VOTING PROTOCOLS ON THE STAR GRAPH

ABSTRACT. Given a finite graph $G = (V, E)$ and an assignment $\nu_0 : V \to \{0, 1\}$ that is the initial opinion of each vertex, discordant push voting is a non-deterministic protocol that produces a sequence of functions $\nu_1, \nu_2, \ldots$ which terminates as soon as a consensus is reached. More precisely, at each round a discordant vertex u (i.e., one that has a neighbour with a different opinion) is chosen uniformly at random, and then this vertex chooses a neighbour v with different vote uniformly at random, and forces v to change its opinion to that of u. The game stops when a consensus state is reached, that is, when the function ν_k is constant. In case of the discordant pull protocol we simply choose a discordant vertex uniformly at random and change its opinion. It was shown that this protocol is expected to stop in $\frac{1}{4}n^2 + O(n^{3/2})$ steps for the cycle graph if the process starts from the worst possible initial state, where $n = |V|$. The best known estimate for the star graph was that the expected time to reach consensus is between $C_1 n^2 \log n$ and $C_2 n^2 \log n$ for some positive constants C_1, C_2 (in the worst case). We show that the expected time for the push protocol to reach consensus on the star graph is $\frac{1}{8}n^2 \log n + O(n^2)$, and it is $\frac{1}{6}n^2 + O(n \log n)$ for the pull protocol, assuming the worst initial case.

Keywords: Voting, finite graph, cycle, star, push protocol

2010 *Mathematics subject classification*: Primary 60K37; Secondary 60J10, 91D10

1. INTRODUCTION

Models of voting in finite graphs have been studied intensively for decades, see [5, 8]. 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 (vertices of the graph) can affect the vote of their neighbours according to the given rules.

We note that many alternative definitions were investigated in the literature. Continuous time voting processes were studied in [5, 7]. In [6, 1] the graph evolves together with the opinions of the vertices. Connections of voting processes and coalescing random walks were investigated in [7, 9], and for other recent applications see [10, 3].

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. Such a protocol can be synchronous (see [4] 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 [4] as a common generalisation 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 that vertex forces a randomly chosen neighbour to adopt the opinion of u.
- Pull protocol: each round a vertex u is chosen uniformly at random, and that vertex is forced by a randomly chosen neighbour 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 [2]. 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 [2] 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 $T_{\text{push}} = \frac{1}{8} n^2 \log n + O(n^2)$ time. The pull protocol is the fastest out of the three above defined processes for the star graph. Its expected runtime is $T_{\text{pull}} = \frac{1}{6}n^2 + O(n \log n)$. It was already discussed in [2] that the oblivious protocol has expected runtime $T_{\text{oblivious}} = \frac{1}{4}n^2 + O(n)$. These results are somewhat counter-intuitive. As it was pointed out in [2], for a typical graph the push protocol should be the fastest out of the three, and the pull voting should be the slowest.

2. Preliminaries

2.1. **General notations.** Given an absorbing Markov chain P with transient states T. 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 notations, $N = (I - Q)^{-1}$ denotes the fundamental matrix of the Markov chain. We denote by $\underline{1}$ the column vector of length $|T|$ all of whose entries equal to 1. It is well-known that the expected times to absorption from each transient state as initial state are the coordinates of the vector $N\underline{1}$.

2.2. **Push voting on the star graph.** In order to make the problem more transparent, we define a Markov chain in the following way. Each possible state of the voting process is described by the number of neighbours of the central vertex in the star that have the opposite opinion as the center. If there are $i > 0$ such vertices, i.e., we are in state i, then the process can evolve to two possible states in general. If we pick a vertex out of the i discordant neighbours 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 neighbours 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, which is clearly the one with an equal number of zeros and ones (or as close to equal as possible). Note that it is impossible to reach state $n - 1$ from here: indeed, all the edges would be discordant, and none of the discordant voting protocols can move into such a state. 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 with this matrix on the left and the vector $\underline{1}$ on the right, and we are interested in the $(k-1)$-st element of that solution vector, corresponding to the worst case. Our strategy is to apply elementary steps of Gaussian elimination on the rows of A to obtain the row vector that is all zero except for the $(k-1)$-st element, which is 1. The following program shows the steps that need to be applied. We use standard notations that are used in several programming languages (e.g., Python); note that $A[i]$ is the i-th row of the matrix, where indexation of rows start by 1 (unlike with Python, where the first index of an array is 0).

The first step is a special one:

$A[k-1]* = k$

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

$A[k+i]- = \frac{k+i}{k+i+1} A[k-i-1]$

$A[k-i-2]+ = (A[k-i-1] + A[k+i])$

$A[k-i-2]* = (k-i-1)$

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

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

Lemma 2.1. *Let $(c_i)_{i=0\cdots(k-1)}$ be a sequence satisfying the same recursion 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. Trivial calculation. □

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) *There is an $\ell \leq \sqrt{2} \cdot \sqrt{k}$ such that $a_\ell \geq b_\ell$, and the smallest such index ℓ is at least $\frac{1}{2} \cdot \sqrt{k}$ if $k \geq 100$.*

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 $a_i \leq (1 - \frac{i}{k})a_{i-1} + 2k - 2i \leq (1 - \frac{i}{k})(2i-1)k + 2k - 2i = (2i+1)k - 2i^2 - i \leq (2i+1)k$.

The third item is shown indirectly: so assume that $a_i < b_i$ for all $i \leq \sqrt{2} \cdot \sqrt{k}$. Then by Lemma 2.1 the series (a_i) is strictly monotone increasing in any index less than $\sqrt{2} \cdot \sqrt{k}$. By item 1 of the present lemma, this means that $a_i \geq \frac{1}{\sqrt{2}} \cdot k^{3/2}$ for all $\frac{1}{\sqrt{2}} \cdot \sqrt{k} \leq i \leq \sqrt{2} \cdot \sqrt{k}$. On the other hand, if ℓ is the integer part of $\sqrt{2} \cdot \sqrt{k}$, then $\ell + 1 \geq \sqrt{2} \cdot \sqrt{k}$, so $b_\ell = \frac{k^2}{\ell+1} - (\ell+1) \leq \frac{k^2}{\sqrt{2} \cdot \sqrt{k}} - (\sqrt{2} \cdot \sqrt{k}) < \frac{1}{\sqrt{2}} \cdot k^{3/2} \leq a_\ell$, 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 $\frac{1}{2} \cdot \sqrt{k} \leq \ell \leq \sqrt{2} \cdot \sqrt{k}$ be the smallest index (provided by Lemma 2.2) such that $a_\ell \geq b_\ell$.*

(1) *For all $i \leq \ell$, $0 \leq a_i \leq 3k^{3/2}$, and in particular $0 \leq \sum_{i=1}^{\ell} a_i \leq 5k^2$.*
(2) $0 \leq \varepsilon_\ell < 6k$

(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* $0 \leq \sum_{i=\ell}^{k-2} \varepsilon_i < 6k^2$.

(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. The first item follows from item 2 of Lemma 2.2 and 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 following inequalities:

- $a_{\ell-1} < b_{\ell-1}$, by minimality of ℓ; that is, $a_{\ell-1} - b_{\ell-1} < 0$
- $a_\ell - a_{\ell-1} \leq 2k$ by the recursive rule that defines the series (a_i)
- finally, $b_{\ell-1} - b_\ell = (\frac{k^2}{\ell} - \ell) - (\frac{k^2}{\ell+1} - \ell - 1) < \frac{k^2}{\ell^2} \leq 4k$, as $\frac{1}{2} \cdot \sqrt{k} \leq \ell$

Adding up these three inequalities yields the second item of the lemma.

The third item is shown by first observing that $a_i = (b_{i-1} + \varepsilon_{i-1}) \cdot \frac{k-i}{k+i} + 2k - 2i = (\frac{k^2-i^2}{i} + \varepsilon_{i-1}) \cdot \frac{k-i}{k+i} + 2k - 2i = \frac{(k^2-2ki+i^2)+2ki-2i^2}{i} + \frac{k-i}{k+i} \cdot \varepsilon_{i-1} = \frac{k^2}{i} - i + \frac{k-i}{k+i} \cdot \varepsilon_{i-1} = \frac{k^2}{i+1} - (i+1) + \frac{k-i}{k+i} \cdot \varepsilon_{i-1} + \frac{k^2}{i(i+1)} + 1 = b_i + \frac{k-i}{k+i} \cdot \varepsilon_{i-1} + \frac{k^2}{i(i+1)} + 1$. 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 $i \geq \ell$. For the upper estimation of the sum $\sum_{i=\ell}^{k-2} \varepsilon_i$ we observe that $\frac{k-i}{k+i} \leq 1 - \frac{1}{2\sqrt{k}}$ for all $i \geq \ell$, as $\frac{1}{2} \cdot \sqrt{k} \leq \ell \leq i$. Hence, for any $\ell \leq i_1 < i_2 < \cdots < i_t$ we have $\sum_{u=1}^{t} \prod_{v=1}^{u} \frac{k-i_v}{k+i_v} \leq \sum_{m=0}^{\infty} (1 - \frac{1}{2\sqrt{k}})^m = 2\sqrt{k}$. So $\sum_{i=\ell}^{k-2} \varepsilon_i = \varepsilon_\ell + (\frac{k-\ell}{k+\ell} \varepsilon_\ell + \frac{k^2}{\ell(\ell+1)} + 1) + (\frac{k-(\ell+1)}{k+(\ell+1)} (\frac{k-\ell}{k+\ell} \varepsilon_\ell + \frac{k^2}{\ell(\ell+1)} + 1) + \frac{k^2}{(\ell+1)(\ell+2)} + 1) + \cdots$ If we expand all the parantheses in this expression, then the sum of all coefficients of ε_ℓ is thus 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 $\sum_{i=\ell}^{k-2} \varepsilon_i \leq 2\sqrt{k}\varepsilon_\ell + 2\sqrt{k} \sum_{i=\ell}^{k-2} \frac{k^2}{i(i+1)} + k^2/2 \leq 2\sqrt{k} \cdot 6k + 2k^{5/2} \sum_{i=\ell}^{k-2} (\frac{1}{i} - \frac{1}{i+1}) + k^2/2 \leq 12k^{3/2} + 2k^{5/2} \cdot \frac{1}{\ell} + k^2/2 = 12k^{3/2} + 4k^2 + k^2/2 \leq 6k^2$ as $k \geq 100$.

We now prove the fourth item. $\sum_{i=0}^{k-2} a_i = \sum_{i=0}^{\ell-1} a_i + \sum_{i=\ell}^{k-2} a_i = O(k^2) + \sum_{i=\ell}^{k-2} a_i$, by item 1. So the sum is $O(k^2) + \sum_{i=\ell}^{k-2} b_i + \sum_{i=\ell}^{k-2} \varepsilon_i = O(k^2) + \sum_{i=\ell}^{k-2} b_i = O(k^2) + \sum_{i=\ell}^{k-2} \frac{k^2}{i+1} - \sum_{i=\ell}^{k-2}(i+1) = O(k^2) + k^2 \cdot \sum_{i=\ell}^{k-2} \frac{1}{i+1} = O(k^2) + k^2 \cdot (\sum_{i=1}^{k-1} \frac{1}{i} - \sum_{i=1}^{\ell} \frac{1}{i}) = O(k^2) + k^2 \cdot (\log k - \log \ell + O(1)) = O(k^2) + k^2 \cdot (\log k - \log \ell)$.

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

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

□

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

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 visualise 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 $(k-1)$-st, we need to compute $A[1] - (A[k-1] + A[k] + \cdots + A[n-2])$. The current values on the right-hand sides of the equations corresponding to rows $1, k-1, k, k+1, k+2, \ldots, n-2$ are $a_{k-2}, a_0, 1 - \frac{k}{k+1} \cdot a_0, 1 - \frac{k+1}{k+2} \cdot a_1, 1 - \frac{k+2}{k+3} \cdot a_2, \ldots, 1 - \frac{k+(k-3)}{k+(k-3+1)} \cdot a_{k-3}$, respectively.

Thus the value of the $(k-1)$-st unknown is $a_{k-2} - k + (\frac{k}{k+1} \cdot a_0 - 1) + (\frac{k+1}{k+2} \cdot a_1 - 1) + \cdots + (\frac{k+(k-3)}{k+(k-3+1)} \cdot a_{k-3} - 1) = (1 + O(\frac{1}{k})) \cdot (\sum_{i=0}^{k-2} a_i) - 2 = \frac{1}{2} \cdot k^2 \log k + O(k^2) = \frac{1}{8} \cdot n^2 \log n + O(n^2)$. □

2.3. **Pull voting on the star graph.**

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

Proof. We prove for odd n. The same notations are 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); $k = (n+1)/2$:

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

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:
$A[k-1]* = \frac{k}{k-1}$
Then we apply the following triple steps for $i = 0, 1, \ldots, k-3$:
$A[k+i] -= \frac{1}{k+i+1} A[k-i-1]$

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

$x_{k-1}=\frac{k}{k-1}$
$x_{k-2}=(\frac{k}{k-1}\cdot\frac{k}{k+1}+2)\cdot\frac{k-1}{k-2}=\frac{k^2+2(k^2-1)}{k^2-1}\cdot\frac{k-1}{k-2}$
$x_{k-3}=((\frac{k}{k-1}\cdot\frac{k}{k+1}+2)\cdot\frac{k-1}{k-2}\cdot\frac{k+1}{k+2}+2)\cdot\frac{k-2}{k-3}=\frac{k^2+2(k^2-1)+2(k^2-4)}{k^2-4}\cdot\frac{k-2}{k-3}$
$x_{k-(i+1)}=(\cdots((\frac{k}{k-1}\cdot\frac{k}{k+1}+2)\cdot\frac{k-1}{k-2}\cdot\frac{k+1}{k+2}+2)\cdots)\cdot\frac{k-i}{k-(i+1)}=$
$=\frac{k^2+2(k^2-1)+\cdots+2(k^2-i^2)}{k^2-i^2}\cdot\frac{k-i}{k-(i+1)}=\frac{k^2(2i+1)-2\cdot\frac{i(i+1)(2i+1)}{6}}{k^2-i^2}\cdot\frac{k-i}{k-(i+1)}=\frac{3k^2(2i+1)-i(i+1)(2i+1)}{3(k+i)(k-(i+1))}$

In particular, $x_1=\frac{3k^2(2k-3)-(k-2)(k-1)(2k-3)}{3(2k-2)(k-(k-2+1))}=\frac{4k^3+O(k^2)}{6k+O(1)}=\frac{2}{3}k^2+O(k)$.

To obtain the row vector with a single 1 in the $(k-1)$-st entry, we need to return the following difference: $A[1]-(A[k]+A[k+1]+\cdots+A[n-2])$, see the illustration in the previous subsection. Hence, the value of the $(k-1)$-st unknown is $x_1-(x_k+x_{k+1}+\cdots+x_{n-2})$. In order to show that the sum $-(x_k+x_{k+1}+\cdots+x_{n-2})$ is negligible compared to x_1, we use the following equations (for $i\geq 1$):

$-x_{k+i}=\frac{1}{i+1}\cdot x_{k-(i+1)}-1$

Hence, the sum $-(x_k+x_{k+1}+\cdots+x_{n-2})$ can be written as

$-\frac{k}{k-1}+\sum_{i=1}^{k-3}(\frac{1}{i+1}\cdot x_{k-(i+1)}-1)=O(k)+\sum_{i=1}^{k-3}\frac{1}{i+1}\cdot x_{k-(i+1)}=$
$=O(k)+\sum_{i=1}^{k-3}\frac{1}{i+1}\cdot\frac{3k^2(2i+1)-i(i+1)(2i+1)}{3(k+i)(k-(i+1))}$

Using $\frac{2i+1}{i+1}\leq 2$ and $0\leq i(i+1)\leq k^2$ we obtain

$O(k)+O(k^2)\cdot\sum_{i=1}^{k-3}\frac{1}{(k+i)(k-i)}$

Using $\frac{1}{k+i}\leq\frac{1}{k}$ yields

$O(k)\cdot(O(1)+\sum_{i=1}^{k-3}\frac{1}{k-i})\leq O(k)(\log k+O(1))=O(k\log k)$.

So the biggest element of the solution vector is $\frac{2}{3}k^2+O(k\log k)=\frac{1}{6}n^2+O(n\log n)$.

□

References

[1] Basu, R., and Sly, A. Evolving voter model on dense random graphs. *Annals of Applied Probability 27*, 2 (2017), 1235–1288.

[2] 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.

[3] 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.

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

[5] Donnelly, P., and Welsh, D. Finite particle systems and infection models. *Mathematical Proceedings of the Cambridge Philosophical Society 94*, 1 (1983), 167–182.

[6] Durrett, R., Gleeson, J. P., Lloyd, A. L., Mucha, P. J., Shi, F., Sivakoff, D., Socolar, J. E. S., and Varghese, C. Graph fission in an evolving voter model. *Proceedings of the National Academy of Sciences of the United States of America 109*, 10 (2012), 3682–3687.

[7] Hassin, Y., and Peleg, D. Distributed probabilistic polling and applications to proportionate agreement. *Information and Computation 171*, 2 (2001), 248–268.

[8] Nakata, T., Imahayashi, H., and Yamashita, M. *Probabilistic local majority voting for the agreement problem on finite graphs*. Springer, 1999.

[9] Oliveira, R. On the coalescence time of reversible random walks. *Transactions of the American Mathematical Society 364*, 4 (2012), 2109–2128.

[10] Oliveira, R. Mean field conditions for coalescing random walks. *The Annals of Probability 41*, 5 (2013), 3420–3461.