Secure Polynomial-Time Democratic Elections in Distributed Systems

Arnav Kashyap
Department of Electrical and
Computer Engineering
The University of Texas at Austin
Austin, Texas, United States
arnavskashyap@utexas.edu

***Abstract*—Elections with non-binary choices in distributed systems have been shown to provide sub-optimal results. In this paper, we study the preferential non-binary election problem in presence of Byzantine behavior under a probabilistic model. Applying a probabilistic model raises new questions about how to approximate consensus vectors outside of strict validity conditions. We discuss existing social welfare schemes for democratic elections in distributed systems in the presence of Byzantine behavior under this model and present two new social welfare schemes that strike a balance between ideal outcomes and computational complexity. We analyze the performance of these scheme through simulations to compare their efficacy in producing the most desirable social welfare rankings.**

I. Introduction

Many distributed systems algorithms require fault-tolerant consensus, generalized in the Byzantine Agreement problem. One prominent example is the leader election problem, which requires that all processes in the system agree on a leader who is then allowed to do some privileged tasks. Most protocols for leader election simply select the process with the lowest or highest id as the leader. For many applications, however, processes may have preferences for who should be the leader. Ideally, an optimal leader election algorithm would take these preferences into account to produce a more optimal ranking.

These preferential elections have several applications. In distributed databases, such as AWS Dynamo DB and Apache ZooKeeper, a single node may be needed to coordinate some tasks. Nodes may prefer certain leaders based on their communication latencies or loads. Using preferential elections, distributed systems can dynamically adapt to changes in the network and optimize resource utilization while maintaining consistency and coordination throughout the network. In distributed machine learning, different parties often collect and own different data. Each party may wish to locally train its own machine learning model. If a new data item needs to be judged, the parties might need to collaborate to make a collective decision. For example, a hospital might be authorized to use the patient data it collects to train an image recognition model, but not to share that data with other hospitals. Still, these hospitals may still want to collaborate to decide on a correct diagnosis together, requiring robust consensus.

Fortunately, preferential elections are well studied in a discipline called social choice theory. Arrow's theorem, an important result on this topic, shows that in all cases where preferences are ranked, it is impossible to formulate a social ordering without violating at least one of the following desirable conditions of a fair voting system: non-dictatorship, pareto efficiency, independence of irrelevant alternatives, and unrestricted domain [2]. In distributed systems, many works have applied insights from social choice theory to preferential Byzantine voting for multi-valued consensus [1]. These works apply social welfare schemes to find an optimal ranking in fault-tolerant elections with three or more candidates. Much of the existing work has been on defining and satisfying new validity conditions. Tseng [3] extends the democratic election problem to asynchronous systems, shows impossibility results, and proposes two relaxed validity conditions with voting algorithms that satisfy them. Xu et. al [4] show impossibility results and provide consensus algorithms for exact fault-tolerant consensus under a *voting* validity condition. Melnyk et. al. [5] show impossibility results and provide consensus algorithms using a modification of Kemeny-median for *Pareto* validity. Little work, however, has been done on approximation algorithms that achieve consensus that trend towards ideality in polynomial-time. Mathieu and Maurus [6] consider approximation algorithms for top-list aggregation and partial lists, but not in the presence of Byzantine processes.

The contributions of this paper are the following:

– We introduce a probabilistic model for the problem of democratic elections in distributed systems by applying the Mallows model for evaluating Social Welfare functions in distributed settings with Byzantine faults.

– We propose three new social welfare functions called Pruned-Footrule, Pruned^2-Footrule, and Trust-Vote and demonstrate their efficacy against other popular social welfare schemes.

II. Preliminaries

We follow Chauhan and Garg's definitions of *ranking*, *vote*, *ballot*, and *social welfare function* [1]. Specifically, a ranking is a total order over a fixed set of candidates, a vote is an individual voter's preference ranking over the set of candidates, a ballot is a collection of votes, and a social welfare function takes a ballot as input and produces a ranking as output.

Formally, these are defined as follows: Let $\mathcal{A}$ be a set of choices/candidates and $\{P_1,...,P_n\}$ be the set of n voters. Let $\mathcal{L}$ denote the set of linear orders on $\mathcal{A}$ ($\mathcal{L}$ is isomorphic to the set of permutations on $\mathcal{A}$). The preferences of each voter P_i are given by $\prec_i \in \mathcal{L}$, where $a \prec_i b$ means that P_i prefers choice b to choice a. A social welfare function $\mathcal{W}$ is a function of the form $\mathcal{W}$: $\mathcal{L}^n \rightarrow \mathcal{A}$.

We also introduce the Mallows model, one of the most popular means of generating and explaining ranking data. Some even refer to it as the normal distribution over permutations. The model has two components, the central order, the ideal ranking for our purposes, and the dispersion parameter $\phi \in [0, 1]$. Depending on the value of ϕ, the Mallows model generates random rankings that are more concentrated around the central one or are more evenly spread over the space of all permutations.

Formally, the Mallows model $M_{\phi,m,a_{ideal}}$ is parameterized by a central order $a_{ideal} \in \mathcal{L}$ over $k = |\mathcal{A}|$ alternatives, and a dispersion parameter $\phi \in [0, 1]$. The probability of sampling a ranking $a \in \mathcal{L}(\mathcal{A})$ under $M_{\phi,m,a_{ideal}}$ is $\frac{1}{Z(\phi,m)}\phi^{k(a_i,a)}$, where $Z(\phi, m)$ is a normalization constant known to be $(1+\phi)(1+\phi+\phi^2)...(1+\phi+...+\phi^{m-1})$. Consequently, for $\phi = 0$ only the central order a_i is sampled, whereas using $\phi = 1$ leads to a uniform distribution over rankings from $\mathcal{L}$.

III. System Under Test

We assume a synchronous distributed system consisting of n processes. Any two nodes can communicate with each other privately, and the induced communication graph is complete. Of the n nodes in the system, at most f can be Byzantine. Byzantine nodes are assumed to have complete knowledge of the execution of the algorithm, including the states of all nodes, contents of messages the other nodes send to each other, and the algorithm specification. Moreover, in the context of voting scenarios, Byzantine nodes may also collude, bribe, or lobby.

For the synchronous model of communication, [7] and [8] showed that agreement can only be guaranteed when $f < n/3$. Unless otherwise stated, we assume that this bound of $f < n/3$ holds throughout this paper. All non-faulty processes in the system are called *good processes*, and the faulty processes are called *bad processes*. The terms nodes, voters, and processes refer to the same entities and are used interchangeably. The set of candidates, $\mathcal{A}$, is known to all the processes in the system and each process votes with its strict preferences as a total order over the elements of $\mathcal{A}$.

For all schemes, after exchanging their votes with all the other processes in the system, the processes participate in $O(f)$ rounds of agreement to ensure that all the good processes agree on the same ballot.

IV. Byzantine Social Welfare (BSW)

In the Byzantine Social Welfare (BSW) problem, the goal is to produce a ranking, a total order over k candidates, as the result of election.

Chauhan and Garg assessed and evaluated the results of **PlacePlurality**, **PairwiseComparison**, **BordaCount**, **Kemeny-Young**, and **Pruned-Kemeny** [1]. When all good processes lean towards some fixed ideal ranking, Pruned-Kemeny's output ranking is significantly more accurate (measured by its distance to the ideal ranking) than all other schemes, especially when good processes were weakly inclined towards the ideal ranking. In terms of computational complexity, however, both Kemeny-Young and Pruned-Kemeny are NP-Hard.

We propose two schemes called Pruned-Footrule and Pruned^2-Footrule that balance between the benefits of Pruned-Kemeny and the computational complexity of simpler methods. First, we define Spearman-Footrule ranking aggregation.

The Spearman's footrule distance is given by:

$d_f = \sum_i |v_1(i) - v_2(i)|$ where i is a candidate and $v_1(i)$ is the rank of element i in vote v_1; this measures the total element-wise displacement between two rankings.

For example, let $c \prec_{v_1} b \prec_{v_1} a$ be v_1's ranking and $c \prec_{v_1} a \prec_{v_1} b$ be v_2's ranking. The Spearman-Footrule distance between these two rankings is 2. If v_2's ranking is $a \prec_{v_2} b \prec_{v_2} c$, then the Spearman-Footrule distance between v_1 and v_2 is 4.

In a famous result, Diaconis and Graham [9] showed that this distance differs by at most a factor of 2 from the Kendall's tau distance used in the Kemeny-Young and Pruned-Kemeny SWFs:

Theorem 1. Diaconis–Graham (DG) inequality.
$\forall v \in \mathcal{A}$, $d_k(v) \leq d_f(v) \leq 2d_k(v)$.

This inequality is in fact tight and Spearman's footrule distance is a 2-approximation of Kendall-tau distance. Using Spearman's footrule distance for ranking aggregation mirrors the Kemeny-Young scheme. The goal is to identify a ranking that minimizes the average Spearman's footrule distance across rankings in the overall ballot. That is, it returns v which minimizes:

$\frac{1}{n}\sum_{i=1}^{n}\sum_{a=0}^{m-1}|v_i(a) - v(a)|$.

This problem can be solved by solving a minimum-cost bipartite matching problem between items and their positions where the cost $C(a, r)$ between a candidate a and rank r is set to $\sum_i |v_i(a) - r|$. We use the well-known Hungarian algorithm to solve this problem, which has a runtime of $O(n^3)$. Algorithm 1 presents the steps involved for Spearman Footrule ranking aggregation. For a detailed analysis of the scheme, see [10] and [11].

Pruned-Footrule: We propose a scheme called Pruned-Footrule that aims to capture the benefits of Chauhan and Garg's Pruned-Kemeny described in [1]. Pruned-Footrule constructs a *pruned* ballot $\mathcal{B}'$ in the same manner as Pruned-Kemeny. Whereas Pruned-Kemeny uses Kendall-tau distance for finding the f-most distant rankings and ranking aggregation, however, Pruned-Footrule uses the Spearman-Footrule distance. The distance for any ranking r is the sum of its Spearman-Footrule distances from each of the rankings in the pruned ballot $\mathcal{B}'$. The result of the scheme is the ranking with

Algorithm 1 Spearman Footrule SWF

$\mathcal{B}$ = agreed upon ballot of n votes
for each candidate $a \in \mathcal{A}$ **do**
  for each rank $r \in [0, ..., k-1]$ **do**
    $C(a, r) = SpearmanCost(a, r, \mathcal{B})$
  end for
end for
Apply **min-cost-perfect-matching** to assign candidate a to rank σ_a, minimizing $\sum_{a=0}^{k-1} C(a, \sigma_a)$
return the resulting ranking σ

Algorithm 2 SpearmanCost($candidate$,$position$,$ballot$)

$cost = 0$
for each vote v in $ballot$ **do**
  $cost = cost + |v(candidate) - position|$
end for
return cost

the minimum distance. The steps of the scheme are presented in Algorithm 3.

Unlike Spearman-Footrule, however, Pruned-Footrule cannot be reduced to a single minimum cost bipartite matching problem. Spearman-Footrule's reduction finds the most optimal ranking for the single original ballot. In Pruned-Footrule, each permutation of candidates r_i can construct a different pruned ballot $\mathcal{B}'_i$ because different combinations of votes in the original ballot $\mathcal{B}$ will be farthest from r. It is possible to find the single most distant ranking from any permutation with minimum cost bipartite matching by aggregating rankings using a ballot with reversed votes. As the number of candidates grows, however, the single most distant ranking will appear in the ballot less frequently. If the scheme were to only remove occurrences of this ranking, there is no bound on how many rankings will be removed in the pruned ballot and the scheme will likely perform poorly.

For each pruned ballot, finding the optimal ranking can be solved by solving the minimum cost bipartite matching problem. This is equivalent to using Algorithm 1, replacing $\mathcal{B}$ with $\mathcal{B}' =$ agreed-upon ballot of $n - f$ votes. For each of the $k!$ permutations of k candidates, Pruned-Footrule finds the Spearman-Footrule distance of that ranking from each vote in the ballot. Thus, its complexity is $O(k!n)$. In the context of distributed systems, the round and message complexities for agreement on the the ballots, performed before application of the schemes, are essentially the complexities of the protocols used reach agreement. We use the Gradecast based Byzantine agreement protocol mainly because this protocol provides the early termination property. This agreement requires $O(f)$ rounds, and has the message complexity of $O(fn^3)$ as described in [12].

Pruned^2-Footrule: We also propose a scheme called Pruned^2-Footrule that aims to capture the Pruned-Footrule with the computational complexity of Spearman-Footrule. Two intuitions motivate this scheme. First, good voters' rankings

Algorithm 3 Pruned-Footrule SWF

$\mathcal{P}$ = set of all permutations of k candidates
$\mathcal{B}$ = agreed upon ballot of n votes
$minDistance = \infty, bestRank = nil$
for each ranking $r \in \mathcal{P}$ **do**
  $\mathcal{F} = f$ most distant rankings from r in $\mathcal{B}$
  $\mathcal{B}' = \mathcal{B} \setminus \mathcal{F}$
  $distance = SpearmanFootruleDistance(r,\mathcal{B}')$
  if $distance < minDistance$ **then**
    $minDistance = distance$
    $bestRank = r$
  end if
end for
return $bestRank$

Algorithm 4 SpearmanFootruleDistance($ranking$, $ballot$)

$distance = 0$
for each vote $v \in ballot$ **do**
  for each candidate $i \in ballot$ **do**
    $distance = distance + |ranking(i) - v(i)|$
  end for
end for
return $distance$

are inclined towards the ideal ranking. This follows from Chauhan and Garg's observation that: "good voters, while indicating their preferences, are inclined towards the overall well-being of the system" [1]. We exploit this observation to restrict the number of rankings we consider. A naive approach applies Pruned-Footrule but only considers rankings in the ballot. For greater numbers of candidates or weakly-inclined good voters, however, the ballot is unlikely to include the ideal ranking or rankings that are close to ideal.

The second motivation for Pruned^2-Footrule is that unlike Kemeny-Young, Spearman-Footrule does not require comparing each ranking (permutation of candidates). Kemeny-Young and Pruned-Kemeny finds an optimal ranking by computing the Kendall-tau distances to a given ballot for all permutations of candidates and returning the ranking with the minimum distance. For each ranking r_i and its respective pruned ballot $\mathcal{B}'_i$, Pruned-Kemeny computes the distance of that ranking from its ballot. For example, let r_i be the optimal ranking for the pruned ballot $\mathcal{B}'_j$ produced by a different ranking r_j. $r_i \neq r_j$. If a Pruned-Kemeny run does not explicitly consider r_i, it cannot return r_i as the optimal ranking, even if it considers r_j and its pruned ballot $\mathcal{B}'_j$. Unlike Kemeny-Young, Spearman-Footrule separates the rankings from their respective pruned ballots. For any given ballot, Spearman-Footrule computes its optimal ranking by finding the lowest cost (expressed by distance to the ballot) candidate-position matching in polynomial time. Returning to the earlier example, Pruned-Footrule will return r_i for the pruned ballot $\mathcal{B}'_j$ produced by a ranking r_j, even if r_j is considered and r_i is not.

We exploit these two observations to restrict the number of

rankings that we consider to be ideal. The restriction on $\mathcal{P}$ is attained in the following manner:

Let $\mathcal{B}$ be the agreed upon ballot of n votes. Compute a pruned set of possible rankings $\mathcal{P}'$ by setting $\mathcal{P}' = \mathcal{B}+$ the optimal Spearman-Footrule ranking for the pruned ballot $\mathcal{B}'$ for each vote v in $\mathcal{B}$. Pruned$_2$-Footrule constructs a 'pruned' ballot $\mathcal{B}'$ in the same manner as Pruned-Footrule. The result of the scheme is the ranking in $\mathcal{P}'$ with the minimum Spearman-Footrule distance. The steps of the scheme are presented in Algorithm 5.

Algorithm 5 Pruned^2-Footrule SWF

$\mathcal{B}$ = agreed upon ballot of n votes
$\mathcal{P}'$ = $\mathcal{B}$
$minDistance = \infty, bestRank = nil$
for each ranking $r \in \mathcal{P}'$ **do**
  $\mathcal{F} = f$ most distant rankings from v in $\mathcal{B}$
  $\mathcal{B}' = \mathcal{B} \setminus \mathcal{F}$
  if $r \in \mathcal{B}$ **then**
    for candidate $a \in \mathcal{A}$ **do**
      for each position $p \in [0, ..., k-1]$ **do**
        $C(a, p) = SpearmanCost(a, p, \mathcal{B}')$
      end for
    end for
    Apply **min-cost-perfect-matching** to assign candidate a to rank σ_a, minimizing $\sum_{a=0}^{k-1} C(a, \sigma_a)$
    $\mathcal{P}'$ = $\mathcal{P}'$ + the resulting ranking σ
  end if
  $distance = SpearmanFootruleDistance(r,\mathcal{B}')$
  if $distance < minDistance$ **then**
    $minDistance = distance$
    $bestRank = r$
  end if
end for
return $bestRank$

For both Pruned-Footrule and Pruned^2-Footrule, computing the f-most distant rankings from a ranking r is equivalent to finding the f rankings in the original ballot with the highest Spearman-Footrule distance from r. This requires iterating through every candidate in every vote. While this leads to greater computational complexity, other approaches would perform poorly. For any ballot, minimum-cost-perfect-matching can find its single most distant ranking by aggregating rankings using a ballot with reversed votes. As the number of candidates grows, however, the single most distant ranking will appear in the ballot less frequently. If the scheme were to only remove occurrences of this ranking, there is no bound on how many rankings will be removed in the pruned ballot and the scheme would likely perform poorly.

Iterating over pruned ballots, instead of all permutations of candidates, allows Pruned^2-Footrule to use the polynomial-time algorithm for Spearman-Footrule aggregation. For each pruned ballot, finding the optimal ranking can be solved by solving the minimum cost bipartite matching problem. This is equivalent to using Algorithm 1, replacing $\mathcal{B}$ with $\mathcal{B}' =$ agreed upon ballot of $n - f$ votes. Finding a solution to the election problem using this Pruned^2-Footrule scheme requires finding the f-most distances in $O(mn)$ time and running the Hungarian algorithm $2n$ times in the worst case, resulting in a $O(mn^2 + n^4)$ complexity. A simple optimization that skips a repeated ranking guarantees that Pruned^2-Footrule will never consider more rankings than Pruned-Footrule.

V. Simulation Results

We now list the details of our experimental setup to evaluate social welfare schemes in computing ideal social welfare rankings. Our setup mirrors the experimental setup of [1] with the addition of the Mallow's model. The following details remain the same:

Let ω represent an ideal ranking. Good processes' votes are inclined towards ω. Let $goodProb$ denote the probability of a good process ranking two candidates a and b in the same order as ω and $badProb$ denote the probability of a bad process ranking a and b in the reverse order of ω. We fix the following values:

$n = 100, f = 33, badProb = 0.9$

We also vary the number of candidates k in the range [3,8]. For each value of k, we vary the value of $goodProb$ from 0.55 to 0.90 in step increments of 0.05. For each resulting configuration, we generate 50 ballots with 100 votes each using the Mallow's probabilistic model on individual votes based on $goodProb$ and $badProb$. For detailed reasoning on these values, we refer the reader to [1].

We now detail our application of the Mallow's model. Let $\mathbb{E}[d_k]$ be the expected number of pairwise differences, or Kendall-tau distance, for a voter's ranking. For k candidates, the total number of pairwise comparisons or maximum number of pairwise differences is $\binom{k}{2}$. The expected Kendall-tau distance is $(1 - goodProb) \cdot \binom{k}{2}$ for good processes and $(badProb) \cdot \binom{k}{2}$ for bad processes. In the Mallow's model, $\mathbb{E}[d_k] = \frac{k\phi}{1-\phi} - \sum_{i=1}^{k} i\frac{\phi^i}{1-\phi^i} \approx \frac{\phi}{1-\phi} \cdot \binom{k}{2}$, so $\phi \approx \frac{\mathbb{E}[d_k]}{\mathbb{E}[d_k]+\binom{k}{2}}$ [13].

We apply the discussed schemes on the ballots generated by the Mallow's model and find the Kendall-tau distance of the result rankings from the ideal rankings. We then compute the average distance over all 50 ballots for each configuration.

Figure 1 in Appendix A shows the variation in the computed average distance values. These plots show that Pruned-Footrule produces results that are about as close to the ideal ranking as Pruned-Kemeny even for low values of $goodProb$. Setting aside Pruned-Kemeny and Pruned-Footrule, Pruned^2-Footrule produces results that are much closer than the other election algorithms, especially as the number of candidates increases and for low values of $goodProb$.

VI. Pruned-Footrule Validity

We now assess the validity results of Pruned-Footrule and Pruned^2-Footrule against Pruned-Kemeny. Pruned-Kemeny satisfies S: If u is the top choice of all good voters, then u must be the winner and S': If u' is the last choice of all good voters, then u' must not be the winner.

Theorem 2. *Pruned-Footrule satisfies S and S' for* $\begin{cases} g > m^2 + 2 & k = 2m \\ g > m^2 + m + 2 & k = 2m+1 \end{cases}$ *where* $g = \frac{n}{f}$

Proof: First, we prove that Pruned-Footrule satisfies S for the restricted g. Assume that g satisfies its requirement for some k and Pruned-Footrule violates S. Thus, when all good processes put u as their top choice, Pruned-Footrule's output ranking r puts u in position p_u where p_u is not the top position. Hence, there is a candidate w in the top position in r. Construct a new ranking r' by moving u into the top position and moving all candidates above u down one position. We now show that r' would have a lower Spearman-Footrule distance than r, which would be a contradiction. Since r' puts u at the top of the list, the distance of u is 0 for the rankings of each good process, while r's distance of u from the good process is p_u. Similarly, since r' moves w down one position and w is not the top choice of any of the good processes, the distance of w in r' is 1 less than the same distance in r. In the worst case, shifting each of the $p_u - 1$ candidates down one position increases the distance of that candidate from a good process's ranking by 1 in r'. In total, each good process is $(p_u + 1) - (p_u - 1) = 2$ units closer to r' than r. In the worst case, r' may discard f good votes when it constructs the pruned ballot. It may also disagree with the votes of all bad processes, contributing the max distance for each bad process. For Spearman-Footrule distance, the max distance is:

$$max(d_f) = \begin{cases} 2m^2 & k = 2m \\ 2m^2 + 2m & k = 2m+1 \end{cases} \text{ [11]}$$

Thus, in the worst case, the overall Spearman-Footrule distance decreases by $2(n-f) - 2f - max(d_f)f =$

$$\begin{cases} 2n - (4 + 2m^2)f & k = 2m \\ 2n - (4 + 2m^2 + 2m)f & k = 2m+1 \end{cases}$$

compared to r. The first term 2(n-f) is due to the reduction of 2 by each of $n-f$ processes and the second term indicates the adjustment of removing f reductions from pruning. If all f bad processes provide the exact opposite rankings, they each increment the distance by $max(d_f)$. Since $n > gf$, the score of r' is strictly greater than that of r, which means r being selected as the final outcome of Pruned-Footrule is a contradiction. S' can be shown similarly by placing v at the bottom of each good vote.

Theorem 3. *Spearman-Footrule satisfies S and S' for* $\begin{cases} g > m^2 & k = 2m \\ g > m^2 + m + 2 & k = 2m+1 \end{cases}$ *where* $g = \frac{n}{f}$

We do not provide a formal proof for this as Spearman-Footrule is a special case of Pruned-Footrule in which 0 rankings are eliminated from the ballot. The reduced distances from all $n - f$ good processes are thus included, so the second term from the previous theorem is eliminated. The proof immediately follows from Theorem 2.

We also do not prove this guarantee for Pruned^2-Footrule because it is non-deterministic based on voters' rankings. We aim to use Pruned-Footrule as a rough approximation of the general performance of Pruned^2-Footrule. Table 1 shows the values of g for which Pruned-Footrule and Spearman-Footrule guarantee S and S'.

TABLE I
VALUES OF g FOR WHICH FOOTRULE-BASED SCHEMES GUARANTEES S AND S'

k	g for Pruned-Footrule	g for Spearman-Footrule
3	4	3
4	6	5
5	8	7
6	11	10
7	14	13
8	18	17

These values for g do not provide strong guarantees in most cases, especially for Pruned-Footrule. Fortunately, the worst case scenario they assume is highly unlikely. Melnyk et. al. [5] showed that the opposite ranking of the Kemeny median always gives a worst possible solution which Byzantine processes can choose. For bad processes to not be eliminated during pruning, they must restrict how close they get to the opposite ranking, limiting their influence. This observation motivates us to evaluate Pruned-Footrule in a probabilistic model to account for which votes will be eliminated from the ballot.

We model the likelihood of that a good process or bad process is pruned from the ballot of the ideal ranking using the expected Kendall-tau distance for good and bad processes from the Mallows model. Given the tight inequality of Theroem 1, we then assume the worst-case scenario in which each incorrect pairwise comparison in bad processes contributes the minimum deviation of 1 in terms of Spearman-Footrule distance and each incorrect pairwise comparison in good processes contributes the maximum deviation of 2 in terms of Spearman-Footrule distance. The variance of the Kendall-tau distance in the Mallows model is:

$var(d_k) = \frac{k\phi}{(1-\phi)^2} - \sum_{i=1}^{k} i^2 \frac{\phi^i}{(1-\phi^i)^2} \approx \frac{k(k-1)(2\phi-1)}{(1-\phi)^2}$ $= \frac{4k\mathbb{E}[d_k](k-1)(2\mathbb{E}[d_k]+k^2-k)}{(k^2-k)^2}$ using the approximation of ϕ [13].

Figures 2 and 3 show the means $\pm$ one standard deviation for each value of goodProb for Kendall-tau and Spearman-Footrule distances respectively. As evident from the plots, in both schemes, it is far more likely that bad processes will be pruned from the ballot when considering the ideal ranking than good processes. Melnyk et. al. [5] also showed that there exists a tournament graph corresponding to a ballot for which the Byzantine processes may change the edge weights such that no deterministic algorithm can output a ranking which is better than a $\frac{g}{g-2}$-approximation of the Kemeny median of all good processes, where $g = \frac{n}{f}$. Our model and simulation results demonstrate that both methods come very close to this approximation for $g = 3$.

VII. EXTENSION: TRUST-VOTE

So far, we have only considered a fully connected distributed systems, which require significant communication

overhead. One alternative network model for distributed systems that still benefits from the centralized control offered by leader election is social overlay networks. These networks link users based on social relationships. In contrast to other peer-to-peer networks, social overlay networks restrict connectivity to nodes whose users share a mutual trust relationship in the real world. This approach improves privacy by limiting the exposure of identifying information, such as the node's IP address, to trusted participants. The SWFs described above require a fully connected system to establish consensus in the presence of malicious voters. Fortunately, social overlay networks' unique properties can be exploited for secure leader election, assuming we can relax our assumption of Byzantine voters. While an adversary may be able to insert as many nodes into a social overlay network as they wish, they must socially engineer users to create a link or communication channel with them. Thus, social overlay networks might contain an arbitrary number of bad nodes but the number of links between bad and good nodes is likely to be small in comparison to the total number of links. Good nodes will connect more to other good nodes. We propose two schemes for preferential leader election in this model. First, nodes can use local three-majority voting to achieve local consensus. Every node periodically requests the vote of three randomly chosen neighbors. If at least two respond with the same vote, then the node adopts that candidate as its leader. Otherwise, it adopts a random vote from the three responses. If all nodes start with a different vote-value, three-majority voting initially behaves similar to the voter model but as soon as a vote-value is adopted by a sufficient number of nodes, three-majority voting will converge quickly [14]. Second, each node u can maintain a trust vector for each node v that is has a link with. u can tracks the deviation of each v's preferences from the final consensus ranking. If u consistently submits rankings over the last y rounds that are far from the final agreed upon ranking, its trust score decreases. The preference of each node is weighted by its trust score. Assuming bad nodes must comply with this part of the algorithm, their influence is significantly limited by this adjustment.

VIII. Future Work

Determining the provable guarantees of Pruned-Footrule and Spearman-Footrule for other validity conditions remains an important open challenge for this work. Additionally, presenting impossibility results and showing possibilites for Spearman-Footrule-based schemes can be helpful. Another interesting problem is to explore other Polynomial-Time-Approximation-Schemes for Kemeny-median under the probabilistic model we introduce.

IX. Conclusion

In this paper, we present two new social welfare schemes for democratic elections in distributed systems using Spearman-Footrule distance. We showed possibilities that satisfy specific validity requirements and evaluated Kemeny-based and Spearman-Footrule-based social welfare functions. We introduced a novel probabilistic model based on the Mallow's model for evaluating the efficacy of social welfare functions in the presence of Byzantinen processes. The results of our simulations show that Pruned-Footrule is nearly identical in optimality to Pruned-Kemeny and Pruned^2-Footrule provides significantly better results than other polynomial-time computations by approximating Pruned-Footrule. Pruned^2-Footrule strikes a strong balance between ideality and computational complexity, especially as the number of candidates grows.

REFERENCES

[1] Chauhan, Himanshu, and Vijay K. Garg. Democratic elections in faulty distributed systems. In International Conference on Distributed Computing and Networking, pp. 176-191. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[2] Arrow, K.J.: A difficulty in the concept of social welfare. Journal of Political Economy 58, 328–346 (1950)

[3] Tseng, L. (2017, January). Voting in the presence of byzantine faults. In 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC) (pp. 1-10). IEEE.

[4] Xu, Zhangchen, Yuetai Li, Chenglin Feng, and Lei Zhang. "Exact Fault-Tolerant Consensus with Voting Validity." In 2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 842-852. IEEE, 2023.

[5] Melnyk, D., Wang, Y., Wattenhofer, R. (2018). Byzantine Preferential Voting. In: Christodoulou, G., Harks, T. (eds) Web and Internet Economics. WINE 2018. Lecture Notes in Computer Science(), vol 11316. Springer, Cham.

[6] Mathieu, Claire, and Simon Mauras. "How to aggregate Top-lists: Approximation algorithms via scores and average ranks." In Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 2810-2822. Society for Industrial and Applied Mathematics, 2020.

[7] Pease, M., Shostak, R., Lamport, L.: Reaching agreement in the presence of faults. Journal of ACM 27, 228–234 (1980)

[8] G. Neiger. "Distributed consensus revisited," Inf. Process. Lett., 49(4):195–201, Feb. 1994.

[9] Diaconis, Persi, and Ronald L. Graham. "Spearman's footrule as a measure of disarray." Journal of the Royal Statistical Society Series B: Statistical Methodology 39.2 (1977): 262-268.

[10] Dwork, Cynthia, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. "Rank aggregation methods for the web." In Proceedings of the 10th international conference on World Wide Web, pp. 613-622. 2001.

[11] Kumar, Ravi, and Sergei Vassilvitskii. "Generalized distances between rankings." In Proceedings of the 19th international conference on World wide web, pp. 571-580. 2010.

[12] Ben-Or, M., Dolev, D., Hoch, E.N.: Simple gradecast based algorithms. CoRR, vol. abs/1007.1049 v3 (2010)

[13] Boehmer, Niclas, Piotr Faliszewski, and Sonja Kraiczy. "Properties of the mallows model depending on the number of alternatives: a warning for an experimentalist." In International Conference on Machine Learning, pp. 2689-2711. PMLR, 2023.

[14] , Simple dynamics for plurality consensus, Distributed Computing (2017), 293–306.

X. Appendix A: Simulation Plots

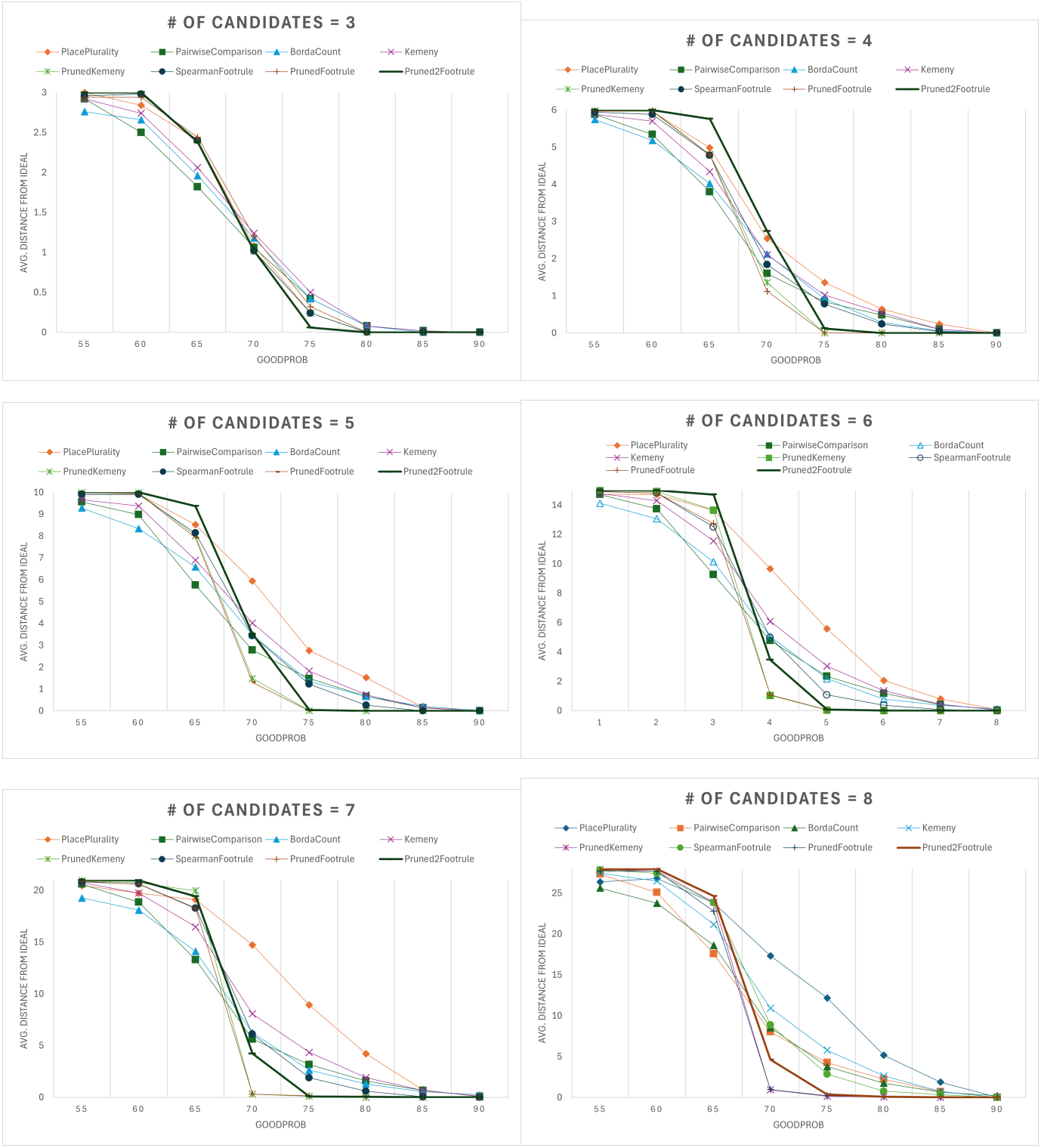

Fig. 1. Comparison of Average Distance of Results from Ideal

XI. Appendix B: Mallows Model Plots for Kemeny

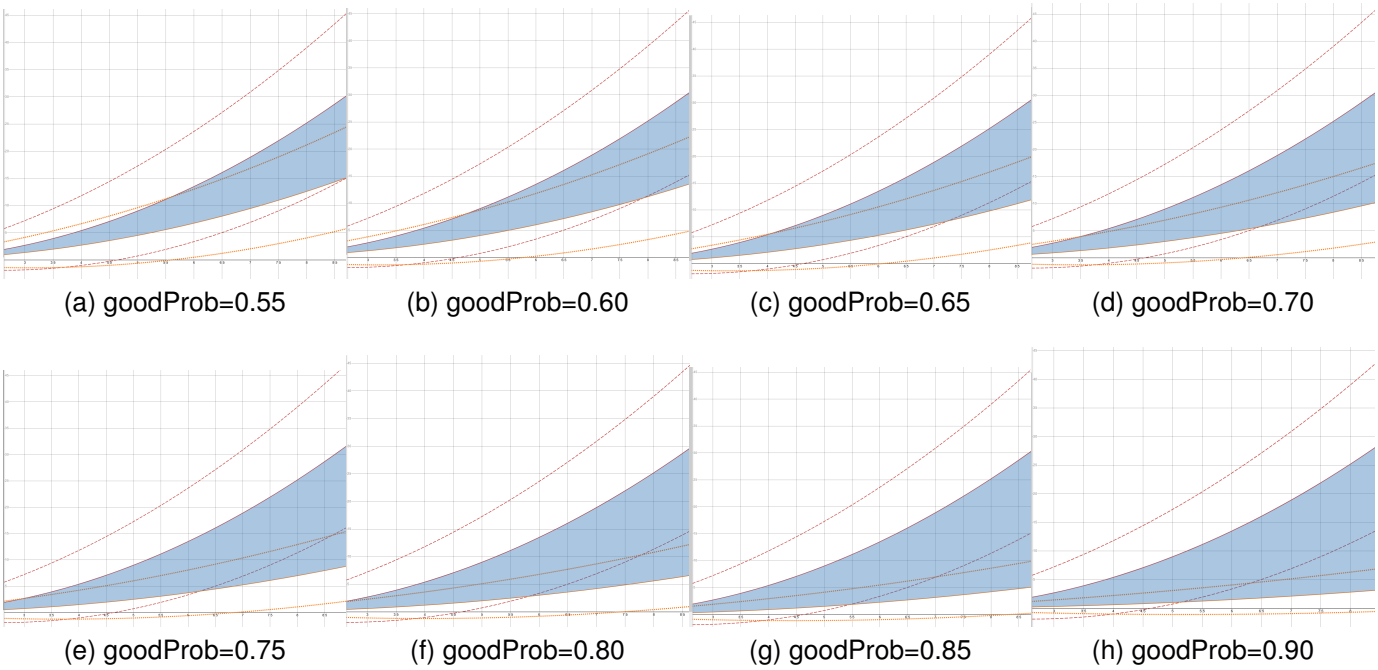

Fig. 2. Red shows mean of bad processes $\pm$ one standard deviation. Orange shows mean of mean of good processes $\pm$ one standard deviation. Blue shows difference between the mean of the good processes and bad processes. x-axis is k number of candidates and y-axis is the Kendall-tau distance

XII. Appendix C: Mallows Model Plots for Spearman-Footrule

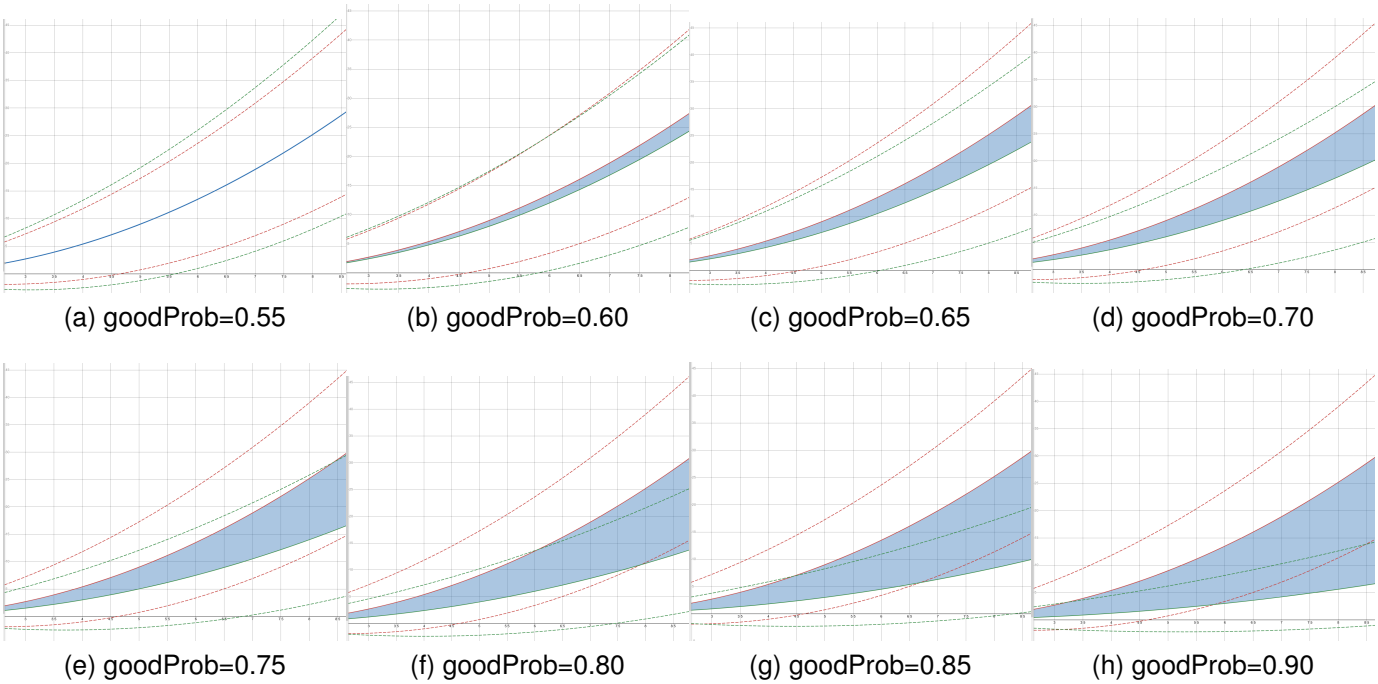

Fig. 3. Red shows mean of bad processes $\pm$ one standard deviation. Green shows mean of mean of good processes $\pm$ one standard deviation. Blue shows difference between the mean of the good processes and bad processes. x-axis is k number of candidates and y-axis is the Spearman-Footrule distance