Building Consensus via Iterative Voting

Farzad Farnoud (Hassanzadeh)[*], **Eitan Yaakobi**[†], **Behrouz Touri**[*], **Olgica Milenkovic**[*], and **Jehoshua Bruck**[†]
[*]Department of Electrical and Computer Engineering, University of Illinois, Urbana Champaign, Urbana, Illinois 61801, U.S.A.
[†]Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125, U.S.A
{hassanz1, touri1, milenkov}@illinois.edu, {yaakobi, bruck}@caltech.edu

***Abstract*—THIS PAPER IS ELIGIBLE FOR THE STUDENT PAPER AWARD. In networked systems comprised of many agents, it is often required to reach a common operating point of all agents, termed the *network consensus*. We consider two iterative methods for reaching a ranking (ordering) consensus over a voter network, where the initial preference of every voter is of the form of a full ordering of candidates. The voters are allowed, one at a time and based on some random scheme, to change their vote to bring them "closer" to the opinions of selected subsets of peers. The first consensus method is based on changing votes one adjacent swap at a time; the second method is based on changing a vote via averaging with the votes of peers, potentially leading to many adjacent swaps at a time vote. For the first model, we characterize convergence points and conditions for convergence. For the second model, we prove convergence to a global ranking and derive the rate of convergence to this consensus.**

I. Introduction

In the past two decades, the theory of rank aggregation and social choice has received renewed attention because of its wide-ranging applications, including recommender systems (collaborative filtering), meta-search engines, decision making among humans or automated agents, sports and other competitions, as well as elections [6], [9], [13], [14], [17].

Rank aggregation refers to a class of methods for collecting preferences and collectively making decisions based upon them. In particular, every voter has a preference ordering, which may be a full (or partial) ranking of candidates based on the voter's preferences. A unique voters' representative full (or partial) ranking of the candidates is determined based upon some decision rule applied to the set of individual rankings. There exist many decision rules for vote aggregation, including the simple majority and plurality rules, Borda's scoring method, and the Kemeny-Young approach [2], [3].

In most voting decision rules, it is assumed that the aggregate is formed in one step, without allowing the voters to adapt their rankings to the rankings of other voters. However, in many applications, a different approach – dynamic voting over networks – may be required or may actually represent a more realistic model of social opinion aggregation. One may regard this dynamic procedure as *iterative voting*, where voters are allowed to change their preference orderings based on fully or partially observed preferences of other voters. An equilibrium point is reached when no voter seeks to change its preference. This approach is particularly useful for planning in autonomous systems, consensus decision making among humans in social networks, and modeling the evolution of preferences of individuals. Examples include *doodle* and *whenisgood* event schedulers, and consensus decision making in Wikipedia [20].

We consider the problem of reaching consensus in iterative voting systems wherein every voter assigns its initial preference in the form of a full ordering of all candidates. Information about preferences of voters is disseminated by either public announcements or by communicating along the edges of a given connectivity graph. Based on the information shared by other voters, voters are allowed to change their votes one at a time and based on some random scheme, so as to increase general agreement in the group. Notice that the goal of the iterative exchange of information is to achieve consensus on the ranking of candidates, thereby eliminating the need to track quantitative information of the voters.

As already mentioned, there exists a number of rank aggregation methods that may be considered in an iterative setting and for which voters may be assumed to communicate within a networked environment. We focus on two such models that fall into two conceptually different classess of aggregation methods, namely distance-based and score-based rules. Distance based aggregation amounts to finding the median of a set of rankings, according to a predefined distance function. Score based aggregation is based on the simple idea that each position in an ordering has a score associated with it, which may be used to assess the global ranking of a candidate. Furthermore, to illustrate the full generality of the class of iterative voting problems, we consider two different voter communication strategies.

The first model is based on pairwise comparisons, and it falls into the family of so called Condorcet methods. Condorcet methods are aggregation rules that rank the candidate that wins every pairwise for which the solution is guaranteed to be transitive and hence a linear order. The underlying distance measure is the Kendall distance [2]. In this scheme, each voter starts with a preference ordering of all candidates which is *known to all other voters*. At each time instant, a voter is chosen randomly. The chosen voter then tries to identify two adjacently ranked candidates a and b, such that at least as many voters prefer b to a as they prefer a to b. If such a pair exists, the voter switches the positions of a and b in her ranking. Performing such an operation is guaranteed to decrease the sum of pairwise Kendall tau distances between the voters. This model is combinatorial in nature, and discussed in Section II.

The second model is based on the simple Borda scoring method for rank aggregation. In the Borda method a candidate is awarded a number of points for each vote that depends on its position in that vote. The ranking in the consensus is determined based on the average number of points a candidate is awarded, with ties broken arbitrarily. In our model, we assume that the voters are only allowed to communicate on edges of a given social interaction graph, in which vertices correspond to voters. At each time, an edge is chosen at random and the corresponding voters are asked to average their

Borda scores. This model resembles classical gossiping over networks, with the major difference being that one requires the consensus to be achieved *only on the ordering of the candidates*, and not on their total score. The method is discussed in Section III.

In both models, the voters are assumed to be truthful, which eliminates the option of strategic voting studied in [17], [18].

Related Work: The results closest in theme to the one we discuss are the recent analysis by Ephrati and Rosenschein [6], who proposed a voting algorithm to reach consensus which is resistant to strategic voting and where agents do not have to reveal their entire preference orderings; the work by Meir et al. [17], introducing iterative voting based on the plurality rule and using game theory; and a recent extension of the aforementioned results reported in [18] and [19], including the k-approval, Veto, and Borda rules. In all of the above papers, it is assumed that a vote aggregation rule is used to obtain a representative ranking after votes are not changed any longer by voters. Note that it is not required to achieve consensus, i.e. to have all voters reach a same ordering of candidates. This is in stark contrast with our scheme, in which all candidates are required to change their ranking until reaching the same preference ordering.

In [16], the authors studied consensus in a network of voters, where a voter may adopt the opinion of one of her neighbors at random, or adopt the opinion of the majority of her neighbors. In our work, each opinion is represented by a *ranking of a set of candidates*, while in [16], opinions are chosen from a finite set that has no particular structure; there, it was also assumed that a vote can be changed to any other vote. The schemes described in what follows allow a voter only to make "small" changes at any time.

II. Consensus via Distance Based Rank Aggregation

Assume there are n voters and m candidates. Every voter provides a full ordering of m candidates, i.e., a permutation σ in $\mathbb{S}_m$, the symmetric group of order $m!$. For $\sigma, \pi \in \mathbb{S}_m$, we let $d_\tau(\sigma, \pi)$ denote the *Kendall τ distance* between σ and π, that is, the smallest number of adjacent swaps needed to transform σ into π.

The vector of votes $\Sigma = (\sigma_1, \dots, \sigma_n)$ is called a *profile*. For every two different candidates $a, b \in [m]$, where $[m] = \{1, \dots, m\}$, $n_{ab}(\Sigma)$ denotes the number of voters who rank a higher than b. Hence, $n_{ab}(\Sigma) + n_{ba}(\Sigma) = n$. To simplify notation, whenever the profile Σ is clear from the context we simply write n_{ab} instead of $n_{ab}(\Sigma)$. We also say that candidate a is *winning* against candidate b if $n_{ab} > n_{ba}$.

The sum $\sum_{j<i} d_\tau(\sigma_i, \sigma_j)$ is the *profile potential* and is denoted by $\psi(\Sigma)$. It is easy to see that

$$\psi(\Sigma) = \sum_{1 \leqslant j < i \leqslant n} d_\tau(\sigma_i, \sigma_j) = \sum_{1 \leqslant a < b \leqslant m} n_{ab}(n - n_{ba}).$$

The vote aggregation algorithm, Algorithm 1, mimics a social process in which each voter may change her opinion regarding the relative order one pair of candidates at a time. Such a change is the smallest change that can be applied to a ranking. Furthermore, *any* change in a voter's ranking can be viewed as a sequence of changes in the relative order of pairs of candidates. Note that changing the relative order of two candidates in a ranking is only possible if the candidates are adjacent in that ranking.

The algorithm may be succinctly described as follows: at each iteration, one of the voters is allowed to change her ranking $\sigma = (\cdots, a, b, \cdots)$ by swapping two adjacent candidates a and b, resulting in the ranking $\sigma' = (\cdots, b, a, \cdots)$; the swap is allowed iff there are at least as many voters that prefer b to a as there are voters that prefer a to b (equivalently, a voter is allowed to change her/his vote from σ to σ' if $d_\tau(\sigma, \sigma') = 1$ and the value of the potential function decreases by introducing the swap). Consensus or global agreement is reached when all voters reach an agreement on the same ranking, or alternatively, when the potential function of the votes equals zero. If the potential function does not equal zero, but no voter is allowed to change her vote, we say that the algorithm reached a local minima.

Algorithm 1. ITR-VOT$(\sigma_1, \dots, \sigma_n)$
Input: A profile of n rankings: $(\sigma_1, \dots, \sigma_n) \in (\mathbb{S}_m)^n$.
1. $\boldsymbol{b} = (b_1, \dots, b_n) = (0, \dots, 0) \in \{0, 1\}^n$.
2. For $i = 1, \dots, n$ do:
   if $\exists\ a, b$ s.t. $\sigma_i = (\cdots ab \cdots)$ and $n_{ba} \geqslant n_{ab}$, set $b_i = 1$.
3. If $\boldsymbol{b} = (0, \dots, 0)$, Return $(\sigma_1, \dots, \sigma_n)$.
4. Choose randomly $1 \leqslant i \leqslant n$, such that $b_i = 1$,
   and find a pair a, b satisfying the condition in Step 2.
5. Swap a and b in σ_i to generate σ_i'.
6. Replace σ_i with σ_i' in the profile.
7. Go to step 1.

Remark 1. Note that while Algorithm 1 is not described as a distributed algorithm, it can be easily rewritten in this form: voters, one at a time and in some arbitrary order, check if they are allowed to make an adjacent swap. The process terminates once no changes to votes are allowed.

Let $\pi_1, \dots, \pi_{m!}$ be some ordering of the $m!$ permutations in $\mathbb{S}_m$. Given a profile $\Sigma \in (\mathbb{S}_m)^n$, we define $m!$ variables $n_1, \dots, n_{m!}$, where n_j indicates the number of voters who decided in favor of the j-th ranking, π_j. Note that $\sum_{j=1}^{m!} n_j = n$. With a slight abuse of notation, we also call the vector $\boldsymbol{n} = (n_1, \dots, n_{m!})$ a *profile*, since it provides the same information as the previously defined list of profiles.

We say that a profile $\boldsymbol{n} = (n_1, \dots, n_{m!})$ has a *good move* if there exists an allowed swap as defined in Step 2 of Algorithm 1. A good move decreases the profile potential. A profile $\boldsymbol{n}$ is called a *good profile* if it has a good move, and is called a *bad profile* otherwise. Note that Algorithm 1 terminates when it reaches a bad profile. A bad profile is also called a *local minimum* since it is a local minimum point for the profile potential function ψ. We also say that the algorithm *converged* when reaching a bad profile if all voters arrived at the same permutation, and say that the algorithm *diverged* otherwise.

Example 1. Consider the following profile

$$\Sigma_1 = (123, 231, 132).$$

In this case, both the second and third voter have a good move. The second voter may change her permutation to 213, while

the third voter may change her vote to 123. Assume that the second voter changed her permutation. In the following step, the same voters have a good move. Assume now that the third voter changed her permutation to 123. In the last step, the second voter will change her permutation to 123 as well. Hence, Algorithm 1 converges to $(123, 123, 123)$ given the profile Σ_1.

As another example, note that Algorithm 1 diverges on the profile $\Sigma_2 = (123, 231, 312)$.

A. Characterizing Profiles

In what follows, we characterize some useful properties of profiles that will allow us to assess when Algorithm 1 converges. We start with describing a sufficient condition for a good profile.

Lemma 1. *A profile $\boldsymbol{n}$ is good if there exist two candidates a, b and two rankings σ_i, σ_j, such that σ_i is of the form $\sigma_i = \cdots ab \cdots$, while σ_j is of the form $\sigma_j = \cdots ba \cdots$.*

Proof: Suppose this is not the case, i.e., $\boldsymbol{n}$ is a bad profile. Since swapping a and b in σ_i is not a good move, we have $n_{ab} > n_{ba}$. Similarly, since swapping b and a in σ_j is not a good move, we have $n_{ba} > n_{ab}$, which leads to a contradiction. ∎

Lemma 1 implies that a profile $\boldsymbol{n}$ at a local minimum cannot contain two votes differing in a single adjacent swap, or a vote and its reverse.

Lemma 2. *Let $\boldsymbol{n}$ be a profile at a local minimum and σ a permutation of one of the voters. Then the permutation σ eliminates*

$$N(m) = \sum_{k=1}^{m-1} (-1)^{k+1} \binom{m-1}{k} (m-k)! \approx m! - \frac{m!}{e}$$

other permutations from being included in the profile.

From Lemma 2, it follows that a bad profile can have approximately at most $\frac{m!}{e}$ different permutations. We next show how to improve this upper bound.

For a given profile $\boldsymbol{n}$, we construct its *profile graph*, $G_{\boldsymbol{n}} = (V_{G_{\boldsymbol{n}}}, E_{G_{\boldsymbol{n}}})$, also known as a *tournament graph*, as follows. The vertices set of the profile graph consists of all candidates, i.e. $V_{G_{\boldsymbol{n}}} = [m]$, and there is an edge (a, b) if candidate a is winning against candidate b (i.e., if $n_{ab} > n_{ba}$). For simplicity, we assume that the number of voters is odd so that it is impossible to have $n_{ab} = n_{ba}$ for any pair a, b. As a result, for any profile $\boldsymbol{n}$, the graph $G_{\boldsymbol{n}}$ is a complete directed graph with no inner loops or symmetric edges. The next lemma is a generalization of Lemma 1.

Lemma 3. *If there exist two candidates a, b and a voter with ranking $\sigma_i = \cdots ab \cdots$, while $(b, a) \in E_{G_{\boldsymbol{n}}}$, then $\boldsymbol{n}$ is a good profile.*

Proof: Since $(b, a) \in E_{G_{\boldsymbol{n}}}$, we know that $n_{ba} > n_{ab}$. Therefore, according to the conditions of Step 2 in Algorithm 1, the i-th voter has a good move and thus $\boldsymbol{n}$ is a good profile. ∎

From Lemma 3 we conclude that a voter cannot reduce the potential function value if its permutation corresponds to a Hamiltonian path on the graph $G_{\boldsymbol{n}}$. Hence, given an upper bound on the number of different Hamiltonian paths in a tournament graph, one may derive an upper bound on the number of different permutations in a bad profile. Let $T(m)$ denote the maximum number of Hamiltonian paths in a tournament graph with m candidates. Alon [5] proved that $T(m) < cm^{3/2} \cdot m! 2^{-(m-1)}$ and Friedgut and Kahn [10] subsequently showed that $T(m) < O(m^{3/2-\xi} m! 2^{-m})$, where $\xi < 0.2507\ldots$. We hence have the following lemma.

Lemma 4. *If the number of different permutations in a profile is greater than $T(m)$, then the profile is necessarily a good profile.*

When combined, the above lemmas lead to the theorem below.

Theorem 5. *If n is large enough, then the fraction of good profiles is at least*

$$\frac{\sum_{\ell=T(m)+1}^{m!} \binom{m!}{\ell} \cdot \ell! \cdot S(n, \ell)}{(m!)^{\ell}},$$

where $S(n, \ell) = \frac{1}{\ell!} \sum_{t=0}^{\ell} (-1)^{\ell-t} \binom{\ell}{t} t^n$. In particular, as n grows, this fraction approaches one.

A direct result of the above theorem is that the number of local minimum points to the total number of points in $(\mathbb{S}_m)^n$ converges to 0 as $n \to \infty$.

B. Convergence Properties

In voting theory, a *Condorcet winner* is a candidate who is winning against every other candidate in pairwise competition. Note that in general a Condorcet winner may not exist. In the tournament graph, a Condorcet winner corresponds to a vertex with zero in-degree. A *Condorcet loser* is defined similarly.

Lemma 6. *If the initial profile has a Condorcet winner, then after Algorithm 1 terminates, the Condorcet winner is ranked first in every vote in the resulting profile.*

Similarly, it is possible to show that if there is a Condorcet loser, after the algorithm terminates, the Condorcet loser is ranked last in every vote. We omit the proof of Lemma 6 as it is a special case of a more general lemma, Lemma 7, stated below.

A *winning set* in a profile is a set $S \subseteq [m]$ such that every candidate $a \in S$ is winning against every candidate $b \notin S$.

Lemma 7. *Assume $S \subseteq [m]$ is a winning set in the initial profile. After Algorithm 1 terminates, every $a \in S$ is ranked higher than every $b \notin S$ in every vote in the resulting profile.*

Proof: Without loss of generality, assume that S is the set $\{1, \ldots, i\}$, for some $i \in [m]$. At any step of Algorithm 1, assume that there is a ranking σ_j of some voter j such that at least one candidate indexed by an element in $\{1, \ldots, i\}$ is ranked lower than at least one candidate indexed by an element in $\{i+1, \ldots, m\}$. Then, there are two candidates $1 \leqslant k \leqslant i$ and $i+1 \leqslant \ell \leqslant m$ such that σ_j is of the form $\sigma_j = \cdots \ell k \cdots$. Thus, swapping k and ℓ is a good move. When the algorithm

terminates, there are no good moves, so all the candidates in S have to be ranked higher than those in $[m] \setminus S$ in every vote. ■

If the graph $G_{\boldsymbol{n}}$ does not have a cycle, then there exists a permutation $\pi^* = a_1 \cdots a_m$ such that $\{a_1, \ldots, a_i\}$ is a winning set for each $i \in [m]$. Hence, if $G_{\boldsymbol{n}}$ has no cycles, then Algorithm 1 converges to π^*. Furthermore, π^* is the solution to the Kemeny-Young problem,

$$\arg\min_{\pi \in \mathbb{S}_m} \sum_{i=1}^{m} d_\tau(\pi, \sigma_i).$$

In other words, π^* is the median of Σ under the Kendall τ distance. The number of steps needed for the algorithm to converge to π^* equals

$$\sum_{i=1}^{m} d_\tau(\pi^*, \sigma_i),$$

since each move decreases the distance of a ranking to π^* by one.

On the other hand, whenever there is a cycle in $G_{\boldsymbol{n}}$, Algorithm 1 diverges. To see this, note that Algorithm 1 cannot change the directions of the edges in the tournament graph, so it is not possible for the algorithm to "break" a cycle. Thus, we have the following theorem.

Theorem 8. *Algorithm* 1 *converges if and only if the graph* $G_{\boldsymbol{n}}$ *does not have a cycle.*

It is known that if every voter draws her ranking uniformly at random from $\mathbb{S}_m$ (sometimes called the *impartial culture* model), then the tournament graph will have a cycle with probability which is approaching one as the number of candidates grows; see e.g. [4], [7]. Therefore, the same statement holds regarding the convergence of Algorithm 1, even though if n is large enough, the fraction of good states approaches one. In practical social choice scenarios, cycles occur rarely [8], [12].

It is also possible to extend Algorithm 1 so as to accommodate more than one adjacent swap, i.e. to accommodate the update procedure to include permutations at distance larger than one but less than b from each other. An interesting result is that if $b = m - 1$, then every profile is a good profile and thus Algorithm 1 always converges. Furthermore, the value $b = m - 1$ is tight in terms of ensuring that every profile is a good profile. Details of this analysis are omitted due to lack of space.

III. Consensus via Iterative Borda Scoring

We describe next a method for iterative vote aggregation based on Borda [1]. The gist of the approach is to assign scores to candidates based on their ranking: in any ordering, the first candidate receives m points, the second candidate receives $m - 1$, points and so on. The total score of a candidate equals the cumulative sum of her scores over all votes cast, and ranking is performed in decreasing order of the scores. Other scoring methods are possible as well, for example assigning score s_i to the ith ranked candidate, where $s_1, s_2, \cdots, s_m$ is a predetermined decreasing sequence.

In this setting, consensus may be achieved through gossiping protocols [15] in a networked set of voters. The voter network $G = ([n], E)$ is a connected, undirected graph with vertices representing the voters, and an edge set E that captures the social connectivity patterns of the voters. Initially, each vertex i has a preference ordering σ_i of m candidates, based on which an initial score is assigned to each candidate. Let $b_i(0)$ denote the vector of initial scores assigned by voter i, based on some initial ordering of the candidates. At each time instance, one picks an edge $\{x, y\} \in E$ with probability $p_{xy} > 0$, where $\sum_{\{x,y\} \in E} p_{xy} = 1$. Then, agents x and y each update their score vector to the average of the two vectors. The procedure is repeated until the correct global ranking is reached, if allowed by the distributed protocol

Algorithm 2. ITR-BORDA

For $t = 0, 1, 2, \ldots$ do:

1. Pick an edge $\{i, j\} \in E$ with probability p_{ij}.
2. Let voters i, j exchange their estimates $b_i(t), b_j(t)$.
3. Let $b_i(t+1) = b_{i'}(t+1) = \frac{1}{2}(b_i(t) + b_{i'}(t))$.
4. For $\ell \neq i, j$, set $b_\ell(t+1) = b_\ell(t)$.
5. Stop when *all voters agree on one ranking*.

Let $\bar{b} = \frac{1}{m} \sum_{i=1}^{m} b_i(0)$. As a direct consequence of the results in [11], we have the following lemma.

Lemma 9 *If* $G = ([m], E)$ *is connected, then* $\lim_{t \to \infty} b_i(t) = \bar{b}$ *holds almost surely.*

As a result, the score vector of each candidate converges to the average score of the candidate, i.e., its Borda score. However, our goal is to find the correct ordering of candidates based on their score in $\bar{b}$, rather than the vector $\bar{b}$ itself. Thus, it is not important that the estimates of the ranking vectors converges to $\bar{b}$, but that the estimates of the actual ranks are correct. In other words, if for some time t, for all agents i, the ordering of $b_i(t)$ matches the ordering of $\bar{b}$ for all $i \in [m]$, then the society has achieved consensus over the *ranking* of the candidates. We derive next a probabilistic bound on the number of iterations needed to reach the Borda ranking in a distributed fashion.

Without loss of generality, assume that $\bar{b}$ is ordered, i.e. $\bar{b}^1 \leqslant \bar{b}^2 \leqslant \cdots \leqslant \bar{b}^n$, with superscripts denoting candidates. We say that t is a consensus time for the *aggregate ranking* if the ordering of $\bar{b}_i(t)$ matches the ordering of $\bar{b}$ for all $i \in [m]$. The following result is a consequence of this definition.

Lemma 10 *If* t *is a consensus time for the ranking, then any* $t' > t$ *is a consensus time for the ranking.*

Proof: It suffice to show the result for $t' = t + 1$. Let $\{i, i'\}$ be the edge that is chosen randomly at time t. Since t is a consensus time for the ranking, we have $b_i^1(t) \leqslant \cdots \leqslant b_i^n(t)$ and $b_{i'}^1(t) \leqslant \cdots \leqslant b_{i'}^n(t)$, and thus we also have

$$\begin{aligned} b_i^1(t+1) &= \frac{1}{2}\left(b_i^1(t) + b_{i'}^1(t)\right) \\ &\leqslant \cdots \leqslant b_i^n(t+1) \\ &= \frac{1}{2}\left(b_i^n(t) + b_{i'}^n(t)\right), \end{aligned}$$

which proves the claim. ■

Define *the consensus time* as:

$$T = \min\{t \geqslant 0 \mid t \text{ is a consensus time for the ordering}\}.$$

Note that for the random gossip scheme, T is a random variable and if we have an adapted process for the random choice of edges, T is a stopping time. Our goal is to provide a probabilistic bound for T.

To this end, let $r^j = \min\left\{\bar{b}^{j+1} - \bar{b}^j, \bar{b}^j - \bar{b}^{j-1}\right\}$ and let $d^j = \max_i b_i^j(0) - \min_i b_i^j(0)$. That is, r^j equals the minimum distance of the average rating of j from its neighboring candidates, while d^j equals the spread of the initial ratings of the agents for the candidate j. Then, we have the following result.

Theorem 11 *The consensus time T of the ordering satisfies*

$$P(T > t) \leqslant 4m\lambda_2^t(W) \sum_{j=1}^{n} \left(\frac{d^j}{r^j}\right)^2,$$

where $W = \sum_{\{i,i'\}\in E} P_{ii'}\left(I - \frac{1}{2}(e_i - e_{i'})(e_i - e_{i'})^T\right)$, $e_i = [0 \cdots 0\ 1\ 0 \cdots 0]^T$ is an $m \times 1$ vector with ith element equal to one, and $\lambda_2(W)$ is the second largest eigenvalue of W.

Proof: Let $b^j(t)$ be the score vector of candidate j at time t, obtained from the votes of m agents and let $y^j(t) = b^j(t) - \bar{b}^j$. Note that $\|y^j(t)\|^2 \leqslant \left(\frac{r^j}{2}\right)^2$ implies that $|b_i^j(t) - \bar{b}^j| \leqslant \frac{r^j}{2}$, $\forall i \in [n]$. Thus, if $\|y^j(t)\|^2 \leqslant \left(\frac{r^j}{2}\right)^2$ holds $\forall j \in [n]$, then

$$b_i^j(t) \leqslant \bar{b}^j + \frac{r^j}{2} \leqslant \frac{1}{2}(\bar{b}^j + \bar{b}^{j+1}),$$

where the last inequality follows from the fact that $r^j \leqslant \bar{b}^{j+1} - \bar{b}^j$. Similarly, we have:

$$b_i^{j+1}(t) \geqslant \bar{b}^{j+1} - \frac{r^{j+1}}{2} \geqslant \frac{1}{2}(\bar{b}^{j+1} + \bar{b}^j),$$

which follows from $r^{j+1} \leqslant \bar{b}^{j+1} - \bar{b}^j$. Hence,

$$\begin{aligned} b_i^1(t) &\leqslant \frac{1}{2}(\bar{b}^2 + \bar{b}^1) \leqslant b_i^2(t) \leqslant \frac{1}{2}(\bar{b}^3 + \bar{b}^2) \\ &\leqslant \cdots \leqslant \frac{1}{2}(\bar{b}^{n-1} + \bar{b}^n) \leqslant b_i^m(t), \end{aligned}$$

and t is a consensus time for the algorithm. Thus,

$$\{T > t\} \subseteq \bigcup_{j=1}^{n} \left\{\|y^j(t)\|^2 \geqslant \left(\frac{r^j}{2}\right)^2\right\},$$

which, after invoking the union bound, leads to

$$P(T > t) \leqslant \sum_{j=1}^{n} P\left(\|y^j(t)\|^2 \geqslant \left(\frac{r^j}{2}\right)^2\right). \qquad (1)$$

Markov's inequality gives

$$P\left(\|y^j(t)\|^2 \geqslant \left(\frac{r^j}{2}\right)^2\right) \leqslant \left(\frac{2}{r^j}\right)^2 E\left[\|y^j(t)\|^2\right].$$

Using the analysis in [11], it can be shown that

$$E\left[\|y^j(t)\|^2\right] \leqslant \lambda_2^t(W)\|y^j(0)\|^2 \leqslant m\lambda_2^t\left(d^j\right)^2.$$

Combining the above two relations, we find

$$P\left(\|y^j(t)\|^2 \geqslant \left(\frac{r^j}{2}\right)^2\right) \leqslant 4m\lambda_2^t\left(\frac{d^j}{r^j}\right)^2.$$

Replacing the last inequality in (1) proves the assertion. ■

Note that from [11] it follows that if G is connected, then $\lambda_2 < 1$, and thus the probability $P(T > t)$ decays exponentially.

IV. Acknowledgments

The authors are grateful to Angelia Nedich for useful discussions. The work was partially supported by NSF grants CCF 0939370, CCF 0644427, ECCS-0801795 and BSF grant 2010075.

References

[1] J.-C. Borda, "Mémoire sur les élections au scrutin," *Histoire de l'Académie royale des sciences*, 1784.

[2] J. G. Kemeny, "Mathematics without numbers," *Daedalus*, vol. 88, no. 4, pp. pp. 577–591, 1959.

[3] H. Young and A. Levenglick, "A consistent extension of condorcet's election principle," *SIAM J. Appl. Math.*, vol. 35, no. 2, pp. 285–300, 1978.

[4] W.H. Riker, *Liberalism against Populism*, W. H. Freeman and Co., San Francisco, 1982.

[5] N. Alon, "The maximum number of Hamiltonian paths in tournaments," *Combinatorica*, vol. 10, pp. 319–324, 1990.

[6] E. Ephrati and J. Rosenschein, "Multi-agent planning as a dynamic search for social consensus," in *Proc. 13 Int. Joint Conf. on Artificial Intelligence (IJCAI)*, vol. 1, (San Mateo, CA), pp. 423–9, 1993.

[7] K. Shepsle and M. Bonchek, *Analyzing Politics: Rationality, Bhavior and Institutions*, Norton, New York, 1997.

[8] A.S. Tangian, "Unlikelihood of Condorcet's paradox in a large society," *Soc. Choice Welfare*, vol. 17, pp. 337–365, 2000.

[9] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation revisited." Manuscript, Available: www.eecs.harvard.edu/ michaelm/CS222/rank2.pdf, 2001.

[10] E. Friedgut and J. Kahn, "On the number of Hamiltonian cycles in a tournament," *Combinatorics, Probability and Computing*, vol.,14, pp. 769–781, 2005.

[11] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[12] M. Regenwetter, B. Grofman, A.A.J. Marley, and I.M. Tsetlin, *Behavioral Social Choice*, Cambridge University Press, 2006.

[13] V. Conitzer, A. Davenport, and J. Kalagnanam, "Improved bounds for computing kemeny rankings," in *Proc. 21st National Conf. on Artificial Intelligence*, (Boston, Massachusetts), 2006.

[14] F. Schalekamp and A. van Zuylen, "Rank aggregation: Together we're strong," *Proc. of 11th ALENEX*, pp. 38–51, 2009.

[15] D. Shah, *Gossip Algorithms*, vol. 3. Foundations and Trends in Networking, 2009.

[16] M. Yildiz, R. Pagliari, A. Ozdaglar, and A. Scaglione, "Voting models in random networks," in *Information Theory and Applications Workshop (ITA)*, pp. 1 –7, Feb. 2010.

[17] R. Meir, M. Polukarov, J.S. Rosenschein, and N.R. Jennings, "Convergence to equilibria in plurality voting," *Proc. 24th Conf. on Artificial Intelligence*, pp. 823–828, Atlanta, July 2010.

[18] O. Lev and J.S. Rosenschein, "Convergence of iterative voting," *Proc. 11th Int. Conf. on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 611–618, Valencia, Spain, June 2012.

[19] R. Reyhani and M.C. Wilson, "Best reply dynamices for scoring rules," *Proc. Int. Conf. on Artificial Intelligence*, Las Vegas, Nevada, July 2012.

[20] *http://en.wikipedia.org/wiki/Wikipedia:Consensus*, Wikipedia Concensus Policy.