Acta Scientiarum Mathematicarum
https://doi.org/10.1007/s44146-024-00125-x

Acta Scientiarum Mathematicarum

Voting protocols on the star graph

Kamilla Kátai-Urbán, András Pongrácz and Csaba Szabó

Abstract. Let $G = (V, E)$ be a finite graph together with an initial assignment $V \to \{0, 1\}$ that represents the opinion of each vertex. Then discordant push voting is a discrete, non-deterministic protocol that alters the opinion of one vertex at a time until a consensus is reached. More precisely, at each round a discordant vertex u (i.e., one that has a neighbor with a different opinion) is chosen uniformly at random, and then we choose a neighbor v with different vote uniformly at random, and force v to change its opinion to that of u. In case of the discordant pull protocol we simply choose a discordant vertex uniformly at random and change its opinion. In this paper, we give asymptotically sharp estimations for the worst expected runtime of the discordant push and pull protocols on the star graph.

Mathematics Subject Classification. Primary 91A22, Secondary 60J10.

Keywords. Discordant voting, Push, Pull, Star graph.

1. Introduction

Models of voting in finite graphs have been studied intensively for decades, see [2,6,10–12]. 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. Each step, the participants can affect the vote of their neighbors according to the rules given.

We note that many alternative definitions were investigated in the literature. Continuous-time voting processes were studied in [6,9]. In [1,8] the

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, the ÚNKP-18-4 and the ÚNKP-19-4 New National Excellence Program of the Ministry of Human Capacities, and by the NRDI grant KKP 138270. On behalf of all authors, the corresponding author states that there is no Conflict of interest.

Published online: 08 April 2024 Birkhäuser

graph evolves together with the opinions of the vertices. Connections of voting processes and coalescing random walks were investigated in [9,13], and for other recent applications see [4,14].

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. 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: 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: each round a vertex u is chosen uniformly at random, and a randomly chosen neighbor of u adopts the opinion of u.
- Pull protocol: each round a vertex u is chosen uniformly at random, and u adopts the opinion of a randomly chosen neighbor of u.

Linear voting models are somewhat impractical, as it is typical that many idle rounds occur in the process. 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 in a round have the more popular opinion. If this is the case, then none of the opinions are modified in that round. 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).

The goal of every voting scheme that we study now is to reach consensus, that is, a state where all participants have the same opinion. The topic of the present paper is the expected time T to reach consensus with the discordant push, pull and oblivious processes on the star graph with n vertices. It was proven in [3] that the discordant push process has an expected runtime between $C_1 n^2 \log n$ and $C_2 n^2 \log n$ at worst, with some positive constants C_1, C_2. We improve these bounds, showing that the discordant push protocol reaches consensus on the star graph with n vertices in $T_{\text{push}} = \frac{1}{8}n^2 \log n + O(n^2)$ expected time. The pull protocol is the fastest out of the three above defined processes on the star graph. Its expected runtime is $T_{\text{pull}} = \frac{1}{6}n^2 + \frac{1}{6}n \log n + O(n)$. It was already discussed in [3] that the oblivious protocol has expected runtime $T_{\text{oblivious}} = \frac{1}{4}n^2 + O(n)$. These results are somewhat counter-intuitive. As mentioned in [3], for a typical graph the push protocol should be the fastest out of the three, and the pull voting should be the slowest. The analogous problems for cycle graphs were solved in [15].

2. Preliminaries

2.1. General notation

Given an absorbing Markov chain P. As usual, we denote by Q the upper left minor of the canonical form of $P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$. So Q is the transition matrix restricted to the transient states. Following standard notation, $N = (I - Q)^{-1}$ denotes the fundamental matrix of the Markov chain. We denote by $\underline{1}$ the column vector all of whose entries are 1, and whose length equals to the number of transient states. It is well-known that the expected times to absorption from each transient state as initial state are the coordinates of the vector $N\underline{1}$. For an introduction to Markov chains see [7]. Throughout the paper log denotes the natural logarithm and w is the central vertex of the given star graph with n vertices.

2.2. Push voting on the star graph

In order to make the problem more transparent, we define a Markov chain that perfectly describes the voting process. A number is assigned to each possible list of opinions of the vertices that might occur during the voting process, namely the number of neighbors of the center w in the star that have the opposite opinion as w. Hence, all possible lists of opinions such that w has i neighbors that disagree with w are identified with one state, and we simply refer to it as state i. This way the 2^n possible lists of opinions are replaced by only n states, as $0 \leq i \leq n-1$, and instead of two absorbing states we only have the one, namely $i = 0$. If $i > 0$, then the process can evolve to two possible states. If we pick a vertex out of the i discordant neighbors of the center, then the vote of the center is altered, hence we reach the state $n-1-i$. If we pick the center, then one of its discordant neighbors is pushed, thus we end up in state $i-1$. So the probability of transition from i to $n-1-i$ is $\frac{i}{i+1}$, and the probability of transition from i to $i-1$ is $\frac{1}{i+1}$. We are interested in the expected time to reach consensus from the worst case. Note that the worst case is the one with an equal number m of zeros and ones (or as close to equal as possible). This can be extracted from the calculation below, but it is also quite intuitive. The center changes color with high probability, so the expected runtime from i and $n-1-i$ has to be close, and if we start the process from m, then it will reach i or $n-i$ for all i.

Note that it is impossible to reach state $n-1$ from m, or in fact from any state different from $n-1$: indeed, in state $n-1$ all the edges are discordant, and none of the discordant voting protocols can move into such a state, as in each step two vertices are forced to agree. So we may omit state $n-1$, and only consider the transient states $1, \ldots, n-2$ and the unique absorbing state 0.

The $(n-2) \times (n-2)$ matrix $I - Q$ derived from the transition matrix looks as follows (we do the illustration and the calculation for odd n, the case of even n is very similar, and of course, the same estimation is obtained in the end); $k = (n+1)/2$:

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} \\
-\frac{1}{3} & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & -\frac{2}{3} & 0 \\
0 & -\frac{1}{4} & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & -\frac{3}{4} & 0 & 0 \\
0 & 0 & -\frac{1}{5} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & -\frac{4}{5} & 0 & 0 & 0 \\
\vdots & \vdots & & \ddots & \ddots & & & & & \iddots & & & & \vdots \\
 & \cdots & 0 & 0 & -\frac{1}{k-2} & 1 & 0 & 0 & 0 & -\frac{k-3}{k-2} & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & -\frac{1}{k-1} & 1 & 0 & -\frac{k-2}{k-1} & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & 0 & -\frac{1}{k} & \frac{1}{k} & 0 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & 0 & -\frac{k}{k+1} & -\frac{1}{k+1} & 1 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & -\frac{k+1}{k+2} & 0 & 0 & -\frac{1}{k+2} & 1 & 0 & 0 & \cdots & \\
 & & \vdots & & \iddots & & \vdots & & & \ddots & \ddots & \vdots & & \\
0 & 0 & 0 & -\frac{n-5}{n-4} & 0 & \cdots & & \cdots & 0 & -\frac{1}{n-4} & 1 & 0 & 0 & 0 \\
0 & 0 & -\frac{n-4}{n-3} & 0 & 0 & \cdots & & \cdots & 0 & 0 & -\frac{1}{n-3} & 1 & 0 & 0 \\
0 & -\frac{n-3}{n-2} & 0 & 0 & 0 & \cdots & & \cdots & 0 & 0 & 0 & -\frac{1}{n-2} & 1 & 0 \\
-\frac{n-2}{n-1} & 0 & 0 & 0 & 0 & \cdots & & \cdots & 0 & 0 & 0 & 0 & -\frac{1}{n-1} & 1
\end{pmatrix}$$

This matrix is denoted by A. We need to solve the system of linear equations $A\underline{z} = \underline{1}$ in order to compute $N\underline{1}$, see Sect. 2.1. The $(k-1)$-th element of that solution vector corresponds to the worst case. We shall apply elementary steps of Gaussian elimination on the rows of A to obtain the row vector that is all zero except for the $(k-1)$-th element, which is 1. The following program shows the steps that need to be applied. We use standard notation that is used in several programming languages (e.g., Python); note that $A[i]$ is the i-th row of the matrix, where indexation of rows starts by 1 (unlike in Python, where the first index of an array is 0), and the symbols $-=, += $ and $*=$ mean that the left hand side is decreased, increased and multiplied by the right hand side, respectively.

The first step is a special one: $A[k-1] *= k$.

Then we apply the following triples of steps for $i = 0, 1, \ldots, k-3$:

$$\begin{aligned}
A[k+i] &-= \frac{k+i}{k+i+1} A[k-i-1] \\
A[k-i-2] &+= (A[k-i-1] + A[k+i]) \\
A[k-i-2] &*= (k-i-1)
\end{aligned}$$

Initially, there is 1 on the right-hand side of every equation. We denote by $\underline{x}$ the vector obtained after applying all these steps to the right-hand side. The combined result of the first two steps in the triple of steps above is that in the $(k-i-2)$-th line we obtain $1 + (x_{k-i-1} + (1 - \frac{k+i}{k+i+1} \cdot x_{k-i-1})) = 2 + \frac{1}{k+i+1} \cdot x_{k-i-1}$. After applying the third step, we obtain the value $x_{k-i-2} = 2(k-i-1) + \frac{k-i-1}{k+i+1} \cdot x_{k-i-1}$.

Putting $a_i = x_{k-1-i}$, the sequence $(a_i)_{i=0,\ldots,(k-1)}$ is uniquely determined by the base condition $a_0 = k$ and the following recursive definition: $a_i = a_{i-1} \cdot \frac{k-i}{k+i} + 2k - 2i$. We define $b_i = \frac{k^2}{i+1} - (i+1)$ and $\varepsilon_i = a_i - b_i$ for $i = 0, \ldots, k-1$.

Lemma 2.1. *Let $(c_i)_{i=0,\ldots,(k-1)}$ be a sequence satisfying the same recursive definition as a_i, that is, $c_i = c_{i-1} \cdot \frac{k-i}{k+i} + 2k - 2i$. Then $c_i > c_{i-1}$ iff $c_{i-1} < b_{i-1}$, $c_i = c_{i-1}$ iff $c_{i-1} = b_{i-1}$, and $c_i < c_{i-1}$ iff $c_{i-1} > b_{i-1}$.*

Proof. Obvious from the definition of the sequences. □

Lemma 2.2.

1. *For all* $i \leq \frac{1}{\sqrt{2}} \cdot \sqrt{k}$ *we have* $a_i \geq ik$.
2. *For all* i *we have* $a_i \leq (2i+1)k$.
3. *If* $k \geq 100$, *then there is an* $\ell \leq 2 \cdot \sqrt{k}$ *such that* $a_\ell \geq b_\ell$, *and the smallest such index* ℓ *is at least* $\frac{1}{2} \cdot \sqrt{k}$.

Proof. The first two items are shown by induction. Both statements hold for the initial value $i = 0$. If $i \leq \frac{1}{\sqrt{2}} \cdot \sqrt{k}$, then by the induction hypothesis we have $a_i = \frac{k-i}{k+i} \cdot a_{i-1} + 2k - 2i \geq (1 - \frac{2i}{k})a_{i-1} + 2k - 2i \geq (1 - \frac{2i}{k})(i-1)k + 2k - 2i = ik - 2i^2 + k \geq ik$.

The second item follows from

$$\begin{aligned} a_i &\leq \left(1 - \frac{i}{k}\right) a_{i-1} + 2k - 2i \leq \left(1 - \frac{i}{k}\right)(2i-1)k + 2k - 2i \\ &= (2i+1)k - 2i^2 - i \leq (2i+1)k. \end{aligned}$$

The third item is shown indirectly. Assume that $a_i < b_i$ for all $i \leq 2 \cdot \sqrt{k}$. Then by Lemma 2.1 the series (a_i) is strictly monotone increasing at any index less than $2 \cdot \sqrt{k}$. By item 1 of the current lemma, this means that $a_i \geq \left(\frac{1}{\sqrt{2}} \cdot \sqrt{k} - 1\right) k$ for all $\frac{1}{\sqrt{2}} \cdot \sqrt{k} \leq i \leq 2 \cdot \sqrt{k}$. On the other hand, if ℓ is the floor of $2 \cdot \sqrt{k}$, then $\ell + 1 \geq 2 \cdot \sqrt{k}$, so $b_\ell = \frac{k^2}{\ell+1} - (\ell+1) \leq \frac{k^2}{2 \cdot \sqrt{k}} - (2 \cdot \sqrt{k}) < \frac{1}{2} \cdot k^{3/2} < \left(\frac{1}{\sqrt{2}} \cdot \sqrt{k} - 1\right) k \leq a_\ell$ if $k \geq 100$, a contradiction. The smallest index ℓ such that $a_\ell \geq b_\ell$ cannot be smaller then $\frac{1}{2} \cdot \sqrt{k}$: the series (a_i) is strictly monotone increasing in that region, and if $\ell \leq \frac{1}{2} \cdot \sqrt{k} + 1$, then by item 2 we have $a_\ell \leq (2\ell+1)k = (\sqrt{k}+3)k = k^{3/2} + 3k$, whereas the series (b_i) is strictly monotone decreasing and $b_\ell \geq \frac{k^2}{\ell+1} - (\ell+1) \geq \frac{k^2}{\frac{1}{2} \cdot \sqrt{k} + 2} - (\frac{1}{2} \cdot \sqrt{k} + 2)$, which for $k \geq 100$ is bigger than $k^{3/2} + 3k$. □

Lemma 2.3. *Let* $k \geq 100$, *and let* ℓ *be the smallest index such that* $a_\ell \geq b_\ell$ *(cf. Lemma 2.2).*

1. *For all* $i \leq \ell$ *we have* $0 \leq a_i \leq 5k^{3/2}$, *and* $0 \leq \sum_{i=1}^{\ell} a_i \leq 10k^2$.
2. $0 \leq \varepsilon_\ell < 7k$
3. *For all* $i \geq \ell + 1$ *we have* $0 \leq \varepsilon_i = \frac{k-i}{k+i} \cdot \varepsilon_{i-1} + \frac{k^2}{i(i+1)} + 1$, *and moreover* $0 \leq \sum_{i=\ell}^{k-2} \varepsilon_i < 6k^2$.
4. *The sum of all the* a_i *is* $\sum_{i=0}^{k-2} a_i = \frac{1}{2} \cdot k^2 \log k + O(k^2)$.

Proof. By Lemma 2.2 we have $\frac{1}{2} \cdot \sqrt{k} \leq \ell \leq 2 \cdot \sqrt{k}$. The first item follows from item 2 of Lemma 2.2 and from the fact that the series (a_i) is monotone increasing in the first ℓ indices by Lemma 2.1.

For the second item, we use the inequalities

- $a_{\ell-1} < b_{\ell-1}$, by the minimality of ℓ; equivalently, $a_{\ell-1} - b_{\ell-1} < 0$,
- $a_\ell - a_{\ell-1} \le 2k$ by the recursive rule that defines the series (a_i), and
- $b_{\ell-1} - b_\ell = (\frac{k^2}{\ell} - \ell) - (\frac{k^2}{\ell+1} - \ell - 1) < \frac{k^2}{\ell^2} + 1 \le 5k$, as $\frac{1}{2} \cdot \sqrt{k} \le \ell$.

Adding up these three inequalities yields the second item of the lemma. The third item is shown by first observing that

$$\begin{aligned} a_i &= (b_{i-1} + \varepsilon_{i-1}) \cdot \frac{k-i}{k+i} + 2k - 2i = \left(\frac{k^2 - i^2}{i} + \varepsilon_{i-1}\right) \cdot \frac{k-i}{k+i} + 2k - 2i \\ &= \frac{(k^2 - 2ki + i^2) + 2ki - 2i^2}{i} + \frac{k-i}{k+i} \cdot \varepsilon_{i-1} = \frac{k^2}{i} - i + \frac{k-i}{k+i} \cdot \varepsilon_{i-1} \\ &= \frac{k^2}{i+1} - (i+1) + \frac{k-i}{k+i} \cdot \varepsilon_{i-1} + \frac{k^2}{i(i+1)} + 1 = b_i + \frac{k-i}{k+i} \cdot \varepsilon_{i-1} + \frac{k^2}{i(i+1)} + 1. \end{aligned}$$

This calculation verifies $\varepsilon_i = \frac{k-i}{k+i} \cdot \varepsilon_{i-1} + \frac{k^2}{i(i+1)} + 1$, and in particular all the ε_i are non-negative for $i \ge \ell$. For the upper estimation of the sum $\sum_{i=\ell}^{k-2} \varepsilon_i$ we observe that $\frac{k-i}{k+i} \le 1 - \frac{1}{2\sqrt{k}}$ for all $i \ge \ell$, as $\frac{1}{2} \cdot \sqrt{k} \le \ell \le i$. Hence, for any tuple $\ell \le i_1 < i_2 < \cdots < i_t$ we have $\sum_{u=1}^{t} \prod_{v=1}^{u} \frac{k-i_v}{k+i_v} \le \sum_{m=0}^{\infty} (1 - \frac{1}{2\sqrt{k}})^m = 2\sqrt{k}$. Thus

$$\begin{aligned} \sum_{i=\ell}^{k-2} \varepsilon_i &= \varepsilon_\ell + \left(\frac{k-\ell}{k+\ell}\varepsilon_\ell + \frac{k^2}{\ell(\ell+1)} + 1\right) \\ &\quad + \left(\frac{k-(\ell+1)}{k+(\ell+1)} \cdot \left(\frac{k-\ell}{k+\ell}\varepsilon_\ell + \frac{k^2}{\ell(\ell+1)} + 1\right) + \frac{k^2}{(\ell+1)(\ell+2)} + 1\right) + \cdots \end{aligned}$$

and by expanding all the parentheses in this expression, it is clear that the sum of all coefficients of ε_ℓ is at most $2\sqrt{k}$, and so is the sum of all coefficients of any of the $\frac{k^2}{i(i+1)}$. We may simply estimate from above the coefficient of each occurrence of 1 by 1: there are less then $k^2/2$ occurrences. This way we obtain the upper estimation

$$\begin{aligned} \sum_{i=\ell}^{k-2} \varepsilon_i &\le 2\sqrt{k}\varepsilon_\ell + 2\sqrt{k} \cdot \sum_{i=\ell}^{k-2} \frac{k^2}{i(i+1)} + \frac{k^2}{2} \\ &\le 2\sqrt{k} \cdot 7k + 2k^{5/2} \sum_{i=\ell}^{k-2} \left(\frac{1}{i} - \frac{1}{i+1}\right) + \frac{k^2}{2} \le 14k^{3/2} + 2k^{5/2} \cdot \frac{1}{\ell} + \frac{k^2}{2} \\ &\le 14k^{3/2} + 4k^2 + \frac{k^2}{2} \le 6k^2 \end{aligned}$$

as $k \ge 100$.

We now prove the fourth item. By item 1 we have $\sum_{i=0}^{k-2} a_i = \sum_{i=0}^{\ell-1} a_i + \sum_{i=\ell}^{k-2} a_i = O(k^2) + \sum_{i=\ell}^{k-2} a_i$. So the sum is

$$\begin{aligned}\sum_{i=0}^{k-2} a_i &= O(k^2) + \sum_{i=\ell}^{k-2} b_i + \sum_{i=\ell}^{k-2} \varepsilon_i = O(k^2) + \sum_{i=\ell}^{k-2} b_i \\ &= O(k^2) + \sum_{i=\ell}^{k-2} \frac{k^2}{i+1} - \sum_{i=\ell}^{k-2} (i+1) = O(k^2) + k^2 \cdot \sum_{i=\ell}^{k-2} \frac{1}{i+1} \\ &= O(k^2) + k^2 \cdot \left(\sum_{i=1}^{k-1} \frac{1}{i} - \sum_{i=1}^{\ell} \frac{1}{i} \right) = O(k^2) + k^2 \cdot (\log k - \log \ell + O(1)) \\ &= O(k^2) + k^2 \cdot (\log k - \log \ell).\end{aligned}$$

As $\frac{1}{2} \cdot \sqrt{k} \le \ell \le 2 \cdot \sqrt{k}$, we have $\log \ell = \frac{1}{2} \log k + O(1)$. Thus

$$\sum_{i=0}^{k-2} a_i = O(k^2) + k^2 \cdot (\log k - \frac{\log k}{2} + O(1)) = \frac{k^2 \log k}{2} + O(k^2).$$

□

Theorem 2.4. *The worst expected runtime of the discordant push protocol on the star graph with n vertices is $\frac{1}{8} n^2 \log n + O(n^2)$.*

Proof. We prove for odd n. After running the above discussed steps of Gaussian elimination on the matrix A, we obtain the following matrix. We only visualize the first row $A[1]$ and the rows $A[k-1]$, $A[k]$, $A[k+1], \ldots, A[n-4]$, $A[n-3]$, $A[n-2]$.

$$\begin{pmatrix} 0\,0\,0 \cdots 0\,0\,0 & 0 & 0 & 0 & 0 & \cdots 0 & 0 & 0 & 1 \\ & \vdots & & & & & & & \\ & & & & & & & & \\ 0\,0\,0 \cdots 0\,0\,{-1} & 1 & 0 & 0 & 0 & \cdots 0 & 0 & 0 & 0 \\ 0\,0\,0 \cdots 0\,0\,0 & -1 & 1 & 0 & 0 & \cdots 0 & 0 & 0 & 0 \\ 0\,0\,0 \cdots 0\,0\,0 & 0 & -1 & 1 & 0 & \cdots 0 & 0 & 0 & 0 \\ & \vdots & & \ddots & \ddots & & & & \\ & \vdots & & & \ddots & \ddots & & & \\ & \vdots & & & & \ddots & \ddots & & \\ 0\,0\,0 \cdots 0\,0\,0 & 0 & 0 & 0 & 0 & \cdots -1 & 1 & 0 & 0 \\ 0\,0\,0 \cdots 0\,0\,0 & 0 & 0 & 0 & 0 & \cdots 0 & -1 & 1 & 0 \\ 0\,0\,0 \cdots 0\,0\,0 & 0 & 0 & 0 & 0 & \cdots 0 & 0 & -1 & 1 \end{pmatrix}$$

Hence, to obtain the row vector all of whose entries are 0 except for the $(k-1)$-th, we need to compute $A[1] - (A[k] + \cdots + A[n-2])$. The current values on the right-hand sides of the equations corresponding to rows 1, k, $k+1$, $k+2, \ldots, n-2$ are a_{k-2}, $1 - \frac{k}{k+1} \cdot a_0$, $1 - \frac{k+1}{k+2} \cdot a_1$, $1 - \frac{k+2}{k+3} \cdot a_2, \ldots$, $1 - \frac{k+(k-3)}{k+(k-3+1)} \cdot a_{k-3}$, respectively. Thus the value of the $(k-1)$-th unknown is $a_{k-2} + \left(\frac{k}{k+1} \cdot a_0 - 1\right) + \left(\frac{k+1}{k+2} \cdot a_1 - 1\right) + \cdots + \left(\frac{k+(k-3)}{k+(k-3+1)} \cdot a_{k-3} - 1\right) = \left(1 + O\left(\frac{1}{k}\right)\right) \cdot \left(\sum_{i=0}^{k-2} a_i\right) + O(k) = \frac{1}{2}k^2 \log k + O(k^2) = \frac{1}{8}n^2 \log n + O(n^2)$. □

2.3. Pull voting on the star graph

Theorem 2.5. *The worst expected runtime of the discordant pull protocol on the star graph with n vertices is $\frac{1}{6}n^2 + \frac{1}{6}n \log n + O(n)$.*

Proof. We prove for odd n. The same notation is used as in the case of push voting. There is only a slight difference in the transition matrix: the probability of transition from i to $n-1-i$ is $\frac{1}{i+1}$, and the probability of transition from i to $i-1$ is $\frac{i}{i+1}$.

The $(n-2) \times (n-2)$ matrix $I - Q$ derived from the transition matrix looks as follows (we do the illustration and the calculation for odd n, the case of even n is very similar, and of course, the same estimation is obtained in the end); $k = (n+1)/2$:

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} \\
-\frac{2}{3} & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & -\frac{1}{3} & 0 \\
0 & -\frac{3}{4} & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & -\frac{1}{4} & 0 & 0 \\
0 & 0 & -\frac{4}{5} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & -\frac{1}{5} & 0 & 0 & 0 \\
\vdots & \vdots & & \ddots & \ddots & & & & & \iddots & & & & \vdots \\
 & \cdots & 0 & 0 & -\frac{k-3}{k-2} & 1 & 0 & 0 & 0 & -\frac{1}{k-2} & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & -\frac{k-2}{k-1} & 1 & 0 & -\frac{1}{k-1} & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & 0 & -\frac{k-1}{k} & \frac{k-1}{k} & 0 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & 0 & -\frac{1}{k+1} & -\frac{k}{k+1} & 1 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & -\frac{1}{k+2} & 0 & 0 & -\frac{k+1}{k+2} & 1 & 0 & 0 & \cdots & \\
 & & \vdots & & \iddots & & \vdots & & & \ddots & \ddots & \vdots & & \\
0 & 0 & 0 & -\frac{1}{n-4} & 0 & \cdots & & \cdots & 0 & -\frac{n-5}{n-4} & 1 & 0 & 0 & 0 \\
0 & 0 & -\frac{1}{n-3} & 0 & 0 & \cdots & & \cdots & 0 & 0 & -\frac{n-4}{n-3} & 1 & 0 & 0 \\
0 & -\frac{1}{n-2} & 0 & 0 & 0 & \cdots & & \cdots & 0 & 0 & 0 & -\frac{n-3}{n-2} & 1 & 0 \\
-\frac{1}{n-1} & 0 & 0 & 0 & 0 & \cdots & & \cdots & 0 & 0 & 0 & 0 & -\frac{n-2}{n-1} & 1
\end{pmatrix}$$

This matrix is denoted by B. We follow a similar strategy as in the previous subsection. Of course, some factors need to be modified in the Gaussian elimination steps.

The first step is a special one: $B[k-1]* = \frac{k}{k-1}$.

Then we apply the following triples of steps for $i = 0, 1, \ldots, k-3$:

$$B[k+i]- = \frac{1}{k+i+1} B[k-i-1]$$
$$B[k-i-2]+ = (B[k-i-1] + B[k+i])$$
$$B[k-i-2]* = \frac{k-i-1}{k-i-2}$$

Initially, there is 1 at every entry in the right-hand side vector of the system of linear equations. It is easy to show by induction that after these steps, we have the following values on the right-hand side of the equations:

$$\begin{aligned}
y_{k-1} &= \frac{k}{k-1} \\
y_{k-2} &= \left(\frac{k}{k-1}\cdot\frac{k}{k+1}+2\right)\cdot\frac{k-1}{k-2} = \frac{k^2+2(k^2-1)}{k^2-1}\cdot\frac{k-1}{k-2} \\
y_{k-3} &= \left(\left(\frac{k}{k-1}\cdot\frac{k}{k+1}+2\right)\cdot\frac{k-1}{k-2}\cdot\frac{k+1}{k+2}+2\right)\cdot\frac{k-2}{k-3} \\
&= \frac{k^2+2(k^2-1)+2(k^2-4)}{k^2-4}\cdot\frac{k-2}{k-3}
\end{aligned}$$

and in general

$$\begin{aligned}
&y_{k-(i+1)} \\
&= \left(\cdots\left(\left(\frac{k}{k-1}\cdot\frac{k}{k+1}+2\right)\cdot\frac{k-1}{k-2}\cdot\frac{k+1}{k+2}+2\right)\cdots\right)\cdot\frac{k-i}{k-(i+1)} \\
&= \frac{k^2+2(k^2-1)+\cdots+2(k^2-i^2)}{k^2-i^2}\cdot\frac{k-i}{k-(i+1)} \\
&= \frac{k^2(2i+1)-2\cdot\frac{i(i+1)(2i+1)}{6}}{k^2-i^2}\cdot\frac{k-i}{k-(i+1)} = \frac{3k^2(2i+1)-i(i+1)(2i+1)}{3(k+i)(k-(i+1))}.
\end{aligned}$$

In particular, $y_1 = \frac{3k^2(2k-3)-(k-2)(k-1)(2k-3)}{3(2k-2)(k-(k-2+1))} = \frac{4k^3+O(k^2)}{6k+O(1)} = \frac{2}{3}k^2 + O(k)$.

To obtain the row vector with a single 1 in the $(k-1)$-th entry, we need to return the following difference: $B[1]-(B[k]+B[k+1]+\cdots+B[n-2])$, see the illustration in the previous subsection. Hence, the value of the $(k-1)$-th unknown is y_1+S where S stands for the sum $-(y_k+y_{k+1}+\cdots+y_{n-2})$. In order to estimate S, we use the equations $-y_{k+i} = \frac{1}{k+i+1}\cdot y_{k-(i+1)} - 1$ for $i \geq 0$. That is,

$$\begin{aligned}
S = \sum_{i=0}^{k-3}\left(\frac{1}{k+i+1}\cdot y_{k-(i+1)}-1\right) &= O(k) + \sum_{i=0}^{k-3}\frac{1}{k+i+1}\cdot y_{k-(i+1)} = \\
&= O(k) + \sum_{i=0}^{k-3}\frac{3k^2(2i+1)-i(i+1)(2i+1)}{3(k+i)(k+i+1)(k-(i+1))}.
\end{aligned}$$

Note that the denominator of the summands is at least k^2 for all i. Thus any constant, linear and quadratic term in the numerator is negligible. Indeed, for each value of i, the contribution of such a term is $O(1)$, and there are $O(k)$ different values of i, yielding an $O(k)$ contribution altogether. Hence, we can replace the expression $\frac{3k^2(2i+1)-i(i+1)(2i+1)}{3(k+i)(k+i+1)(k-(i+1))}$ by something else that has the same denominator $3(k+i)(k+i+1)(k-(i+1))$ and the same cubic terms $6k^2i-2i^3$ in its numerator. A straightforward calculation shows that

$$\frac{1}{3}\left(\frac{i}{k-1-i}+\frac{5ki+3i^2}{(k+i)(k+i+1)}\right)$$

is an appropriate replacement in this sense. Thus

$$S = O(k) + \frac{1}{3} \cdot \sum_{i=0}^{k-3} \left(\frac{i}{k-1-i} + \frac{5ki + 3i^2}{(k+i)(k+i+1)} \right).$$

The second fraction $\frac{5ki+3i^2}{(k+i)(k+i+1)}$ in the summand again yields an $O(k)$ error, as $5ki + 3i^3 \leq 8k^2$, and the denominator is at least k^2, making the second fraction at most 8 in all $k-2$ summands of the sum. The first fraction $\frac{i}{k-1-i}$ can be written as $\frac{k-1}{k-1-i} - 1$, and the -1 is once again negligible since it only contributes an $O(k)$ error. In summary, $S = O(k) + \frac{1}{3} \cdot \sum_{i=0}^{k-3} \frac{k-1}{k-1-i} = O(k) + \frac{k-1}{3} \cdot \sum_{j=2}^{k-1} \frac{1}{j}$. Since $\sum_{j=2}^{k-1} \frac{1}{j} = \log k + O(1)$, we have $S = O(k) + \frac{1}{3} k \log k$.

Hence, the desired expected runtime is

$$y_1 + S = \frac{2}{3}k^2 + \frac{1}{3}k \log k + O(k) = \frac{1}{6}n^2 + \frac{1}{6}n \log n + O(n).$$

□

Funding Open access funding provided by HUN-REN Alfréd Rényi Institute of Mathematics.

References

[1] Basu, R., Sly, A.: Evolving voter model on dense random graphs. Annal. Appl. Probab. **27**(2), 1235–1288 (2017)

[2] Carra, D., Cigno, R.L., Russo, A.: On some fundamental properties of p2p push/pull protocols. In *Proceedings of the Second International Conference on Communications and Electronics*, p. 7 pp (2008)

[3] Cooper, C., Dyer, M., Frieze, A., 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., Radzik, T.: Coalescing random walks and voting on connected graphs. SIAM J. Discret. Math. **27**(4), 1748–1758 (2013)

[5] Cooper, C., Rivera, N.: The linear voting model. In: 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016), pp. 2021–2032

[6] Donnelly, P., Welsh, D.: Finite particle systems and infection models. Math. Proc. Cambridge Philos. Soc. **94**(1), 167–182 (1983)

[7] Doyle, P.G., Snell, J.L.: Random walks and electric networks. Mathematical Association of America, (1984)

[8] Durrett, R., Gleeson, J.P., Lloyd, A.L., Mucha, P.J., Shi, F., Sivakoff, D., Socolar, J.E.S., Varghese, C.: Graph fission in an evolving voter model. Proc. Natl. Acad. Sci. U.S.A. **109**(10), 3682–3687 (2012)

[9] Hassin, Y., Peleg, D.: Distributed probabilistic polling and applications to proportionate agreement. Inf. Comput. **171**(2), 248–268 (2001)

[10] Hedetniemi, S.M., Hedetniemi, S.T., Liestman, A.L.: A survey of gossiping and broadcasting in communication networks. Networks **18**(4), 319–349 (1988)

[11] Karp, R., Schindelhauer, C., Shenke, S., Vocking, B.: Randomized rumor spreading. In: FOCS '00 Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pp. 565–574 (2000)

[12] Nakata, T., Imahayashi, H., Yamashita, M.: Probabilistic local majority voting for the agreement problem on finite graphs. Springer, (1999)

[13] Oliveira, R.: On the coalescence time of reversible random walks. Trans. Am. Math. Soc. **364**(4), 2109–2128 (2012)

[14] Oliveira, R.: Mean field conditions for coalescing random walks. Ann. Probab. **41**(5), 3420–3461 (2013)

[15] Pongrácz, A.: Discordant voting protocols for cyclically linked agents. Electron. J. Combinator. 27, 1, (2020)

Kamilla Kátai-Urbán
Bolyai Institute
University of Szeged
Aradi Vértanúk square 1
Szeged 6720
Hungary
e-mail: katai@math.u-szeged.hu

András Pongrácz (✉)
Alfréd Rényi Institute of Mathematics
Reáltanoda street 13-15
Budapest 4032
Hungary
e-mail: pongracz.andras@renyi.hu

Csaba Szabó
Department of Algebra and Number Theory
Eötvös Loránd University
Pázmány Péter sétány 1/C
Budapest 1117
Hungary
e-mail: csaba@cs.elte.hu

Received: March 3, 2024.

Accepted: March 4, 2024.