Notes And Explanations of Paper ”Discordant Voting Process on Finite Graphs”

Wang Yi - 518030910413

July 3, 2020

1 Final Assignment Info

This final assignment is to give some summarization and more vivid explanation on the 2018 SIAM paper of ”Discordant Voting Processes on Finite Graphs” written by Colin Cooper, Martin Dyer, Alan Frieze and Nicolas Rivera. My main work in this assignment is:

- Giving a more detailed explanation on basic model the paper used (e.g. Discordant Voting Process, push and pull chains, etc) by text and graph.
- Explaining the structure of proof by graph and some summary text.
- Some note for detailed proof steps, which can be used as supplementary material while reading the paper.

2 Introduction to Asynchronous Discordant Voting Process and Our Conclusion

2.1 Voting Process And Its Application in Real World

Voting is a natural model, oringinated in the complex networks communitity. To understand its structure, we can use a common situation of daily life.
Assume that there’s a group of people, holding different opinions for a question (e.g. acceptance, rejection, netrual). Some of people are the friend, penpal or other relationship of another person in the group, so they can communicate directly. In the communication, if two group members hold the same opinion, then they both reserve their opinion. Else, a debate will occur between them and finally one will be convinced by the other. Finally, the group members reach a consensus towards the question.
To discribe this situation, the model of voting process occurs. To give a not precise and mathmatical definition, it’s an undirected graph that each vertex has a color.The neighboring vertices interact pairwise with each other, updating their color in a fixed way, in which the vertices are in the same color.

2.2 Random Voting

A random voting means when choose some vertex or edge to update, then choose in random. To object to choose depends on ways to update.

2.3 Basic Ways to Update

- Push: Pick a random vertex set its random neighbor's color to its color.
- Pull: Pick a random vertex and set its color to its random neighbor's color.
- Oblivious: Pick a random edge and set the color of both endpoints to a randomly choosed endpoint.

2.4 Synchronous Voting and Asynchronous Voting

In the asynchronous voting, all the order, you can not choose two object at the same time. Consequently, the three ways to change are well defined.
However, with a synchronous voting, you can choose two objects in the same time, so push and oblivious are not well defined, which is showed in the Figure 1.

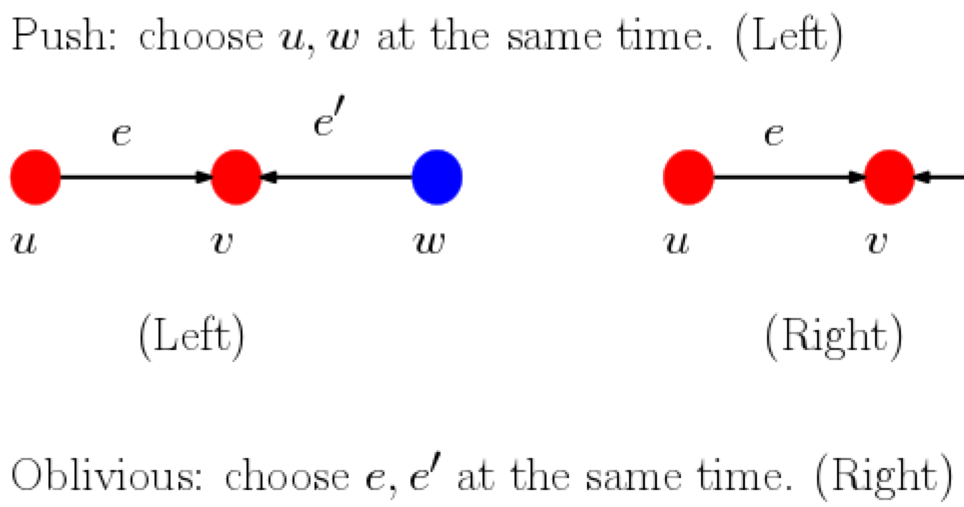


Figure 1: The condition that synchronous voting meets conflicts

In this paper, the discordant condition is asynchronous, so that push, pull and oblivious are all well-defined.

2.5 Discordant Voting

The discordant voting has the following features.

- The initial opinion number is 2, which is represented by red and blue.
- The interaction only acts at discordant edges, which means edge with two endpoints in different colors.

2.6 Our Goal - Time to Consensus in the Asynchronous Discordant Voting Process

Since the process is asynchronous, in each update step, a vertex's color is changed. Denote T as the step needed to reach the consensus state. This paper's main goal is to calculate the expection of T in different kind of connected n-vertex graphs, starting from R, B(means the numbers of red and blue vertices) with $R = B = \frac{n}{2}$.

The conclusion of this paper is in the following Figure 2.

	Discordant voting			Asynchronous voting		
	Push	Pull	Oblivious	Push	Pull	Oblivious
Complete graph K_n	$\Theta(n \log n)$	$\Theta(2^n)$		$O(n^2)$	$O(n^2)$	$O(n^4)$
Cycle C_n	$\Theta(n^2)$	$\Theta(n^2)$		$O(n^2)$	$O(n^2)$	$O(n^3)$
Star graph S_n	$\Theta(n^2 \log n)$	$O(n^2)$	$n^2/4$	$O(n^2)$	$O(n^2)$	$O(n^3)$
Double star S_n^*	$\Omega(2^{n/5})$	$O(n^4)$		$O(n^3)$	$O(n^4)$	$O(n^3)$
Barbell graph	$\Omega(2^{n/10})$	$\Theta(2^{n/2})$		$O(n^4)$	$O(n^4)$	$O(n^4)$

Figure 2: Comparison of expected time to consensus **E**T for discordant and ordinary asynchronous voting protocols on connected n-vertex graphs

In the following part of this note, we will conclude how this chart was proved in this paper.

2.7 Asynchronous Discordant Voting Model's Formal Definition

Given a graph $G = (V, E), n = |V|$.
$X(v) \in \{0, 1\}$ means the opinion of each vertex $v \in V$.
The discordant edge $e = uv \in E$ has the feature $X(u) \neq X(v)$.
$K(X_t)$ means the set of discordant edges with a opinion X_t at time t.
Absorbing states means $X(v) = 0$ for all $v \in V$, or $X(v) = 1$ for all $v \in V$.
And then three Markov chains are defined for the update rule.

Definition 2.1. ***Push*** *Choose* $v_t \in D(X_t)$ *and a discordant neighbor* u_t*, uniformly at random, let* $X_{t+1}(u_t) \leftarrow X_t(v_t)$ *and* $Xt + 1(w) \leftarrow X_t(w)$ *otherwise.*

Definition 2.2. ***Pull*** *Choose* $v_t \in D(X_t)$ *and a discordant neighbor* u_t*, uniformly at random, let* $X_{t+1}(v_t) \leftarrow X_t(u_t)$ *and* $Xt + 1(w) \leftarrow X_t(w)$ *otherwise.*

Definition 2.3. ***Oblivious*** *Choose* $\{u_t, v_t\} \in K(X_t)$*, uniformly at random, with probability* $\frac{1}{2}, X_{t+1}(v_t) \leftarrow X_t(u_t)$ *and with probability* $\frac{1}{2}, X_{t+1}(u_t) \leftarrow X_t(v_t)$ *and* $Xt + 1(w) \leftarrow X_t(w)$ *otherwise.*
T is the step at which process reaches absorbing state.

3 Introduction of Target Graphs and Summarization For some of the Proof Structure

In this section, we will devide the proof into three part. Oblivious(both discordant and ordinary), ordinary push, ordinary pull. The rest part of proof about discordant voting will be discussed in the following sections.

3.1 Introduction of Target Graphs

They are easy to understand, with simple graphs for certain n for example. For a example of $n = 8$, we can see Figure 3.
The different feature of this graph, will profoundly infect the expected number of step that we concern in this paper.

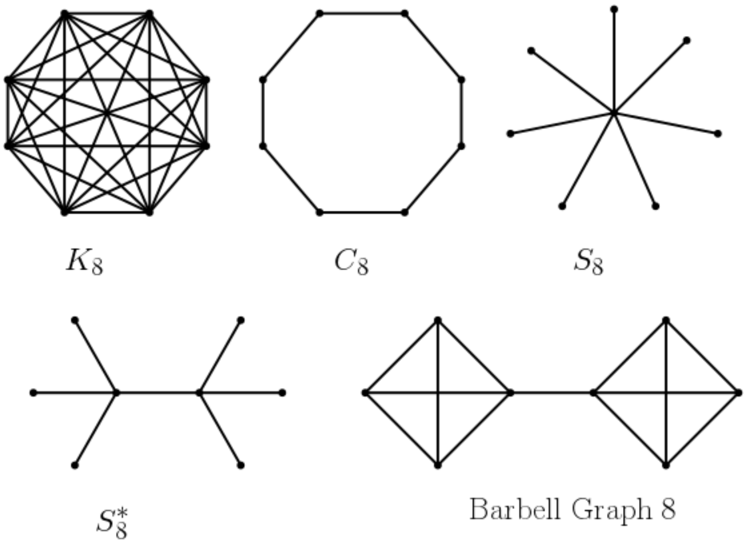


Figure 3: Graphs referred in this paper with $n = 8$

3.2 Proof of Asynchronous Oblivious Voting

In this subsection, we need to prove:

Theorem 3.1. **E**T(Oblivious) $= n^2/4$ *for discordant oblivious voting.*

Theorem 3.2. **E**T(Oblivious) $= O(mn^2)$ *for ordinary asynchronous oblivious voting.*
The main feature of oblivious voting is that no matter which edge to choose, the probability of the number of vertices in both color increases by 1 is the same, which is $1/2$.
Therefore, the problem can be solve naturaly with using some skill of unbiased random walk. So we can prove Theorem 3.1. by consider the result for some certain paramters of unbiased random walk problem.
For theorem 3.2., we prove it by transform it into some result of discordant oblivious voting.
For the prove structure of this subsection, see Figure 4 below. The exact expectation of ordinary oblivious model is in Table 1, following theorem 3.2.

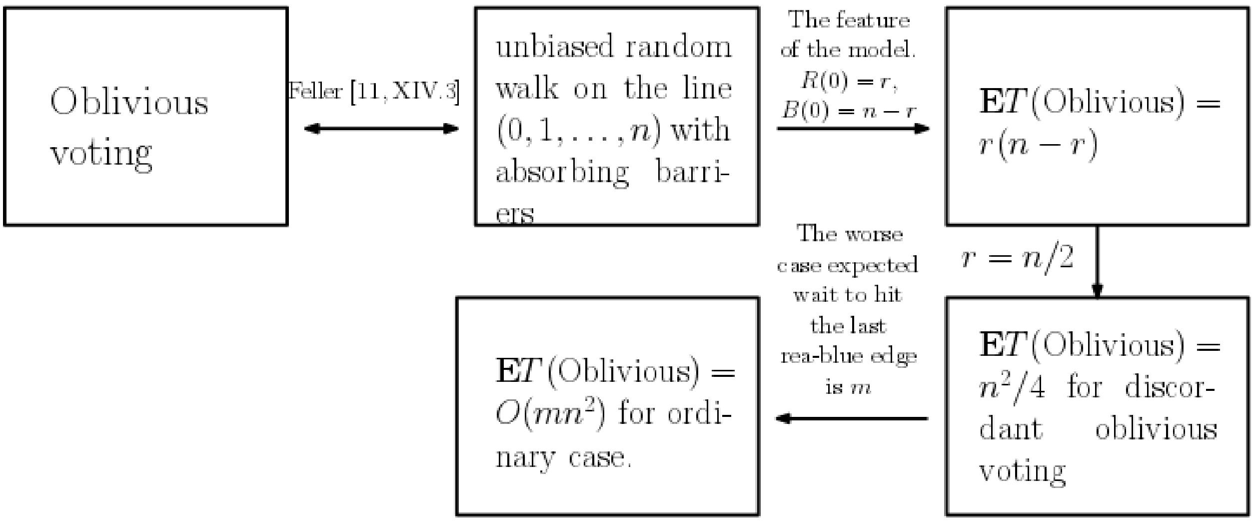


Figure 4: Proof of asynchronous oblivious voting

	m	d_{min}	Φ	oblivious	pull
Complete graph K_n	$n^2/2 - n/2$	$n-1$	$\Omega(1)$	$O(n^4)$	$O(n^2)$
Cycle C_n	n	2	$\Omega(1)$	$O(n^3)$	$O(n^2)$
Star graph S_n	$n-1$	1	$\Omega(1)$	$O(n^3)$	$O(n^2)$
Double star S_n^*	$n-1$	1	$\Omega(1/n)$	$O(n^3)$	$O(n^3)$
Barbell graph n	$n^2/4 - n/2 + 1$	$n/2 - 1$	$\Omega(1/n^2)$	$O(n^4)$	$O(n^4)$

Table 1: An table for pull and oblivious

3.3 Proof of Asynchronous Pull Voting

The proof of asynchronous pull voting is very simple that only use a formula **E**$T = O(nm/d_{min}\Phi)$. The formula was shown in "C. Cooper and N. Rivera. The linear voting model. ICALP 2016.", which is used as a known conclusion in this paper.

For this varibles,m means the number of edges, d_{min} is the minimum degree and Φ the graph conductance. Simply consider of the graph's features, we can get the results in the Table 1.

3.4 Proof of Asynchronous Push Voting

It's still an application of result of "C. Cooper and N. Rivera. The linear voting model. ICALP 2016." The theorem is that,

Theorem 3.3. *For any graph* $G = (V(G), E(G))$*,* **E**$T(\text{push}) = O(1/(G))$*, where*

$$\Psi(G) = \frac{2C(G)}{n d_{\max}} \min_{S \subset V(G)} \frac{1}{\min\{J(S), J(S^c)\}} \sum_{(v,w) \in E(S:S^c)} \frac{1}{d(v)d(w)}$$

The meaning of this paramters can refer to the paper. Therefore, we get the result of Asynchronous push voting.

4 Tools for following proving - push chain and pull chain

Push chain and Pull chain are tools with great importance in the following proof, so we introduce and prove some of their features (upperbound, lowerbound on hitting time) to prepare for the latter proof.

4.1 Birth-and-Death Chain

The push chain and pull chain are based on the concept of Birth-and-Death chain, which satisfy that the interaction in discordant voting process can only change the number of vertices in one color by one.

Definition 4.1. *Birth-and-Death chain on state space* $S = \{0, \ldots, N\}$ *is*

- *A Markov chain* $(X_t)_{t \geq 0}$
- *If given* $X_t = i$ *then the possible values of* X_{t+1} *are* $i+1, i$ *or* $i-1$ *with probability* p_i, r_i *and* q_i *respectively.*
- $q_0 = p_N = 0$.

To meet the model, we assume that $r_i = 0, p_0 = 1, q_N = 1, p_i > 0$ for $i \in \{0, \dots, N-1\}$ and $q_i > 0$ for $i \in \{1, \dots, N\}$.
Denote $\mathbf{E}_i T_j$ as the expection of hitting time of state i, $\min\{t \geq 0 : X_t = i\}$, when the chain has $X_0 = i$. The thing we care about is $\mathbf{E}_0 T_M$. There's a theorem that

Theorem 4.2. *For Birth-and-Death chains with* $p_i = P(i, i+1), q_i = Q(i, i-1)$,

$$\mathbf{E}_0 T_M = \sum_{i=1}^{M} \mathbf{E}_{i-1} T_i = \sum_{i=1}^{M} \sum_{k=0}^{i-1} \frac{1}{p_k} \prod_{j=k+1}^{i-1} \frac{q_j}{p_j}$$

The prove structure of Theorem 4.2. is in the Figure 5.

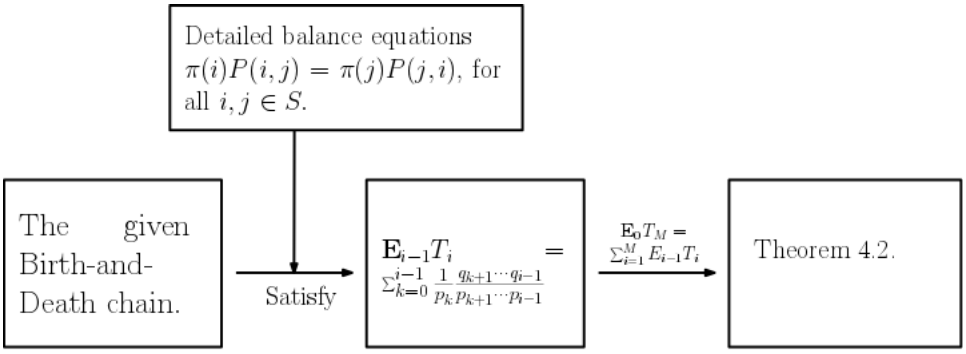


Figure 5: Proof of Theorem 4.2.

4.2 Definition of Push Chain and Pull Chain

Definition 4.3. *Push Chain: Let* Z_t *be the state occupied by the push chain at step* $t \geq 0$. *Let* $\delta \in \{-1, 0, +1\}$ *be fixed. The transition probability* $p_i = P(i, i+1)$ *from* $Z_t = i$, *is given by*

$$p_i = \begin{cases} 1 & \textit{if } i = 0 \\ 1/2 + i/n + \delta/n & \textit{if } i \in \{1, \dots, n/2 - 1\} \\ 0 & \textit{if } i = n/2 \end{cases}$$

Definition 4.4. *Pull Chain: Let* $\overline{Z}_t$ *be the state occupied by the pull chain at step* $t \geq 0$. *The transition probability* $\overline{p}_i = \overline{P}(i, i+1)$ *from* $\overline{Z}_t = i$, *is given by*

$$\overline{p}_i = \begin{cases} 1 & \textit{if } i = 0 \\ 1/2 - i/n - \delta/n & \textit{if } i \in \{1, \dots, n/2 - 1\} \\ 0 & \textit{if } i = n/2 \end{cases}$$

Accoding to the relationship that $\overline{p}_i = \overline{P}(i, i+1) = 1 - P(i, i+1) = P(i, i-1) = q_i$

4.3 Push Chain's Bound of Hitting Time

There's two lemmas to be used in the latter paper that:

Lemma 4.5. *For any $M \le N$, let $\mathbf{E}_0 T_M$ be the expected hitting time of M in the push chain Z_t starting from state 0. Then*

$$\mathbf{E}_0 T_M \le 2N \log M + O(1)$$

Lemma 4.6. *Let $\delta = 0$. Let $\mathbf{E}_0 T_M$ be the expected hitting time of M in the push chain Z_t starting from state 0. There exists a constant C such that, for any $\sqrt{N} \le M = O(N^{3/4})$,*

$$\mathbf{E}_0 T_M \ge C(N \log M/\sqrt{N} + \sqrt{N})$$

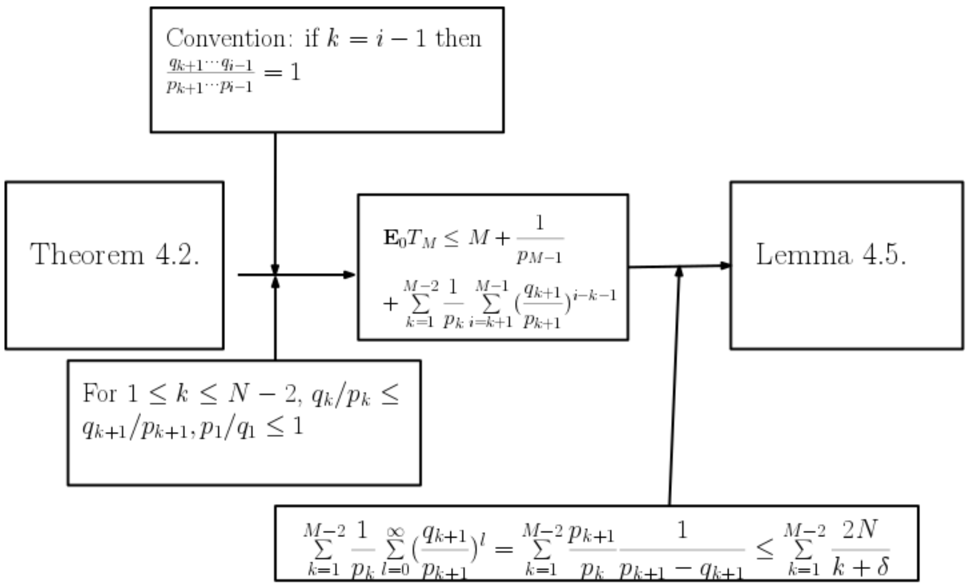


Figure 6: Proof of Lemma 4.5.

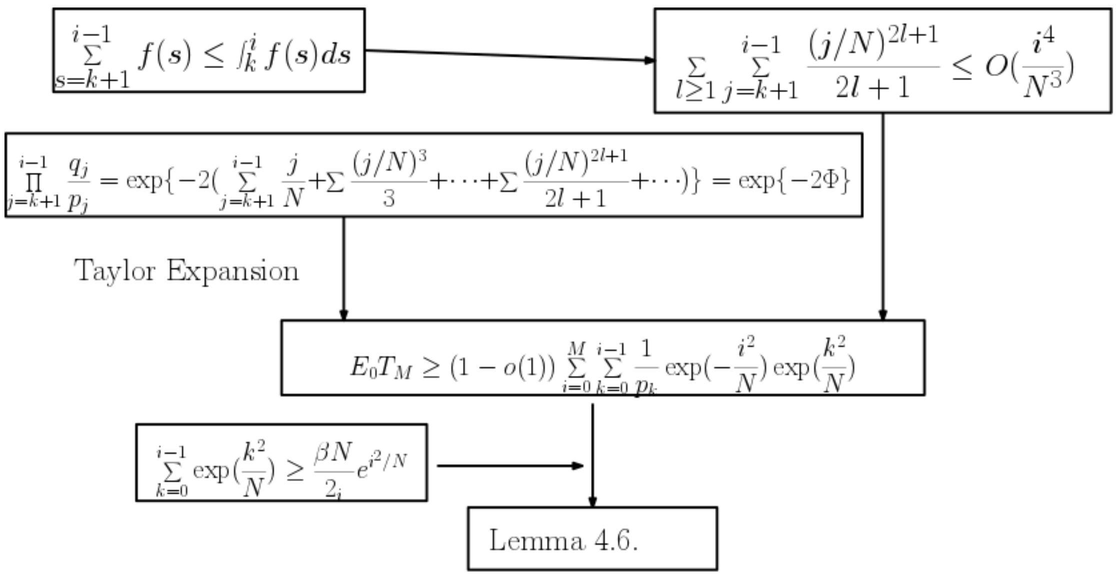


Figure 7: Proof of Lemma 4.6.

5 Proof structure of voting on the complete graph K_n

The theorem concerning the hitting time of voing on the complete graph K_n can be claim in the below, accoding to the table above.

Theorem 5.1. *Let* T *be the time to consensus of the asynchronous discordant voting process starting from any initial coloring with an equal number of red and blue vertices* $R = B = n/2$*. For the complete graph* K_n*,*$\mathbf{E}T(Push) = \Theta(n \log n)$ *and* $\mathbf{E}T(Pull) = \Theta(2^n)$*.*
To finish the proof, we need to transform the push process and pull process to model of push and pull processes respectively, and then finish finding the bound of the limit by Lemma 4.5. and Lemma 4.6.
The chain defined by the voting process is that

- $Y_t = \max\{R(t), B(t)\} - n/2$
- B increases at a given step with the probability $B(t)/n$ in push process and $R(t)/n = 1 - B(t)/n$ in the pull process.

The target is $\mathbf{E}_0 T_{N=n/2}$ for Y_t, which equals to the hitting time of this model.
It's easy to find Y_t for push model satisfies push chain Z_t and the pull model satisfies pull chain $\overline{Z}_t$by definition, when $\delta = 0$.
Therefore, the upper bound of hitting time in push model is determined to be $O(n \log n)$ by Lemma 4.5.
And for the pull model, we get $\pi(k) = \binom{n}{N+k}/W, W = \binom{n}{N} + \binom{n}{N+1} + \cdots + \binom{n}{n}$, which can be shown satisfies the detail balance equation.
For the lower bound, use the $\mathbf{E}_{i-1}T_i = \frac{2n}{n+2i} \cdot \frac{1}{\binom{n}{N+i}} \cdot \sum_{k=0}^{i-1} \binom{n}{N+k}$, we get $\mathbf{E}_0 T_N \geq \mathbf{E}_{N-1} T_N = \Omega(2^n)$.
For the upper bound, $\sum_{i=1}^{N} \mathbf{E}_{i-1}T_i \leq 2 \cdot 2^n \cdot \sum_{i=1}^{N} \frac{1}{\binom{n}{N+i}} = O(2^n)$. The inequality holds for the feature of q_i and $\sum_{k=0}^{i-1} \pi(k)$ and $\sum_{i=1}^{N} \frac{1}{\binom{n}{N+i}} = O(1)$ from the result of "B Sury. Sum of the reciprocals of the binomial coefficients. European Journal of Combinatorics, 14 351-353, 1993. ".

6 Proof structure of voting on the Cycle C_n

The theorem concerning the hitting time of voing on the cycle C_n can be claim in the below, accoding to the table above.

Theorem 6.1. *Let* T *be the time to consensus of the asynchronous discordant voting process starting from any initial coloring with an equal number of red and blue vertices* $R = B = n/2$*. For the cycle* C_n*,*$\mathbf{E}T(Push) = \mathbf{E}T(Pull) = \Theta(n^2)$*.*

6.1 New definitions

Definition 6.2. $i+1, i+2, \dots, j$ *is a run of length* $(j - i)$ $(1 \leq j - i < n)$ *if* $X(i) \neq X(i+1) = X(i+2) = \cdots = X(j) \neq X(j+1)$*.*

Definition 6.3. *Singleton is a run of length 1.* Denote $k(X)$ the number of runs.
Denote $r(X)$ the number of paths of a given colour, $r(X) = 1/2k(X)$, and $k_0 = 2r_0 = k(X_0)$.
Then the hitting time is the first t for which $k(X_t) = r(X_t) = 0$ Denote then length of runs $l_1, l_2, \dots, l_k$

respectively.
Denote $\kappa = 2k - s$ for discordant vertices number.

Definition 6.4. *Protential function*

$$\psi(x) = \sum_{i=1}^{k} \sqrt{l_i}$$

Definition 6.5. *Phase* r *of the process means* $\{t : k(X_t) = 2r\}$*, which is an interval* $[t_r, t_{r-1})$.

Definition 6.6. $\delta_v = \mathbf{E}[\psi(X_{t+1}) - \psi(X_t)|v_t = v]$ *is the expected change in* ψ.

Definition 6.7. *Total expected change is* $\Delta = \mathbf{E}[\psi(X_{t+1}) - \psi(X_t)] = 1/\kappa \sum_{v\in D} \delta_v$.
We partition the set K into the following subsets, which is like the part shown in Figure 8.

- $A = \{uv : u$ and v not singleton$\}$
- $B = \{uv : u$ not singleton, v singleton$\}$
- $C = \{uv : u$ and v both singleton$\}$

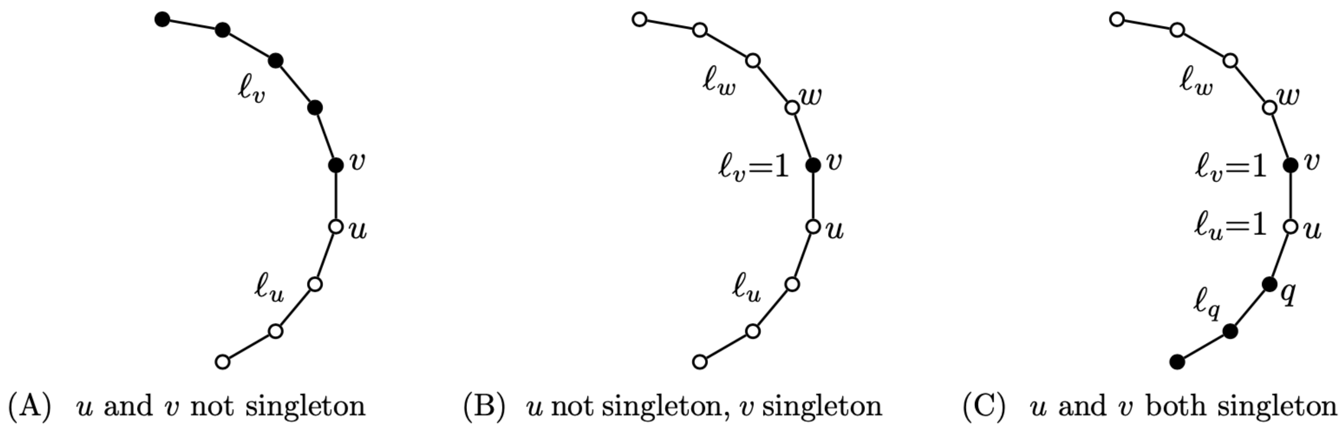


Figure 8: The partition of discordant edge.

Then denote λ_{uv} and δ_{uv}

$$\lambda_{uv} = \begin{cases} \sqrt{l_u} + \sqrt{l_v} & uv \in A \\ \sqrt{l_u} + 1/2\sqrt{l_v} & uv \in B \\ 1/2\sqrt{l_u} + 1/2\sqrt{l_v} & uv \in C \end{cases}$$

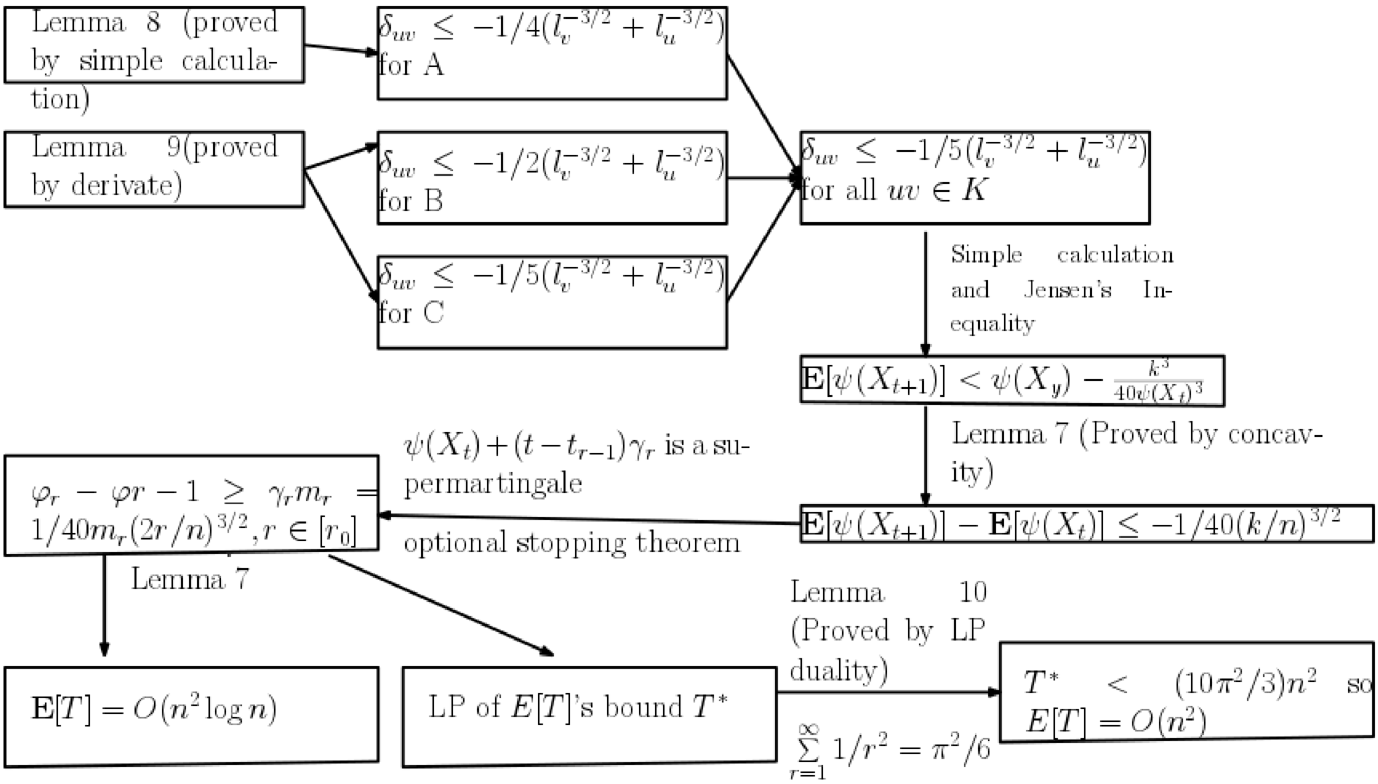
$$\delta_{uv} = \begin{cases} \delta_u + \delta_v & uv \in A \\ \delta_u + 1/2\delta_v & uv \in B \\ 1/2\delta_u + 1/2\delta_v & uv \in C \end{cases}$$

So by definition, we get

$$\psi = 1/2 \sum_{v\in D} \sqrt{l_v} = \sum_{uv\in K} \lambda_{uv}, \delta = 1/\kappa \sum_{v\in D} \delta_v = 1/\kappa \sum_{uv\in K} \delta_{uv}$$

6.2 The proof structure

See the Figure 9 for the proof structure of push voting upper bound.
The proof of pull voting is the similar to push voting. With a $\delta_{uv} < -1/10(l_v^{-3/2} + l_u^{-3/2})$, the expected convergence time estimate are doubled, while there's still $\mathbf{E}[T] = O(n^2)$



Lemma 7: For any configuration X on the n-cycle with k runs, $\psi(X) \leq \sqrt{kn}$

Lemma 8: For all $l \geq 1$, $\sqrt{l+1} + \sqrt{l-1} \leq 2\sqrt{l} - 1/4 l^{-3/2}$

Lemma 9: For all $l_1, l_2 \geq 1$, $\sqrt{l_1} + \sqrt{l_2} + 1 \geq \sqrt{l_1 + l_2 + 1} + (3 - \sqrt{3})$

Lemma 10: Let $0 < b_1 < b_2 < \cdots < b_v$ and $c_1 > c_2 > \cdots > c_v > 0$. Then the linear program $\max \Sigma_{j=1}^{v} c_j x_j$ subject to $\Sigma_{j=1}^{r} x_j \leq b_r$, $x_r \geq 0 (r \in [v])$ has optimal solution $x_1 = b_1, x_j = b_j - b_{j-1} (j = 2, 3, ..., v)$.

Figure 9: The proof of the upper bound push voting.

For the lower bound, we consider this type of graph in Figure 10 below.

In this configuration, push and pull process proceed indentically. Let L_t be the length of red run at t, before convergence $k(X_t) = 2$ and $L_{t+1} \leftarrow L_t - 1$ with probability $1/2$, while $L_{t+1} \leftarrow L_t + 1$ with probability $1/2$. Therefore we can transform the model in to a symmetric simple random walk. In this famous model, we can get the $\mathbf{E}[T]$ is bounded by $\Omega(n^2)$
Accordingly, we get the expected convergence time $\Theta(n^2)$, thus the Theorem 6.1. is proved.

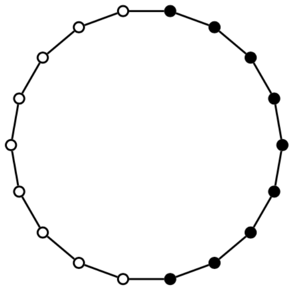


Figure 10: Lower bound configuration

7 Notes of voting on the Star Graph S_n, Double Star S_n^* and Barbell Graph

There's two Theorem in this part.

Theorem 7.1. *Let* T *be the time to consensus in the two-party asynchronous discordant voting process starting from any initial coloring with an equal number of red and blue vertices* $R = B = n/2$
For the star graph S_n, $\mathbf{E}T(Push) = \Theta(n^2 \log n)$, *and* $\mathbf{E}T(Pull) = O(n^2)$.
For the double star S_n^* *with the initial colouring separated in two part,* $\mathbf{E}T(Push) = \Omega(2^{n/5})$, *and* $\mathbf{E}T(Pull) = O(n^4)$.

Theorem 7.2. *Let* T *be the time to consensus in the two-party asynchronous discordant voting process starting from any initial coloring with an equal number of red and blue vertices* $R = B = n/2$.
For the barbell graph, $\mathbf{E}T(Push) = \Omega(2^{2n/5})$,$\mathbf{E}T(Pull) = \Theta(2^n)$
To prove this three kind of graph's convergence time, we payed great attention to central vertices.
We prove the conclusion of star graph by considering states with number of red vertices r and the color of central vertices.
We prove the conclusion of double star graph by considering one side only, and using the concept of trials and calculate the posibility of event about trial for push voting, and mimics the proof that on star graph for pull voting.
For barbell graph, the proof use similar tools above.
The proof of this three kind of graph is clear and easy to understand. And I read this paper because the proof of first two parts and the tool they used fascinates me, so I worked hard for the proof of the first two parts. Therefore, Although I have read the whole paper, I don't want to spend too much time on these part for making the proof more understandable, so I omit detailed explanation, which is really consuming.

References

[1]Colin Cooper, Martin Dyer, Alan Frieze and Nicolas Rivera. Discordant Voting Processes on Finite Graphs. arXiv:1604.06884v3, 2016.