

Discordant voting protocols for cyclically linked agents

András Pongrácz*

Department of Algebra and Number Theory
University of Debrecen
Egyetem square 1, Debrecen, Hungary 4032
pongrazc.andras@science.unideb.hu

Submitted: Mar 12, 2019; Accepted: Feb 9, 2020; Published: Mar 20, 2020

© The author. Released under the CC BY-ND license (International 4.0).

Abstract

Voting protocols, such as the push and the pull protocol, model the behavior of people during an election. These processes have been studied in distributed computing in peer-to-peer networks, and to describe how viruses or rumors spread in a community. We determine the asymptotic behavior of the runtime of discordant linear protocols on the cycle graph and the probability for each consensus to win.

Mathematics Subject Classifications: 91A22, 60J10, 60G50

1 Introduction

Models of voting in finite graphs have been studied intensively for decades, see e.g., [6, 16, 12, 1, 15, 7]. Throughout this paper, a discrete-time voting protocol is defined by specifying a graph and a set of nondeterministic rules. Then the process is divided into rounds. In each round, the participants, i.e., vertices of the graph, can affect the vote of their neighbors according to the given rules.

We note that many alternative definitions were investigated in the literature. Continuous time voting processes were studied in [6, 8]. Somewhat surprisingly, the thorough mathematical investigation of the continuous version preceded that of the discrete analogue of the protocols [16, 8]. In [10] the graph evolves together with the opinions of the vertices. This models the behavior of people who in each round try to convince one

*This work is supported by the EFOP-3.6.2-16-2017-00015 project, which has been supported by the European Union, co-financed by the European Social Fund. The paper was also supported by the National Research, Development and Innovation Fund of Hungary, financed under the FK 124814 and PD 125160 funding schemes, the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, and by the ÚNKP-18-4 and ÚNKP-19-4 New National Excellence Program of the Ministry of Human Capacities.

another and succeed with a given probability. Whenever they fail, they cease to communicate with each other, that is, we delete the edge linking them from the graph. In such a model there are many potential final results, as the graph can disconnect, and in fact, we may end up with many connected components. For more details, see [7, 2]. The application of these randomized protocols in studying how rumor spreads in a society goes back to decades, and it is still an active area [9, 12, 1]. The same can be said about peer-to-peer networks, see e.g., [19, 15, 14]. In this application, opinion is replaced by a piece of information that each computer has at a given time, and they share the data in a randomized way. Connections of voting processes and coalescing random walks were investigated in [8, 17], and for other recent applications see [18, 4].

However, we consider discrete time voting models where the graph is fixed, and the vote is a binary decision. The two options to choose from are 0 and 1, but we usually refer to vertices with opinion 0 as *blue* vertices, and *red* vertices are the ones with opinion 1. Such a protocol can be synchronous (see [5] for examples), i.e., it is allowed that several vertices of the graph change their opinion in one round; otherwise it is asynchronous. The so-called linear voting model was introduced in [5] as a common generalization of many well-studied voting protocols. Three of the most common special cases of asynchronous linear voting are the

- Oblivious protocol: in each round an edge uv is chosen uniformly at random, and then either u adopts the opinion of v or the other way around, with equal probability.
- Push protocol: in each round a vertex u is chosen uniformly at random, and that vertex forces a randomly chosen neighbor to adopt the opinion of u .
- Pull protocol: in each round a vertex u is chosen uniformly at random, and that vertex is forced by a randomly chosen neighbor v to adopt the opinion of v .

From a practical viewpoint, all linear voting models have a common weakness: it is typical that nothing changes in many steps of the process, as it is possible that every participant keeps his own opinion for the next round. E.g., consider push, pull or oblivious voting on the complete graph K_n ; in this particular case, the three protocols coincide. If one opinion is significantly more popular than the other, then with very high probability, both chosen vertices have the more popular opinion. So usually many idle rounds go by before the opinion of some vertex is altered. This example demonstrates the advantage of discordant (oblivious, push, pull) voting protocols, defined in [3]. An edge uv is discordant if u and v have different opinion, and a vertex is discordant if it is in a discordant edge. To define discordant oblivious, push and pull voting, the above three definitions are modified so that whenever a random choice is made, we only allow discordant edges or vertices to be picked (always uniformly at random). Note that in our restricted framework when there are only two opinions, the definition of discordant pull voting simplifies to picking a discordant vertex in each round randomly and switching its opinion.

The goal of every voting scheme that we study now is to reach a consensus, that is, a state where all participants have the same opinion. The topic of the present paper is the expected time to reach consensus with the discordant push, pull and oblivious processes on

the n -cycle. It was proven in [3] that all three processes have a quadratic runtime at worst. In particular, push voting is expected to terminate in at most $33n^2$ steps regardless of the initial state, and from some initial state it is indeed expected to take at least $n^2/4 + O(n)$ time to reach a unanimous vote [3, Section 4]. We improve the bounds and obtain that the precise asymptotical behavior of the worst expected runtime of all three discordant protocols on the n -cycle is $n^2/4 + O(n^{3/2})$, and an effective constant can be given in the error term for each of these protocols. Furthermore, we show that if β and ϱ denote the number of blue and red vertices in the initial state, respectively, then $\beta\varrho$ is in general a good estimate to the expected runtime of these protocols, provided that the initial state be tame. More precisely, in a non-consensus state, there is the same number r of blue and red *runs*, i.e., maximal sets of consecutive vertices of the same color. Then the difference of the expected runtime of each of the three protocols from $\beta\varrho$ is at most $O(rn)$, with an effective constant in the error term.

The other vital problem in case of a random protocol is to compute the probability of each possible outcome to win. We show that in case of the cycle graph the probability of each opinion to win with the discordant push, pull or oblivious protocol is asymptotically proportionate to the number of vertices with that opinion in the initial state, provided that the initial state be tame. More precisely, the blue vertices have winning probability $\beta/n + O(r/n)$, with an effective constant in the error term. By using some probability theory, it can be shown that there must be a state for arbitrarily large n such that the estimation β/n has error 0.005 or more. However, computer simulations suggest that in highly symmetrical initial states (such as the one with alternating runs of lengths one and two), the estimation β/n is quite accurate, a phenomenon we cannot explain yet.

Although some parts of the proof of the positive results require elaborate combinatorial and probabilistic arguments, the core is an elementary linear algebraic lemma (Lemma 1). This paper is a demonstration of how the iterative application of that elementary lemma can yield asymptotically sharp results to basic questions about evolutionary processes, where the transition matrix is typically large but sparse and easy to describe.

2 Preliminaries

2.1 General tools

Throughout this section, P is an absorbing Markov chain with transient states Tran . We denote by Pen the set of potential *penultimate states* in Tran , that is, the states $t \in \text{Tran}$ such that the probability of moving from t to an absorbing state in one step is positive. As usual, we denote by Q the upper left minor of the canonical form of $P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$, see [13]. So Q is the transition matrix restricted to the transient states, and R is the matrix of transition probabilities between transient and absorbing states. Following standard notation, $N = (I - Q)^{-1}$ denotes the fundamental matrix of the Markov chain. In this paper, vectors are column vectors of length $|\text{Tran}|$, usually denoted by $\underline{u}, \underline{v}, \underline{\varepsilon}$, etc. The coordinates are identified with the transient states, so precisely speaking, these are vectors

in \mathbb{R}^{Tran} . We denote by $\underline{1}$ the column vector of length $|\text{Tran}|$ all of whose entries equal to 1. The entry corresponding to the coordinate t in the vector \underline{u} is denoted by $\underline{u}[t]$. It is well-known that if we sum up the entries $\underline{u}[t]$ while randomly walking on the coordinates starting from $t_0 \in \text{Tran}$, then the expected value of this sum before the walk is absorbed is $(N\underline{u})[t_0]$. In particular, the expected times to absorption from each transient state as initial state are the coordinates of the vector $N\underline{1}$; see [13] for further details.

The following lemma is the basic observation of the elementary method we use to improve the upper estimations for the expected time to absorption presented in [3]. We can think about $\underline{x}[t]$ as a “guesstimate” of the expected value of the sum of the entries of \underline{u} during a random walk with initial state t before reaching an absorbing state. In particular, if $\underline{u} = \underline{1}$, then \underline{x} is the guesstimate vector for the time to absorption starting from each transient state.

Lemma 1. *Let $\underline{u}, \underline{x}, \underline{\varepsilon} \in \mathbb{R}^{\text{Tran}}$ be vectors such that $Q\underline{x} = \underline{x} - \underline{u} + \underline{\varepsilon}$. Then $N\underline{u} = \underline{x} + N\underline{\varepsilon}$. In particular, if $Q\underline{x} \leq \underline{x} - \underline{u}$, then $N\underline{u} \leq \underline{x}$ (coordinate-wise).*

Proof. By rearranging the equation we obtain $\underline{u} = (I - Q)\underline{x} + \underline{\varepsilon}$. Multiplying both sides by $N = (I - Q)^{-1}$ yields $N\underline{u} = \underline{x} + N\underline{\varepsilon}$.

Moreover, $N = (I - Q)^{-1} = I + Q + Q^2 + \dots$ is a non-negative matrix. Hence, if $\underline{\varepsilon} \leq \underline{0}$ coordinate-wise, then $N\underline{\varepsilon} \leq N\underline{0} = \underline{0}$, thus $N\underline{u} \leq \underline{x}$. \square

As we mentioned earlier, the vectors $N\underline{u}$ and $N\underline{\varepsilon}$ are the expected value vectors of the sum of the entries of \underline{u} and $\underline{\varepsilon}$ during a random walk (on the coordinates) starting from each transient state. The above elementary lemma is particularly useful when the transition matrix is large but sparse, and the fundamental matrix cannot be computed or represented in a transparent way. This is often the case with evolutionary processes. Note that $Q\underline{x}$ is easy to compute if the matrix is sparse. Furthermore, because of the probabilistic interpretation of $N\underline{\varepsilon}$ and the possibility of applying Lemma 1 iteratively, it is possible to estimate this vector without computing N , as we see later. By successive application of this method, the error can shrink to such a small vector that it is very easy to estimate it, providing us with an efficient estimation of the expected value vector. We spell out an immediate application.

Lemma 2. *Let $\underline{u} \in \mathbb{R}^{\text{Tran}}$ be such that $\underline{u}[t] = 0$ for all $t \in \text{Tran} \setminus \text{Pen}$. Define $\underline{p} \in \mathbb{R}^{\text{Tran}}$ where $\underline{p}[t]$ is the probability of immediate absorption in state t . Let $M := \max_{t \in \text{Pen}} \underline{u}[t]/\underline{p}[t]$. Then the expected sum of the entries of \underline{u} during a random walk from any initial state is at most M .*

Proof. Apply Lemma 1 with the guesstimate vector $\underline{x} = M \cdot \underline{1}$. Note that $\underline{u} \leq M \cdot \underline{p}$, thus $\underline{x} - \underline{u} \geq M \cdot (\underline{1} - \underline{p}) = M \cdot Q\underline{1} = Q\underline{x}$. Hence, $N\underline{u} \leq \underline{x}$ coordinate-wise by Lemma 1. \square

This observation is very advantageous when we are able to cut a process to several phases, and we want to estimate the expected sum of an expression between two phase transitions. In our case, the phases are those parts of the process where the number of runs, i.e., maximal sets of consecutive vertices with the same opinion in the cycle, is

constant. We denote by r the number of blue runs; it is the same as the number of red runs in all states except for the two consensus states, where we set $r = 0$ by definition. Note that r never increases during the process, and it decreases if and only if the opinion of a singleton vertex is switched, in which case r decreases by one. The process halts when $r = 0$.

2.2 Further terminology

We now turn to the problems under consideration, defined in the introduction. Note that the proof is presented for discordant push voting on the n -cycle: the case of pull voting can be done in a similar fashion, and the case of oblivious voting is trivial. Clearly, the voting process is an absorbing Markov chain with 2^n states, whose absorbing states are exactly those two where all the vertices agree.

As in the introduction, the number of blue and red vertices are denoted by β and ρ , respectively. A vertex is a *singleton* if its color differs from both its neighbors' color. The number of singleton blue and singleton red vertices are s_β and s_ρ , respectively. The number of non-singleton blue vertices with (exactly) one red neighbor is m_β ; the number m_ρ is defined analogously for red vertices. A maximal set of consecutive singleton vertices is called an *arc*, and the number of arcs is denoted by ℓ .

3 Expected time to absorption on the cycle

It turns out to be advantageous in the calculation to cut the process into two parts. The first part of the process consists of the steps before we first reach a state with $r \leq 4\sqrt{5}\sqrt{n}$.

3.1 The first part: down to $4\sqrt{5}\sqrt{n}$ blue runs

In this subsection, we show an estimation of the expected length of the first part. The following bound can be extracted from [3, Section 4]. In that paper, a quadratic upper estimation was given to the runtime of the discordant push protocol using some results about stopped martingales. They obtained that it takes at most $33n^2$ steps to reach consensus from any initial state, that is, to reach a state with $r = 0$. However, by carefully modifying the calculation in [3], a more general result can be shown.

Proposition 3. *The expected time to reach a state with r runs is at most $40n^2/r$ from any initial state. In particular, putting $r = 4\sqrt{5}\sqrt{n}$, the first part is expected to terminate in at most $2\sqrt{5}n^{3/2}$ steps.*

Proof. In [3, Lemma 8] and the argument before that, it was shown that the expected time to reach a state with $r = r_1$ blue runs from one with $r = r_0$ blue runs is at most T^* , where T^* is the optimal solution of the following linear program:

$$T^* = \max 10\sqrt{2}n^{3/2} \sum_{r=r_1}^{r_0} \frac{x_r}{r^{3/2}}$$

$$\text{such that } \sum_{j=r_1}^r x_j \leq \sqrt{2rn} \text{ for all } r_1 \leq r \leq r_0$$

$$\text{and } x_r \geq 0 \text{ for all } r_1 \leq r \leq r_0.$$

Moreover, it can be shown that such a linear program attains its optimal solution at $x_{r_1} = \sqrt{2r_1n}$ and $x_r = \sqrt{2rn} - \sqrt{2(r-1)n}$ for all $r_1 + 1 \leq j \leq r_0$. Hence, by using the standard estimations $\sqrt{2rn} - \sqrt{2(r-1)n} \leq \sqrt{\frac{2n}{r}}$ and $\frac{1}{r^2} \leq \frac{1}{r(r-1)} = \frac{1}{r-1} - \frac{1}{r}$, we obtain

$$T^* \leq 10\sqrt{2}n^{3/2} \left(\frac{\sqrt{2r_1n}}{r_1^{3/2}} + \sum_{r=r_1+1}^{r_0} \frac{\sqrt{2rn} - \sqrt{2(r-1)n}}{r^{3/2}} \right) \leq$$

$$10\sqrt{2}n^{3/2} \left(\frac{\sqrt{2n}}{r_1} + \sum_{r=r_1+1}^{r_0} \frac{\sqrt{2n}}{r^2} \right) = 20n^2 \left(\frac{1}{r_1} + \sum_{r=r_1+1}^{r_0} \frac{1}{r^2} \right) \leq$$

$$20n^2 \left(\frac{1}{r_1} + \sum_{r=r_1+1}^{r_0} \left(\frac{1}{r-1} - \frac{1}{r} \right) \right) = 20n^2 \left(\frac{1}{r_1} + \frac{1}{r_1} - \frac{1}{r_0} \right) \leq \frac{40n^2}{r_1}. \quad \square$$

3.2 The second part: from $4\sqrt{5}\sqrt{n}$ runs to consensus

We prove a technical lemma that provides the estimation of the error in the computation of the expected runtime and the winning probabilities.

Lemma 4. *The expected value of the sum of $\left| \frac{s_\beta + m_\beta - s_\varrho - m_\varrho}{s_\beta + m_\beta + s_\varrho + m_\varrho} \right|$ during a random walk until the number of runs decreases is at most $1/2$. In particular, the expected sum of the above expression during the second part of the voting process is at most $2\sqrt{5}\sqrt{n}$.*

Proof. We use Lemma 2. In order to do that, the Markov chain is restricted to those states that have r blue runs, and extended by an absorbing state where we move exactly when in the original Markov chain the number of runs decreases. The penultimate states of this chain are exactly those states with r blue runs in our problem where there is a singleton vertex. Of course, the number $\underline{p}[t]$ (the probability of immediate absorption from t) is the probability that we lose runs, which is the probability that in the original Markov chain a singleton is pushed.

Our next goal is to calculate the probability of this event. We call the states with alternating red and blue vertices *special* states. Such states exist iff n is even, and then there are two of them. Now assume that the state is not special. Let a_1, \dots, a_h be an arc, surrounded by the non-singleton vertices b (a neighbor of a_1) and c (a neighbor of a_h). As we are not in the special states, the arc is not the full set of vertices, and b and c are indeed not singletons. We show that the probability that a singleton be pushed in this arc is $\frac{h+1}{d}$, where $d = s_\beta + s_\varrho + m_\beta + m_\varrho$ is the number of discordant vertices. If $h = 1$, then the vertex a_1 is indeed pushed with probability $\frac{2}{d}$: this happens exactly when b or c is chosen out of the d discordant vertices for pushing their opinion. If $h \geq 2$, then a vertex in the arc is pushed iff

- a vertex is chosen out of b, a_1, \dots, a_h, c for pushing, and
- if that vertex is a_1 or a_h , then the singleton neighbor is chosen.

Hence, the desired probability is $\frac{h+2}{d} - \frac{2}{d} \cdot \frac{1}{2} = \frac{h+1}{d}$. Adding the expression $\frac{h+1}{d}$ for all arcs, we obtain that the probability that a singleton vertex be pushed in the state t is $\underline{p}[t] = \frac{s_\beta + s_\varrho + \ell}{s_\beta + s_\varrho + m_\beta + m_\varrho}$, if t is not a special state.

Let the vector \underline{u} have entries $|\frac{s_\beta + m_\beta - s_\varrho - m_\varrho}{s_\beta + s_\varrho + m_\beta + m_\varrho}|$ for each transient state. Note that this is 0 for non-penultimate transient states, since red and blue runs alternate, so $s_\beta = s_\varrho = 0$ and $m_\beta = m_\varrho$. It is also 0 for the two special states.

So the expression $\frac{\underline{u}[t]}{\underline{p}[t]}$ (cf. Lemma 2) equals to $|\frac{s_\beta + m_\beta - s_\varrho - m_\varrho}{s_\beta + s_\varrho + \ell}|$ for all penultimate states. By Lemma 2 it suffices to show that $\frac{1}{2}$ is an upper bound for this expression. Observe that the expression does not decrease if we replace a red singleton by a pair of adjacent red vertices and likewise for a blue singleton. That is, we replace the two vertices by inserting two edges at the same positions in the cycle, obtaining a new cycle of length $n + 2$, and coloring the endpoints of the edge replacing the red and the blue vertex red and blue, respectively. Indeed, the numerator is not modified by this operation, and the denominator cannot increase, as $s_\beta + s_\varrho$ decreases by 2, and ℓ increases by at most 2. After a finite number of applications of this operation, we reach a state where all singleton vertices have the same color, say blue. In particular, there are no consecutive singleton vertices in the cycle. Thus $s_\beta = \ell$, $s_\varrho = 0$ and $m_\varrho = m_\beta + 2s_\beta$, so the expression simplifies to $|\frac{-s_\beta}{2s_\beta}| = \frac{1}{2}$. If all singletons disappear after a finite number of applications of the above operation, then the numerator of the expression is 0, thus it has been 0 when we started eliminating singletons, as well. \square

3.3 Estimations of the expected runtime on the n -cycle

As we suggested earlier, it seems impossible to compute the fundamental matrix of our Markov chain. However, the upper-left minor Q of the transition matrix is sparse, so Lemmas 1 and 2 can be applied effectively. The way we phrased the result in the introduction provides the right heuristics for the guesstimate vector. The expected runtime of the oblivious protocol is clearly $\beta\varrho$: it is simply the runtime of a drunkard walk with parameter $n = \beta + \varrho$ and initial state β (see [3] for more details). Computer simulations (in SAGE) suggested that the runtime of the three discordant protocols should be close to each other. The intuitive reason is that the transition matrix of the three protocols on the cycle graph coincide in almost all entries. Of course, such an observation can lead to very badly wrong conjectures in general, as the computation of the fundamental matrix involves the calculation of an inverse matrix, which is very sensitive to even small alterations of a few entries of the matrix. Hence, in order to turn this intuition into a precise proof, we use Lemma 1 with guesstimate vector \underline{x} whose entries are $\beta\varrho$ for each transient state.

Theorem 5. *Given any initial state on an n -cycle with r blue runs, β blue and ϱ red vertices. Let T be the expected number of steps for the discordant push voting to reach*

consensus. Then $|T - \beta\rho| \leq rn/2$, and $T \leq n^2/4 + 4\sqrt{5}n^{3/2}$. The worst expected runtime is asymptotically $n^2/4$, obtained when there is only one blue and one red run and $\beta = \lfloor n/2 \rfloor$.

Proof. Let \underline{x} be the column vector of length $2^n - 2$ with coordinates $\beta\rho$ for each transient state. Then the probability of the number of blue vertices to increase by 1, i.e., a blue vertex is pushing, is $\frac{s_\beta + m_\beta}{s_\beta + m_\beta + s_\rho + m_\rho}$, since there are $s_\beta + m_\beta$ discordant blue vertices among the $s_\beta + m_\beta + s_\rho + m_\rho$ vertices. Similarly, the probability of the number of red vertices to increase by 1 is $\frac{s_\rho + m_\rho}{s_\beta + m_\beta + s_\rho + m_\rho}$. In the former case, we arrive at a state where the value of \underline{x} is $(\beta + 1)(\rho - 1)$, and in the latter case it is $(\beta - 1)(\rho + 1)$. If we multiply the value of the vector \underline{x} with the corresponding transition probabilities, and add them up, i.e., we calculate $Q\underline{x}$ (cf. Lemma 1), we obtain:

$$\begin{aligned} & \frac{s_\beta + m_\beta}{s_\beta + m_\beta + s_\rho + m_\rho}(\beta + 1)(\rho - 1) + \frac{s_\rho + m_\rho}{s_\beta + m_\beta + s_\rho + m_\rho}(\beta - 1)(\rho + 1) = \\ & \beta\rho - 1 + \frac{(\rho - \beta)(s_\beta + m_\beta - s_\rho - m_\rho)}{s_\beta + m_\beta + s_\rho + m_\rho} \end{aligned}$$

for all non-penultimate transient states. Using the notation of Lemma 1 with $\underline{u} = \underline{1}$, the entry of the error vector $\underline{\varepsilon} = Q\underline{x} - \underline{x} + \underline{1}$ at the given state is $\frac{(\rho - \beta)(s_\beta + m_\beta - s_\rho - m_\rho)}{s_\beta + m_\beta + s_\rho + m_\rho}$, whose absolute value is at most $n \left| \frac{s_\beta + m_\beta - s_\rho - m_\rho}{s_\beta + m_\beta + s_\rho + m_\rho} \right|$. To obtain the error for penultimate transient states, we calculate it when there is exactly one red vertex, i.e., $s_\rho = 1, m_\rho = 0, s_\beta = 0, m_\beta = 2, \rho = 1, \beta = n - 1$. (The situation when there is exactly one blue vertex is analogous.) In that state $Q\underline{x} = \frac{s_\rho + m_\rho}{s_\beta + m_\beta + s_\rho + m_\rho}(\beta - 1)(\rho + 1) = \frac{2n-4}{3}$, thus $\underline{\varepsilon} = Q\underline{x} - \underline{x} + \underline{1}$ has entry $\frac{2n-4}{3} - (n - 1) + 1 = -\frac{n-2}{3}$. The absolute value of this number is at most $\frac{n}{3} = n \left| \frac{s_\beta + m_\beta - s_\rho - m_\rho}{s_\beta + m_\beta + s_\rho + m_\rho} \right|$ again. Hence, $n \left| \frac{s_\beta + m_\beta - s_\rho - m_\rho}{s_\beta + m_\beta + s_\rho + m_\rho} \right|$ estimates the absolute value of the error $\underline{\varepsilon}$ at all transient states from above, and the expected sum $N\underline{\varepsilon}$ of this expression during a random walk is at most $rn/2$ according to Lemma 4. Hence, by Lemma 1, we have $|N\underline{1} - \underline{x}| = |N\underline{\varepsilon}| \leq rn/2$, where $N\underline{1}$ is T at the initial state and \underline{x} is the guesstimate vector whose entry is $\beta\rho$ at the initial state. Thus $|T - \beta\rho| \leq rn/2$.

In order to prove the second estimation that is independent of the number of runs, we divide the process into two parts as before. By Proposition 3, the first part is expected to end in $T_1 \leq 2\sqrt{5}n^{3/2}$ steps. At the initial state of the second part, let r' be the number of blue runs and let β' and ρ' be the number of red and blue vertices, respectively. If T_2 denotes the expected runtime of the second part of the process, then $|T_2 - \beta'\rho'| \leq r'n/2 \leq 2\sqrt{5}n^{3/2}$ by the first assertion of this theorem, as $r' \leq 4\sqrt{5}\sqrt{n}$. As $\beta'\rho' \leq n^2/4$, we have $T = T_1 + T_2 \leq T_1 + |T_2 - \beta'\rho'| + \beta'\rho' \leq 2\sqrt{5}n^{3/2} + 2\sqrt{5}n^{3/2} + n^2/4 = n^2/4 + 4\sqrt{5}n^{3/2}$.

If there is one blue run and one red run and $\beta = \lfloor n/2 \rfloor$, then $|T - \beta\rho| \leq n$ by the first assertion of the theorem, and clearly $|\beta\rho - n^2/4| \leq 1$. Thus $T = n^2/4 + O(n)$ for this initial state, showing the final assertion in the theorem. \square

Remark 6. In a similar fashion, it can be shown that the expected time for the discordant pull voting to reach consensus on the cycle is also $\beta\rho + O(rn)$, and at most $n^2/4 + O(n^{3/2})$ from any initial state (and it is clearly exactly $\beta\rho$ for the discordant oblivious protocol).

This also makes the worst runtime asymptotically $n^2/4$ by using the same argument for these protocols. The lower estimate $n^2/4 + O(n)$ for the expected runtime from the initial state with two runs of (almost) equal size (for the discordant push, pull and oblivious protocols) was mentioned in [3]. It is clear that switching the cycle to a path makes very little difference in the calculation, and estimations of the same order of magnitude are obtained in case of the discordant push, pull and oblivious protocols on paths, too.

4 Winning probabilities on the cycle

It is enough to estimate the winning probability p of the color blue, the other color then wins with probability $1 - p$. Again, we know from standard theory [13, Theorem 3.3.7] that the matrix NR consists of the probabilities of reaching from transient state i the absorbing state j in the process. So we are only interested in the first column of this $(2^n - 2) \times 2$ matrix. Lemma 1 can be applied, as the problem is to estimate the vector $N\underline{u}$ where \underline{u} is the first column of R . Once again, we state the precise results for the discordant push protocol, but estimations of the same order of magnitude apply to the discordant pull and discordant oblivious protocols, as well.

Theorem 7. *Given any initial state on an n -cycle with r blue runs, β blue and ϱ red vertices. Let p be the probability that the blue consensus is reached with the discordant push protocol. Then $|p - \beta/n| \leq r/2n$.*

Proof. Let \underline{x} be the column vector of length $2^n - 2$ with coordinates $\frac{\beta}{n}$ for each transient state. If we multiply the value of the vector \underline{x} with the corresponding transition probabilities (cf. the proof of Theorem 5), and add them up, i.e., we calculate $Q\underline{x}$, we obtain:

$$\begin{aligned} & \frac{s_\beta + m_\beta}{s_\beta + m_\beta + s_\varrho + m_\varrho} \cdot \frac{\beta + 1}{n} + \frac{s_\varrho + m_\varrho}{s_\beta + m_\beta + s_\varrho + m_\varrho} \cdot \frac{\beta - 1}{n} = \\ & \frac{\beta}{n} + \frac{1}{n} \cdot \frac{s_\beta + m_\beta - s_\varrho - m_\varrho}{s_\beta + m_\beta + s_\varrho + m_\varrho} \end{aligned}$$

for all non-penultimate transient states. We use the notation of Lemma 1 with \underline{u} being the all 0 vector, except for the entries corresponding to the states with exactly one red vertex which are all $\frac{2}{3}$. In particular, for non-penultimate states the error, i.e., the corresponding entry of $\underline{\varepsilon} = Q\underline{x} - \underline{x} + \underline{u}$ is $\frac{1}{n} \cdot \frac{s_\beta + m_\beta - s_\varrho - m_\varrho}{s_\beta + m_\beta + s_\varrho + m_\varrho}$.

For penultimate states with exactly one red vertex, the entry of the vector $Q\underline{x}$ is $\frac{s_\varrho + m_\varrho}{s_\beta + m_\beta + s_\varrho + m_\varrho} \cdot \frac{\beta - 1}{n} = \frac{n - 2}{3n}$. Thus the error is $\frac{n - 2}{3n} - \frac{n - 1}{n} + \frac{2}{3} = \frac{1}{3n}$ which is precisely $\frac{1}{n} \cdot \frac{s_\beta + m_\beta - s_\varrho - m_\varrho}{s_\beta + m_\beta + s_\varrho + m_\varrho}$.

Finally, for penultimate states with exactly one blue vertex, the entry of the vector $Q\underline{x}$ is $\frac{s_\beta + m_\beta}{s_\beta + m_\beta + s_\varrho + m_\varrho} \cdot \frac{\beta + 1}{n} = \frac{2}{3n}$. Thus the error is $\frac{2}{3n} - \frac{1}{n} + 0 = -\frac{1}{3n} = \frac{1}{n} \cdot \frac{s_\beta + m_\beta - s_\varrho - m_\varrho}{s_\beta + m_\beta + s_\varrho + m_\varrho}$.

Hence, the vector with entries $\frac{1}{n} \cdot \frac{s_\beta + m_\beta - s_\varrho - m_\varrho}{s_\beta + m_\beta + s_\varrho + m_\varrho}$ is exactly the error vector $\underline{\varepsilon}$. As the number of runs decreases r times before a consensus is reached, the assertion follows by Lemma 1 and Lemma 4 as in the proof of Theorem 5. \square

According to [5, Theorem 8] the linear push, pull and oblivious protocols on cycle graphs have the property of proportionate agreement: the probability that blue wins is β/n , see also [8]. Hence, Theorem 7 shows that if $r = o(n)$, then the probability to reach the blue consensus with the linear push voting and the discordant push voting protocols have the same asymptotics as $n \rightarrow \infty$. The final result of this paper (Theorem 10) shows that the condition $r = o(n)$ cannot be omitted from this assertion. To prove a gap between the probabilities arising from the linear and discordant push protocols, we recall a Chernoff-Hoeffding type theorem from [11]. The function D stands for the Kullback-Leibler distance, namely $D(q||p) = q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p}$ for all $q, p \in]0, 1[$, and $\exp(x)$ is the exponential function e^x .

Theorem 8. *Let $N \in \mathbb{N}$ and let X_1, \dots, X_N be random variables with range in $\{0, 1\}$. Assume that there is a $\delta \in]0, 1[$ such that for all $1 \leq k \leq N$ and k -element subset $S \subseteq \{1, 2, \dots, N\}$ we have $\mathbb{P}\left(\bigwedge_{i \in S} (X_i = 1)\right) \leq \delta^k$. Then $\mathbb{P}\left(\sum_{i=1}^N X_i \geq \gamma N\right) \leq \exp(-ND(\gamma||\delta))$ for all $\gamma \in]\delta, 1[$.*

If $3 \mid n$ then there is an assignment of colors red and blue to the vertices of the n -cycle such that every blue run is a singleton and every red run is a pair of vertices. If $3 \nmid n$, then by allowing one of the red runs to consist of three or four vertices, we can construct a state with $\lfloor n/3 \rfloor$ discordant blue and $2\lfloor n/3 \rfloor$ discordant red vertices.

Lemma 9. *Let $n, m, i \in \mathbb{Z}$ such that $n \geq 3$, $m = \lfloor n/3 \rfloor$ and $i < m/2$. Assume that the discordant push protocol starts from an initial state on the n -cycle with m runs of each color, such that all the blue runs are singletons and all the red runs have length at least two. Then after i steps the ratio of discordant blue vertices and discordant vertices is at most $\frac{1}{3} + \frac{i}{3m}$.*

Proof. In any given state, let d_b and d_r denote the number of discordant blue and discordant red vertices, and let $d = d_b + d_r$. Then $d_b/d = d_b/(d_b + d_r) = 1/(1 + d_r/d_b)$ is maximal wherever d_r/d_b is minimal. In any asynchronous protocol on a cycle graph (i.e., when in one step only one vertex alters its opinion), the following analysis applies.

Case 1: The number of runs does not change in a step. If color c is spreading, then the number of discordant vertices of color c changes by 0 or +1, and the number of discordant vertices of the other color changes by 0 or -1. Note that the increment of the expression $d_r - d_b$ is at least -2.

Case 2: The number of runs decreases in a step. If color c is spreading, then the number of discordant vertices of color c changes by -2, -1 or 0, and the number of discordant vertices of the other color changes by -1. In particular, the increment of the expression $d_r - d_b$ is at least -1.

Thus in the first i steps, there are always more red discordant vertices than blue, as $i < m/2$ and the difference $d_r - d_b$ in the initial state is m . Moreover, both d_r and d_b are positive numbers throughout the process, as the increments of d_r and d_b are both at least -2 in every step, and in the initial state we have $d_b = m$ and $d_r = 2m$.

From now on, we are looking for the minimum of the expression d_r/d_b after i steps as described in Case 1 and Case 2. That is, we assume that at any point in time any manipulation of the values d_r and d_b is allowed as long as it is consistent with Case 1 or Case 2, whether or not it is combinatorially feasible. We claim that the minimum is attained when in all steps we apply Case 1 such that c is the color blue and the increment of d_b and d_r are $+1$ and -1 , respectively. We are going to refer to this particular choice of parameters in a step as the favored option.

If Case 1 is applied in a step, then to attain the minimum of d_r/d_b , we must choose the favored option: otherwise, switching to the favored option in that step would yield values d'_r and d'_b after i steps such that $d'_r \leq d_r$ and $d'_b \geq d_b$, and at least one of the inequalities would be strict.

Similarly, if Case 2 is applied in a step with c as the color blue, then the increment of d_r has to be -1 and that of d_b has to be 0 to attain the minimum of d_r/d_b . We compare the outcome of this process to the one where in that step we apply the favored option, yielding $d'_r = d_r$ and $d'_b = d_b + 1$. Then the original outcome d_r/d_b is modified to $d_r/(d_b + 1)$, which is clearly smaller.

Finally, if c is chosen as the color red in Case 2, then the increment of d_r has to be -2 and that of d_b has to be -1 , as this choice dominates all other possibilities. We compare the outcome of this process to the one where in that step the favored option is applied yielding $d'_r = d_r + 1$ and $d'_b = d_b + 2$. Then the original outcome d_r/d_b is modified to $(d_r + 1)/(d_b + 2)$, and the calculation

$$\frac{d_r}{d_b} > \frac{d_r + 1}{d_b + 2} \Leftrightarrow d_r d_b + 2d_r > d_b d_r + d_b \Leftrightarrow 2d_r > d_b$$

and the fact that $d_r > d_b > 0$ show that it is worth switching to the favored option, proving the claim.

Hence, the highest possible ratio d_b/d is attained after applying the favored option i times, yielding $d_b = m + i$, $d_r = 2m - i$, $d = 3m$, and consequently, $d_b/d = \frac{1}{3} + \frac{i}{3m}$. \square

Theorem 10. *For any $n \geq 5000$ there exists a state on the n -cycle with β blue vertices and probability p of reaching the blue consensus with the discordant push protocol such that $|p - \beta/n| > 0.005$.*

Proof. Let ε be the largest absolute error of the estimate β/n over all states on the n -cycle. Let $m = \lfloor n/3 \rfloor$, and pick an initial state for the push protocol with m discordant blue and $2m$ discordant red vertices as in Lemma 9. Let $N = \lfloor m/4 \rfloor$, $\delta = 5/12$, $\gamma = 51/120$, and let X_i be the random variable that is 1 if a blue vertex is pushing in the i -th step and 0 otherwise. In particular, the increment of blue vertices in the i -th step for $i \leq N$ is $2X_i - 1$.

According to Lemma 9, in any of the first N steps of the process, the probability that a blue vertex is pushing is at most $\frac{1}{3} + \frac{N}{3m} \leq \frac{1}{3} + \frac{1}{12} = \delta$, independently of the outcome of the earlier steps. Thus the conditions of Theorem 8 apply, and then $\mathbb{P}\left(\sum_{i=1}^N X_i \geq \gamma N\right) \leq \exp(-ND(\gamma\|\delta))$. As $n \geq 5000$, we have $N \geq 416$, and then

$\mathbb{P}\left(\sum_{i=1}^N X_i \geq \gamma N\right) \leq \exp(-ND(\gamma\|\delta)) \leq 0.001$. Moreover, if $\sum_{i=1}^N X_i < \gamma N$, then the number of blue vertices after N steps, which is $m - N + 2 \sum_{i=1}^N X_i$, is at most $m - N + 2\gamma N = m - (1 - 2\gamma)N$. Hence, by the definition of ε and the law of total probability, we have $\mathbb{P}(\text{blue wins}) \geq \frac{m}{n} - \varepsilon$ and

$$\begin{aligned} \mathbb{P}(\text{blue wins}) &= \left(1 - \mathbb{P}\left(\sum_{i=1}^N X_i \geq \gamma N\right)\right) \cdot \mathbb{P}\left(\text{blue wins} \mid \sum_{i=1}^N X_i < \gamma N\right) + \\ &\quad \mathbb{P}\left(\sum_{i=1}^N X_i \geq \gamma N\right) \cdot \mathbb{P}\left(\text{blue wins} \mid \sum_{i=1}^N X_i \geq \gamma N\right) \leq \\ \mathbb{P}\left(\text{blue wins} \mid \sum_{i=1}^N X_i < \gamma N\right) + \mathbb{P}\left(\sum_{i=1}^N X_i \geq \gamma N\right) &\leq \left(\frac{m - (1 - 2\gamma)N}{n} + \varepsilon\right) + 0.001. \end{aligned}$$

By comparing the two estimations for $\mathbb{P}(\text{blue wins})$, we obtain

$$\frac{m}{n} - \varepsilon \leq \frac{m - (1 - 2\gamma)N}{n} + \varepsilon + 0.001$$

and then

$$\frac{1}{2} \cdot \left(\frac{(1 - 2\gamma)N}{n} - 0.001\right) \leq \varepsilon.$$

Using the trivial estimations $n \leq 12N + 11$ and $N \geq 416$, the left hand side is at least $\frac{1}{2} \cdot \left(\frac{1-2\gamma}{12+11/N} - 0.001\right) \geq \frac{1}{2} \cdot \left(\frac{1-51/60}{12+11/416} - 0.001\right) > 0.005$. \square

Due to Proposition 3 and Theorem 7 it is however possible to estimate the desired probability up to an error term $O(1/\sqrt{n})$ by running the process for $O(n^{3/2})$ steps. This is a good trade-off as the expected time to reach a consensus from the worst initial case is $n^2/4 + O(n^{3/2})$, and it is quadratic in general (cf. Theorem 5). In fact, using this observation, it is possible to write a relatively fast program that runs the experiment on the cycle with 5000 vertices 5000 times from the initial state described before Lemma 9. The result suggests that in that very symmetrical state, the estimate $1/3$ for the probability that the blue consensus be reached is highly accurate. If this empirical result is correct, then there must be a more complicated formula than the one suggested by Theorem 7. It should take into consideration the position of blue vertices around the cycle as well as their number, and coincidentally, this formula should assign a value close to $1/3$ to the above mentioned initial state.

5 Further results and future work

It is also possible to obtain asymptotically sharp estimates for the corresponding problems in the star graph with n vertices. This is a typical network when one server is connected

to several clients. It was already pointed out in [3] that, quite counter-intuitively, the discordant pull protocol is faster than the discordant push protocol on such graphs if n is large enough. The author of the present paper together with coauthors were able to refine this result and obtain asymptotically sharp estimations for both expected runtimes. It seems plausible to find estimates to the higher moments of the runtime of these protocols with similar techniques, as well.

Acknowledgements

The author is indebted to the thorough referee for finding some inaccuracies in the first draft, and whose comments also improved the presentation of the paper.

References

- [1] ACAN, H., COLLEVECCHIO, A., MEHRABIAN, A., AND WORMALD, N. On the push&pull protocol for rumour spreading. In *Proceedings of the ACM Symposium on Principles of Distributed Computing* (2015), pp. 405–412.
- [2] BASU, R., AND SLY, A. Evolving voter model on dense random graphs. *Annals of Applied Probability* 27, 2 (2017), 1235–1288.
- [3] COOPER, C., DYER, M., FRIEZE, A., AND RIVERA, N. Discordant voting processes on finite graphs. In *43rd International Colloquium on Automata, Languages, and Programming* (ICALP 2016), pp. 2033–2045.
- [4] COOPER, C., ELSASSER, R., ONO, H., AND RADZIK, T. Coalescing random walks and voting on connected graphs. *SIAM Journal on Discrete Mathematics* 27, 4 (2013), 1748–1758.
- [5] COOPER, C., AND RIVERA, N. The linear voting model. In *43rd International Colloquium on Automata, Languages, and Programming* (ICALP 2016), pp. 2021–2032.
- [6] DONNELLY, P., AND WELSH, D. Finite particle systems and infection models. *Mathematical Proceedings of the Cambridge Philosophical Society* 94, 1 (1983), 167–182.
- [7] DURRETT, R., GLEESON, J. P., LLOYD, A. L., MUCHA, P. J., SHI, F., AND SIVAKOFF, D. Graph fission in an evolving voter model. *Proceedings of the National Academy of Sciences* 109, 10 (2012), 3682–3687.
- [8] HASSIN, Y., AND PELEG, D. Distributed probabilistic polling and applications to proportionate agreement. *Information and Computation* 171, 2 (2001), 248–268.
- [9] HEDETNIEMI, S. M., HEDETNIEMI, S. T., AND LIESTMAN, A. L. A survey of gossiping and broadcasting in communication networks. *Networks* 18, 4 (1988), 319–349.
- [10] HOLME, P., AND NEWMAN, M. E. J. Nonequilibrium phase transition in the coevolution of networks and opinions. *Physical Review E* 74, 5 (2006), 5 pp.

- [11] IMPAGLIAZZO, R., AND KABANETS, V. Constructive proofs of concentration bounds. In *Lecture Notes in Computer Science* (2010), vol. 6302, pp. 617–631.
- [12] KARP, R., SCHINDELHAUER, C., SHENKE, S., AND VOCKING, B. Randomized rumor spreading. In *FOCS '00 Proceedings of the 41st Annual Symposium on Foundations of Computer Science* (2000), pp. 565–574.
- [13] KEMENY, J. G., AND SNELL, J. L. *Finite Markov chains*. Springer-Verlag, 1976.
- [14] LO CIGNO, R., RUSSO, A., AND CARRA, D. On some fundamental properties of p2p push/pull protocols. In *2008 Second International Conference on Communications and Electronics* (June 2008), pp. 67–73.
- [15] LOCHER, T., MEIER, R., SCHMID, S., AND WATTENHOFER, R. Push-to-pull peer-to-peer live streaming. In *Proceedings of the International Symposium on Distributed Computing* (2007), pp. 388–402.
- [16] NAKATA, T., IMAHAYASHI, H., AND YAMASHITA, M. *Probabilistic local majority voting for the agreement problem on finite graphs*. Springer, 1999.
- [17] OLIVEIRA, R. On the coalescence time of reversible random walks. *Transactions of the American Mathematical Society* 364, 4 (2012), 2109–2128.
- [18] OLIVEIRA, R. Mean field conditions for coalescing random walks. *The Annals of Probability* 41, 5 (2013), 3420–3461.
- [19] TRAN, D. A., HUA, K. A., AND DO, T. T. A peer-to-peer architecture for media streaming. *IEEE Journal on Selected Areas in Communications* 22, 1 (2004), 121–133.