto condition consider the case with two candidates: if every non-Byzantine voter ranks $c_1 \succ c_2$, the consensus ranking should also rank $c_1 \succ c_2$. This corresponds to a well-known validity condition in Byzantine agreement – the *All-Same-Validity*: If all correct nodes have the same input value, all correct nodes have to decide on this value. We use the weak Pareto condition to impose a validity rule on Byzantine agreement on rankings:

Pareto-Validity for any pair of candidates c_i and c_j: if all correct nodes rank $c_i \succ c_j$, then the consensus ranking should rank $c_i \succ c_j$ as well.

Given m candidates, Pareto-Validity can be viewed as All-Same-Validity applied on each of the $\binom{m}{2}$ pairs of candidates in a ranking. Note that Byzantine agreement on a ranking is at least as hard as binary Byzantine agreement: Consider a case where the nodes agree on the ranking of the candidates $c_3, \dots c_m$ which they rank last, but not on the two first candidates c_1 and c_2. Pareto-Validity is then satisfied for every binary relation which contains at least one of the candidates $c_3, \dots c_m$. Agreement in this case is then reduced to binary Byzantine agreement on the two candidates c_1 and c_2, under the All-Same-Validity condition.

There is no straightforward way to apply a binary Byzantine agreement protocol to solve Byzantine agreement on rankings. This is because, in contrast to binary relations on two candidates, preference profiles can contain Condorcet cycles, e.g. tree contradicting binary relations $c_i \succ c_j$, $c_j \succ c_k$ and $c_k \succ c_i$ which are each preferred by a majority of nodes. Simply agreeing on each pair of candidates can thus lead to a circular decision which does not form a ranking. In order to get rid of such cycles one could think of applying the quicksort algorithm on the candidates sorted with respect to the majority. This procedure will however violate Pareto-Validity: Consider a candidate c_i that Pareto dominates candidate c_j. Assume that the quicksort algorithm compares both candidates to some third candidate c_k first. Then c_j might win against c_k and c_i might lose, thus swapping c_i and c_j in the consensus ranking. This consideration makes the problem of finding a consensus ranking in the presence of Byzantine nodes rather an instance of multi-valued agreement, as we discuss in Section 4, which makes the problem both interesting and challenging.

3 Related Work

Byzantine agreement was first proposed as the Byzantine Generals problem by Pease, Shostak and Lamport [32, 26]. In these papers the authors showed that three nodes cannot establish agreement in the presence of one Byzantine node even if the communication system is synchronous. Given n nodes, it was shown for the synchronous model that at least $t + 1$ rounds are required to establish agreement [20], where $t < n/3$ is the number of Byzantine nodes in the system; the corresponding upper bound was provided in [8, 9]. For the asynchronous model, the FLP impossibility result [21] states that there is no deterministic

agreement protocol which can tolerate even one Byzantine node. The first randomized algorithm for solving Byzantine agreement proposed in [7] had expected exponential running time for a constant fraction of Byzantine nodes. Recently, the authors of [25] claimed that it is possible to establish agreement within expected polynomial running time using spectral methods.

Byzantine agreement with more than two input values has mostly been considered in approximate agreement [17, 19], where the input values of the nodes converge towards some value over rounds. More recent results seek to establish agreement on a value that makes sense for applications. In [16], the values converge towards a value at most $\sqrt{n \log n}$ positions away from the median. In [35, 28] an exact algorithm to establish agreement on a value that is at most $t/2$ positions away from the median or t positions away from a minimum or a maximum was proposed. In [38, 29, 30], Byzantine agreement was further generalized to several dimensions. There, the nodes converge to a vector inside the convex hull of all correct input vectors. In [37, 13] the authors consider voting in Byzantine systems, they do however only focus on single winners that are determined by applying the plurality rule to the top alternatives of the rankings, a setting which corresponds to standard Byzantine agreement. All previous approaches for multiple dimensions struggle to derive an algorithm which either can tolerate a constant fraction of Byzantine nodes independent on the number of dimensions, or find a solution that is not trivial.

In social choice theory, Byzantine behavior can be interpreted as manipulation of a ballot in an election, in which the manipulating party has full knowledge about all votes. Bartholdi et al. [5] defined manipulation as a preference profile where one single voter can change its ranking such that this voter's most preferred candidate wins the election. Groups of voters have also been considered in this context, but mostly from the perspective of how hard it is for a group of nodes to manipulate the voting result given a certain voting rule [10, 14]. Other types of Byzantine behavior have been considered with respect to robustness of proposed voting rules. In [6], the authors investigate robustness of Borda's mean and median in the presence of outlier ballots. In [33], robustness of scoring rules is considered under arbitrary noise which is described in terms of pairwise swaps of candidates in the ranking of one voter.

In this paper we will consider the Kemeny rule which was first proposed in [22, 23]. The corresponding Kemeny median satisfies additional properties to those presented in Section 2, but it was shown to be NP-hard to compute for an increasing number of candidates and already for four voters in [4, 18]. At least three different 2-approximation algorithms for the Kemeny median have been proposed in [1] and [15]. In [1], the approximation ratio was improved to 4/3 using randomization, and later derandomized in [40]. A good overview over the Kemeny rule and an extended introduction into social choice theory can be found in [12].

4 A Deterministic Algorithm for Pareto-Validity

This section focuses on Byzantine agreement protocols for rankings that satisfy Pareto-Validity. By using a similar idea to single transferable voting[36] and a multi-valued Byzantine agreement algorithm, a ranking satisfying Pareto-Validity can be obtained in $(m-1)\cdot(t+1)$ rounds: In the first $t+1$ rounds, we let the voters apply the King algorithm [9] in order to agree on the top candidate. After this, every node removes this candidate from its ranking. In the next step, they will agree on the top candidate from the reduced rankings, and so on. While this procedure is simple, the number of rounds depends not only on the number of nodes, but also on the number of candidates.

In the following we present a deterministic algorithm which solves this problem in only $t+1$ phases using the same number of messages. We do this by modifying the King algorithm to broadcast rankings instead of single candidates. For convenience, we assume that a broadcast operation also includes sending a message to oneself. In the proposed algorithm, we select $t+1$ different nodes and assign each of them to one of the $t+1$ phases of the algorithm. Such a node is called the dictator of the corresponding phase. This dictator then suggests its own, possibly adjusted, ranking to all nodes, which will always be accepted if the dictator is a correct node. This way, dictators decide on the ranking of all pairs of candidates which do not satisfy the Pareto-Validity. Algorithm 1 presents this procedure in pseudocode.

Since we are dealing with rankings, it is not trivial to see that the nodes always will be able to agree on a proper ranking at the end of the algorithm. The following lemmas state that the nodes can adjust their rankings in Step 9 of Algorithm 1 in order to guarantee Pareto-Validity and that the outcome of the algorithm thus will be a proper ranking. It is easy to see that the algorithm is correct for $t < n/4$ Byzantine nodes, since the correct nodes will not be able to propose binary relations which form a Condorcet cycle in this case. In order to show that the algorithm can tolerate $t < n/3$ Byzantine nodes a well, we need to exploit the fact that no Byzantine node can propose relations that form a Condorcet cycle at any point of the algorithm.

Lemma 1. *There is no Condorcet cycle that can be proposed by the correct nodes if $t < n/3$.*

Note that by the properties of the King algorithm, no two opposite binary relations can be proposed in Step 4 simultaneously. Lemma 1 additionally shows that a Condorcet cycle cannot be proposed in Step 4 and that all proposed pairs can form a ranking. It remains to be proven that the nodes will always be able to adjust their rankings to incorporate the proposed pairs.

Lemma 2. *In Step 9 a correct node will always be able to incorporate the proposed pairs into its own ranking.*

Proof. This is constructed based on the following strategy: Divide the candidates into two sets. The first set contains all candidates which appear in at least one

Algorithm 1 Byzantine agreement protocol on rankings (for $t < n/3$)

Every node v executes the following algorithm
1: **for** phase 1 to $t + 1$ **do**
Communication Round:
2: Broadcast own input ranking r_v
3: **for** all pairs of candidates c_i and c_j **do**
4: **if** c_i is ranked above c_j in at least $n - t$ rankings **then**
5: Broadcast "propose $c_i \succ c_j$"
6: **end if**
7: **end for**
8: **if** some "propose $c_k \succ c_l$" received at least $t + 1$ times **then**
9: Adjust own ranking r_v according to Lemma 2
10: **end if**
11: **if** some "propose $c_k \succ c_l$" received at least $n - t$ times **then**
12: Fix the pair $c_k \succ c_l$
13: **end if**
Dictator Round:
14: Let node w be the predefined dictator of the current phase
15: The dictator broadcasts its ranking $r_{dictator} \coloneqq r_w$
Decision Round:
16: **if** $r_{dictator}$ agrees with r_v in all fixed pairs $c_i \succ c_j$ from Step 12 **then**
17: $r_v \coloneqq r_{dictator}$
18: **end if**
19: **end for**
20: Return r_v

of the pairs proposed by the $t + 1$ nodes in Step 9. This set of nodes will be ranked first. The second set will contain all candidates for which the node has not received any propose message. These candidates will be ranked second and will be dominated by all candidates from the first set. Next, we can rank all candidates in the first set according to the proposed relations, possibly leaving some pairs of the candidates not ranked. In the last step, all candidates which have not been ranked in each of the sets can be ranked by choosing binary relations from the local ranking of the node. This strategy outputs a ranking of candidates in which all proposed binary relations are satisfied. □

The next lemma summarizes the correctness results of Algorithm 1 and states that the consensus ranking will be valid.

Lemma 3. *At the end of Algorithm 1 all nodes will have agreed on the same ranking which additionally satisfies Pareto-Validity.*

5 Kemeny Median with Byzantine Nodes

Weakly Paretian voting rules are often not sufficient to pick a fair ranking from a set of individual preference rankings. In search of the best possible consensus

ranking we have to add restrictions on the voting rules without violating the known impossibility results of Arrow [2]. This leads us to majoritarian SWFs, one of which is the Kemeny rule. In the following we will introduce this rule and use it to derive a better consensus ranking in the presence of Byzantine nodes. Since Byzantine nodes have influence on the final ranking, the corresponding solutions can be qualified with respect to their approximation ratio which we define in Section 5.1. In Section 5.2, we will derive lower bounds on the approximation ratio of the Kemeny median in the presence of Byzantine nodes and further provide a matching upper bound in Section 5.3.

Definition 1 (Kendall's τ distance [24]). *The Kendall's τ distance measures the distance between two rankings r and p on candidates $c_1, \ldots, c_m$ by counting all pairs of candidates on which they disagree:*

$$\tau(r, p) \triangleq |\{(c_i, c_j) \mid c_i \succ_r c_j \text{ and } c_j \succ_p c_i\}|.$$

This metric τ on ballots can be extended to a distance function between a ranking r and a profile $\mathcal{P}$:

$$\tau(r, \mathcal{P}) \triangleq \sum_{p \in \mathcal{P}} \tau(r, p).$$

Definition 2 (Kemeny median). *For a given profile $\mathcal{P}$, the Kemeny median is the ranking r which minimizes $\tau(r, \mathcal{P})$.*

The Kemeny median satisfies many nice properties and to some extent guarantees that the chosen ranking is "fair". The most prominent quality is probably *monotonicity*: if voters increase a candidate's preference level, the ranking result either does not change or the promoted choice increases in overall popularity. This quality makes the median solution more robust to Byzantine behavior. The Kemeny rule is also a Condorcet method, it only depends on the number of voters who prefer one alternative over the other and is reinforcing.

Kendall's τ distance, which is used in the Kemeny rule, essentially captures the nature of multidimensionality in our consensus problem. Although it is not straightforward to properly define dimensions for metric spaces, there exist some widely used definitions such as the equilateral dimension. The equilateral dimension is described by the maximum number of points which lie at equal distance from each other. Using the equilateral dimension makes a lot of sense in many cases, it is for example not difficult to see that the equilateral dimension of a d-dimensional Euclidean space is $d + 1$. Here we also use the equilateral dimension in order to argue that by using the Kemeny rule we are actually solving a multi-dimensional consensus problem. For any m, we can construct rankings $r_i, i = 1, \ldots, \lfloor m/2 \rfloor$ at equal distance as follows: r_i ranks every candidate j as the j-th element in the ranking and only swaps the candidates $2i - 1$ and $2i$. Any pair of rankings in this construction has the same distance 2 to each other and the equilateral dimension of Kendall's τ metric space is therefore at least $\lfloor m/2 \rfloor$.

5.1 Byzantine Setting

The Kemeny median cannot be computed exactly in the presence of Byzantine nodes since they might suggest rankings which have a large distance to the Kemeny median of the correct nodes, thus moving the median ranking away from the actual median. A notion for approximate median rankings is therefore introduced as follows:

Definition 3 (α-approximation of Kemeny median). *Let κ be a Kemeny median of a preference profile $\mathcal{P}$. An α-approximation of κ is a preference ranking κ_α satisfying*

$$\tau(\kappa_\alpha, \mathcal{P}) \leq \alpha \cdot \tau(\kappa, \mathcal{P})$$

As an example consider binary agreement ($m = 2$): Here τ counts the number of correct nodes who disagree with the consensus value. Any binary Byzantine agreement algorithm that satisfies All-Same-Validity will also satisfy $\alpha < n - t - 1$.

Unlike binary agreement, it is not straightforward to see what a Byzantine node would choose as its ranking when the Kemeny rule determines the consensus ranking. Since the input vectors of nodes are rankings, each voter has to propose a strict order between candidates and the corresponding preference relation is transitive. A possible strategy for the Byzantine nodes could then be to choose exactly the opposite ranking of the Kemeny median of all correct nodes. While this strategy can be shown to be optimal, such a solution is not unique for most preference profiles. To see this, assume that all correct nodes agree on the preference $c_i \succ c_j$ such that this pair will always belong to the Kemeny median of the correct rankings. Then, the Byzantine nodes can pick either $c_i \succ c_j$ or $c_j \succ c_i$ for their ranking, since this strategy does not have any influence on the Kemeny median of all rankings. It is therefore difficult for the correct nodes to detect which of the rankings might have been Byzantine.

5.2 Lower Bounds on the Approximation Ratio

In this section we discuss preference profiles that are vulnerable to Byzantine nodes. The first case is based on reducing the rankings to binary agreement and gives the highest approximation ratio for $t < n/3$. Binary agreement does however assume that there are two groups of voters who completely disagree in their preferences. This is somewhat unlikely in practical situations when m is sufficiently large. In the second case we therefore exclude such binary instances and provide a lower bound based on Condorcet cycles within a preference profile which converges to the same value for large m. The approximation ratio usually depends on the ratio n/t, which will be denoted k for the sake of simplicity.

For our analysis, we represent the preference profile $\mathcal{P}$ as a weighted *tournament graph*, i.e., a graph where the nodes represent the candidates and weighted edges represent how many voters prefer one candidate to the other. The sum of the forward and the backward edges should be equal to the total number of voters in the corresponding preference profile. The ranking of a node is a directed

Hamiltonian path following the order of the ranking, and all other edges are derived from the transitivity. For any two candidates we denote the edge between these candidates a *majority edge* if its backward edge has a smaller weight. The backward edge we then call a *minority edge*. A Kemeny median of a weighted tournament graph is the ranking that minimizes the sum of the weights of all backward edges of the graph. Note that rankings restrict the power of Byzantine nodes in the sense that Byzantine nodes only can send transitive tournament graphs where every edge has weight 1.

We first consider all possible preference profiles, in which the worst case is the binary case. This case corresponds to a class of tournament graphs where the Byzantine nodes can redirect all edges by adding t rankings to the preference profiles of the correct nodes. Theorem 1 gives a lower bound for the binary case.

Theorem 1. *There exists a tournament graph corresponding to a preference profile for which the Byzantine nodes may change the edge weights such that no deterministic algorithm can output a ranking which is better than a* $\frac{k}{k-2}$*-approximation of the Kemeny median of all correct nodes, where* $k = n/t$*. For* t *close to* $n/3$*, this gives a* 3*-approximation.*

Proof. This tournament graph is equivalent to binary agreement. Consider therefore one pair of candidates: t Byzantine nodes are only able to change the median, i.e., the majority edge, between these two candidates if they can swap the majority and minority edge by supporting the minority edge with their ranking. Assume the worst case, where the forward and the backward edge both have the same weight $n/2$ after the Byzantine nodes have added their preferences. In this worst case the tournament graph of correct nodes had the weight $n/2$ for the majority edge. Since the correct nodes will not be able to determine the actual majority edge, they might agree on a minority edge with weight $n/2 - t$ instead. The corresponding approximation ratio is then $\frac{n/2}{n/2-t} = \frac{k}{k-2}$. This result can be easily generalized to m candidates by using opposite rankings.

In the following, we present another lower bound using Condorcet cycles which can result in ambiguous views as well. We start with one directed cycle formed by three nodes on the tournament graph and assume that every majority edge has a weight of more than $(n+t)/2$, thus discarding the possibility to reduce any pair of forward and backward edges in the tournament graph to binary agreement. The main difficulty in finding a good example comes from the fact that not every tournament graph has an underlying preference profile.

Theorem 2. *There exists a preference profile containing directed majority cycles in the corresponding tournament graph, for which the Byzantine nodes can add* t *rankings such that no deterministic algorithm can output a ranking with a better approximation ratio to the actual median than* $k/(k-2)$*, for* m *large.*

Proof. Considering a tournament graph formed by one directed cycle of candidates $c_1,\ c_2,\ c_3$, i.e., a directed cycle formed by majority edges. Assume all correct nodes receive a view where $n - 2t - 2$ nodes prefer c_1 to c_2, where (c_1, c_2)

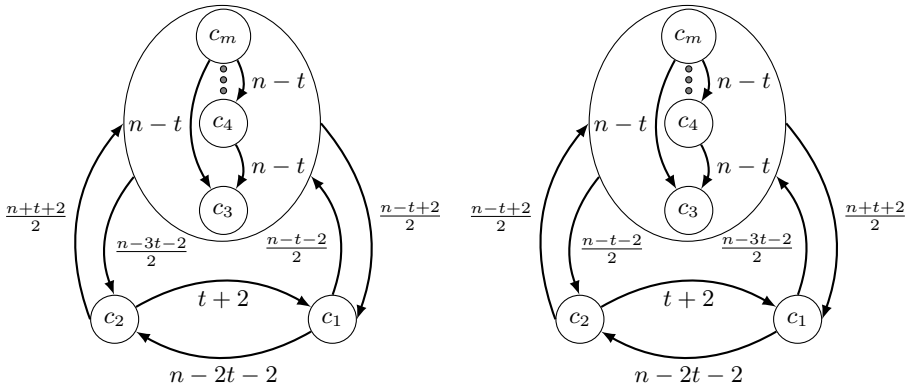

Fig. 1. Two indistinguishable views on m candidates for directed cycles

We have two views which show the profiles of correct nodes only. The left tournament graph results from a profile where $\frac{n-t}{2}-1$ nodes choose $c_1 \succ c_2 \succ c_m \succ \ldots \succ c_3$, $\frac{n-3t}{2}-1$ nodes choose $c_m \succ \ldots \succ c_3 \succ c_1 \succ c_2$ and $t+2$ nodes choose $c_2 \succ c_m \succ \ldots \succ c_3 \succ c_1$. The right tournament graph results from $\frac{n-3t}{2}-1$ nodes choosing $c_1 \succ c_2 \succ c_m \succ \ldots \succ c_3$, $\frac{n-t}{2}-1$ nodes choosing $c_m \succ \ldots \succ c_3 \succ c_1 \succ c_2$ and $t+2$ nodes choosing $c_2 \succ c_m \succ \ldots \succ c_3 \succ c_1$. If the Byzantine nodes add t profiles $c_m \succ \ldots \succ c_3 \succ c_2 \succ c_1$ to the left view, and t profiles $c_2 \succ c_1 \succ c_m \succ \ldots \succ c_3$ to the right view, the resulting profiles become indistinguishable to the correct nodes.

is a majority edge. Then $(n+t)/2+1$ nodes prefer c_2 to c_3 and $(n+t)/2+1$ nodes prefer c_3 to c_1. For $n > 3t+4$, the edge (c_1, c_2) is in the median ranking of all nodes. Since the edges (c_2, c_3) and (c_3, c_1) cannot both be in the median ranking, the nodes have to decide for one of the rankings. In the worst case, one of these two edges was supported by all t Byzantine nodes while the other edge was not supported by any Byzantine node. This leads to two views which are not distinguishable for the correct nodes, as shown in Figure 1. The approximation ratio for these views is

$$\frac{n+t+2}{n-t+2} \approx \frac{k+1}{k-1} < \frac{5}{3}$$

An extension to m candidates gives an approximation ratio of

$$\frac{m \cdot n + 2n + t + 2}{m \cdot (n-2t) + 2n - 3t + 2} \approx \frac{k}{k-2}$$

for large m. □

The received approximation ratio converges to the same approximation ratio as in the binary case for large m, even though we have excluded the binary case from the tournaments. This lower bound underlines the fact that Byzantine agreement on rankings is more complex than binary Byzantine agreement.

5.3 Algorithm for Kemeny Median Approximation

In this section we present a synchronous algorithm for computing a consensus median which matches the lower bound on the approximation ratio presented in the previous section. A simple idea is to use interactive consistency [11, 34]: For $t+1$ rounds, the nodes exchange all information they have received this far and after the $(t+1)$-st round they compute the Kemeny median from a set of rankings which they have received often enough. This algorithm guarantees that the set of rankings will be the same for each node and therefore that all nodes will decide on the same ranking. The main drawback of interactive consistency is that it has a large message complexity. The message complexity of this strategy is in $\Theta(mn^t)$ which is exponential for $t \in \Theta(n)$. Also other approaches, such as agreeing on each ranking upfront require the nodes to reliably broadcast their rankings at least once, which results in a message complexity of at least $O(n^3)$ (each node has to forward every received ranking to all other nodes).

Instead of exchanging large amounts of information, we present an approach where we can directly exploit the fact that the Byzantine nodes cannot change a Kemeny median of the preference profile of the correct nodes by more than a transitive tournament graph with edge weights t. This strategy is presented in Algorithm 2.

Algorithm 2 Byzantine agreement for the Kemeny median (for $t < n/3$)

Every node v executes the following algorithm
1: broadcast own ranking r_v
2: compute the Kemeny median of the received preference profile, call it m_v
3: apply Algorithm 1 with m_v as an input value

Algorithm 2 has the same order of round and message complexity as Algorithm 1 as stated in the next theorem.

Theorem 3. *Algorithm 2 terminates within $t+3$ phases exchanging $O(tn^2 m \log m)$ messages. The computed consensus ranking satisfies the lower bounds from Section 5.2 and Pareto-Validity.*

6 Discussion and Future Work

In this paper we introduced a new Byzantine agreement problem which extends binary Byzantine agreement to rankings. We showed that rules for choosing a consensus ranking in voting theory fit well with requirements from Byzantine agreement. We further considered a special voting rule, the Kemeny median, for which we provided an optimal Byzantine agreement protocol that can tolerate up to $t < n/3$ Byzantine nodes. We do not claim to have chosen the best voting rule at this point, since such a rule simply does not exist due to impossibility results in voting theory. Instead, we think of our results as an inspiration to

consider a larger pool of voting rules, such as approval voting, the Godgson's rule, and many others.

References

1. Ailon, N., Charikar, M., Newman, A.: Aggregating Inconsistent Information: Ranking and Clustering. Journal of the ACM **55**(5), 23:1–23:27 (2008)
2. Arrow, K.J.: Social Choice and Individual Values. CT: Cowles Foundation, New Haven, 1st edn. (1951)
3. Arrow, K.J.: Social Choice and Individual Values. John Wiley, New York, 2nd edn. (1963)
4. Bartholdi, J., Tovey, C.A., Trick, M.A.: Voting Schemes for which It Can Be Difficult to Tell Who Won the Election. Social Choice and Welfare **6**(2), 157–165 (1989)
5. Bartholdi, J.J., Tovey, C.A., Trick, M.A.: The Computational Difficulty of Manipulating an Election. Social Choice and Welfare **6**(3), 227–241 (1989)
6. Bassett, G.W., Persky, J.: Robust voting. Public Choice **990**(3), 299–310 (1999)
7. Ben-Or, M.: Another Advantage of Free Choice (Extended Abstract): Completely Asynchronous Agreement Protocols. In: Proceedings of the Second Annual ACM Symposium on Principles of Distributed Computing. pp. 27–30. PODC '83 (1983)
8. Berman, P., Garay, J.A.: Asymptotically Optimal Distributed Consensus. In: Automata, Languages and Programming: 16th International Colloquium. ICALP (1989)
9. Berman, P., Garay, J.A., Perry, K.J.: Towards Optimal Distributed Consensus. In: 30th Annual Symposium on Foundations of Computer Science. FOCS (October 1989)
10. Betzler, N., Niedermeier, R., Woeginger, G.J.: Unweighted coalitional manipulation under the borda rule is np-hard. In: IJCAI. vol. 11, pp. 55–60 (2011)
11. Bracha, G.: Asynchronous Byzantine Agreement Protocols. Information and Computation **75**(2), 130–143 (1987)
12. Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A.D.: Handbook of Computational Social Choice. Cambridge University Press, New York, NY, USA, 1st edn. (2016)
13. Chauhan, H., Garg, V.K.: Democratic Elections in Faulty Distributed Systems. In: Distributed Computing and Networking. ICDCN 2013. (2013)
14. Davies, J., Katsirelos, G., Narodytska, N., Walsh, T.: Complexity of and algorithms for borda manipulation. In: AAAI. vol. 11, pp. 657–662 (2011)
15. Diaconis, P., Graham, R.L.: Spearman's Footrule as a Measure of Disarray. Journal of the Royal Statistical Society. Series B (Methodological) **39**, 262–268 (1977)
16. Doerr, B., Goldberg, L.A., Minder, L., Sauerwald, T., Scheideler, C.: Stabilizing Consensus with the Power of Two Choices. In: Proceedings of the Twenty-third Annual ACM Symposium on Parallelism in Algorithms and Architectures. SPAA (2011)
17. Dolev, D., Lynch, N.A., Pinter, S.S., Stark, E.W., Weihl, W.E.: Reaching Approximate Agreement in the Presence of Faults. Journal of the ACM **33**(3), 499–516 (1986)
18. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank Aggregation Methods for the Web. In: Proceedings of the 10th International Conference on World Wide Web. pp. 613–622. WWW '01, ACM, New York, NY, USA (2001)

19. Fekete, A.D.: Asymptotically optimal algorithms for approximate agreement. Distributed Computing **4**(1), 9–29 (1990)
20. Fischer, M.J., Lynch, N.A.: A Lower Bound for the Time to Assure Interactive Consistency. Information Processing Letters **14**(4), 183 – 186 (1982)
21. Fischer, M.J., Lynch, N.A., Paterson, M.S.: Impossibility of Distributed Consensus with One Faulty Process. Journal of the ACM **32**(2), 374–382 (1985)
22. Kemeny, J.G.: Mathematics without Numbers. Daedalus **88**(4), 577–591 (1959)
23. Kemeny, J.G., Snell, J.L.: Mathematical models in the social sciences. Introductions to higher mathematics, Blaisdell, Waltham (Mass.) (1962)
24. Kendall, M.G.: A New Measure of Rank Correlation. Biometrika **30**(1/2), 81–93 (1938)
25. King, V., Saia, J.: Byzantine Agreement in Expected Polynomial Time. Journal of the ACM **63**(2), 13:1–13:21 (2016)
26. Lamport, L., Shostak, R., Pease, M.: The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems **4**(3), 382–401 (1982)
27. May, K.O.: A Set of Independent Necessary and Sufficient Conditions for Simple Majority Decision. Econometrica **20**(4), 680–684 (1952)
28. Melnyk, D., Wattenhofer, R.: Byzantine Agreement with Interval Validity. In: 37th Annual IEEE International Symposium on Reliable Distributed Systems. SRDS (2018)
29. Mendes, H., Herlihy, M.: Multidimensional Approximate Agreement in Byzantine Asynchronous Systems. In: Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing. STOC (2013)
30. Mendes, H., Herlihy, M., Vaidya, N., Garg, V.K.: Multidimensional agreement in Byzantine systems. Distributed Computing **28**(6), 423–441 (2015)
31. Pareto, V.: Manuale di Economia Politica con una Introduzione alla Scienza Sociale. Società Editrice Libraria (1919)
32. Pease, M., Shostak, R., Lamport, L.: Reaching Agreement in the Presence of Faults. Journal of the ACM **27**(2), 228–234 (1980)
33. Procaccia, A.D., Rosenschein, J.S., Kaminka, G.A.: On the robustness of preference aggregation in noisy environments. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. p. 66. ACM (2007)
34. Srikanth, T., Toueg, S.: Simulating Authenticated Broadcasts to Derive Simple Fault-Tolerant Algorithms. Distributed Computing **2**(2), 80–94 (1987)
35. Stolz, D., Wattenhofer, R.: Byzantine Agreement with Median Validity. In: 19th International Conference on Priniciples of Distributed Systems. OPODIS (2015)
36. Tideman, N.: The Single Transferable Vote. Journal of Economic Perspectives **9**(1), 27–38 (1995)
37. Tseng, L.: Voting in the presence of byzantine faults. In: 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC) (January 2017)
38. Vaidya, N.H., Garg, V.K.: Byzantine Vector Consensus in Complete Graphs. In: Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing. PODC (2013)
39. Wattenhofer, R.: Distributed Ledger Technology: The Science of the Blockchain. CreateSpace Independent Publishing Platform, USA, 2nd edn. (2017)
40. van Zuylen, A., Williamson, D.P.: Deterministic Algorithms for Rank Aggregation and Other Ranking and Clustering Problems, pp. 260–273. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)