Biased Voting and the Democratic Primary Problem

Michael Kearns and Jinsong Tan

Department of Computer and Information Science
University of Pennsylvania
{mkearns@cis, jinsong@seas}.upenn.edu

Abstract. Inspired by the recent Democratic National Primary, we consider settings in which the members of a distributed population must balance their individual preferences over candidates with a requirement to quickly achieve collective unity. We formalize such settings as the "Democratic Primary Problem" (DPP) over an undirected graph, whose local structure models the social influences acting on individual voters.

After contrasting our model with the extensive literature on diffusion in social networks (in which a force towards collective unity is usually absent), we present the following series of technical results:

– An impossibility result establishing exponential convergence time for the DPP for a broad class of local stochastic updating rules, which includes natural generalizations of the well-studied "voter model" from the diffusion literature (and which is known to converge in polynomial time in the absence of differing individual preferences).
– A new simple and local stochastic updating protocol whose convergence time is provably polynomial on any instance of the DPP. This new protocol allows voters to declare themselves "undecided", and has a temporal structure reminiscent of periodic polling or primaries.
– An extension of the new protocol that we prove is an approximate Nash equilibrium for a game-theoretic version of the DPP.

1 Introduction

The recent Democratic National Primary race highlighted a tension that is common in collective decision-making processes. On the one hand, individual voters clearly held (sometimes strong) preferences between the two main candidates, Barack Obama and Hillary Clinton, and these preferences appeared approximately balanced across the population. On the other hand, as the race progressed there were frequent and increasingly urgent calls for Democrats to "unify" the party — that is, quickly determine the winner and then all rally around that candidate in advance of the general election [13]. There was thus a balancing act between determining the overall (average) preference of voters, and reaching global consensus or unity.

Inspired by these events, we consider settings in which the members of a distributed population must balance their individual preferences over candidates with a requirement to achieve collective unity. We formalize such settings as the "Democratic Primary Problem" (DPP) over an undirected graph, whose local structure models the social influences acting on individual voters. In this model, each voter i is represented by a vertex in the network and a real-valued weight $w_i \in [0,1]$ expressing their preference for one of two candidates or choices that we shall abstractly call *red* and *blue*. Here $w_i = \frac{1}{2}$ is viewed as indifference between the two colors, while $w_i = 0$ (red) and $w_i = 1$ (blue) are "extremist" preferences for one or the other color.

Our overarching goal is to investigate distributed algorithms in which three criteria are met:

1. *Convergence to the Global Preference:* If the global average W of the w_i is even slightly bounded away from $\frac{1}{2}$ (indifference), then *all* members of the population should eventually settle on the globally preferred choice (i.e. all red if $W < \frac{1}{2}$, all blue if $W > \frac{1}{2}$), even if it conflicts with their own preferences (party unity).
2. *Speed of Convergence:* Convergence should occur in time polynomial in the size of the network.
3. *Simplicity and Locality:* Voters should employ "simple" algorithms in which they communicate only *locally* in the network via (stochastic) updates to their color choices. These updates should be "natural" in that they plausibly integrate a voter's individual preferences with the current choices of their neighbors, and do not attempt to encode detailed information, send "signals" to neighbors, etc.

The first two of these criteria are obviously formally precise. While it might be possible to formalize the third as well, we choose not to do so here for the sake of brevity and exposition. However, we are explicitly *not* interested in algorithms in which (for instance) voters attempt to encode and broadcast their underlying preferences w_i as a series of binary choices, or similarly unnatural and complex schemes. In particular, in our main protocol it will be very clear that voters are always updating their current choices in a way that naturally integrates their own preferences and the statistics of current choices in their local neighborhood.

We note that the formalization above clearly omits many important features of "real" elections. Foremost among these is the fact that real elections typically have strong global coordination and communication mechanisms such as polling, while we require that all communication between participants be entirely local in the network. On the other hand, our framework does allow for the presence of high-degree individuals, including ones that are indifferent to the outcome ($w_i = 1/2$) and can thus act as "broadcasters" of current sentiment in their neighborhood. Variation in degrees can also be viewed as a crude model for the increasing variety of global to local media sources (from "mainstream" publications to influential blogs to small discussion groups).

There is a large literature on the diffusion of opinion in social networks [4, 12, 10], but the topic is usually studied in the absence of any force towards collective unity. In many contagion-metaphor models, individuals are more or less susceptible to "catching" an opinion or fad from their neighbors, but are not directly concerned with the global outcome. In contrast, we are specifically interested in scenarios in which individual preferences are present, but are subordinate to reaching a unanimous global consensus.

Our main results are:

- An impossibility result establishing exponential convergence time for the DPP for a broad class of local stochastic updating rules, which includes natural generalizations of the well-studied "voter model" from the diffusion literature (and which is known to converge in polynomial time in the absence of differing individual preferences).
- A new simple and local stochastic updating protocol whose convergence time is provably polynomial on any instance of the DPP. This new protocol allows voters to declare themselves "undecided", and has a temporal structure reminiscent of periodic polling or primaries.
- An extension of the new protocol that we prove is an approximate Nash equilibrium for a game-theoretic version of the DPP.

2 The Democratic Primary Problem

The Democratic primary problem (DPP) is studied over an undirected graph $G = (V, E)$ with n nodes and m edges, where each node i represents an individual voter. Denote by $\mathcal{N}(i)$ the neighbors of i in G; we always consider i as a neighbor of himself.

There are two competing choices or opinions, that, without loss of generality, we shall call *blue* and *red* (or b and r for short). A voter i comes with a real-valued weight $w_i \in [0, 1]$ expressing his preference for one of the two opinions; without loss of generality, let $w_i(b) = w_i$ and $w_i(r) = 1 - w_i$ denote his preference for *blue* and *red*, respectively.

Throughout, we make the assumption that $\left|\sum_{i \in V} w_i(b) - \sum_{i \in V} w_i(r)\right| > \epsilon$ for some constant ϵ, that is, one opinion is always collectively preferred to the other; and we assume that which opinion is preferred is *not* known a priori to anyone and the goal of the DPP is for the entire population to actually figure this out through a distributed algorithm, or protocol, that is *simple and local*, and converges in time polynomial in n to the collectively preferred consensus. Because of the stochastic nature of the protocol we consider, it is implausible to require that it always converges to the collectively preferred consensus. Instead, we require the protocol does so with *high* probability, by which we mean the probability can differ from 1 by an amount that is at most negligible in n. We summarize the definition of DPP in the following.

Democratic Primary Problem (DPP)

Instance: Given an undirected graph $G = (V, E)$ with n nodes, two opinions $\{b, r\}$, and for each $i \in V$, a preference $(w_i(b), w_i(r))$ where $w_i(b), w_i(r) \in [0, 1]$ and $w_i(b) + w_i(r) = 1$. Assume $\exists\ \alpha \in \{b, r\}$ such that $\sum_i w_i(\alpha) > \frac{n}{2} + \epsilon$ for some constant $\epsilon > 0$ and α is not known a priori.

Objective: Design a *simple and local* distributed protocol that in time polynomial in n lets V converge to α with *high* probability.

We will be considering protocols of the following form:

1. (Initialization) At round 0, each node i in V independently and simultaneously initializes to an opinion in $\alpha \in \{b, r\}$ according to $\mathcal{I}$, a randomized function that maps i's local information to an opinion in $\{b, r\}$.
2. (Stochastic Updating) At round $t \geq 1$, a node i is chosen uniformly at random from V; i then picks a neighbor $j \in \mathcal{N}(i)$ according to a possibly non-uniform distribution over $\mathcal{N}(i)$; this distribution is determined by function $\mathcal{F}$, which is a randomized function whose arguments are i's local information. i then converts to j's opinion.

Therefore a protocol is specified by a pair of functions, $(\mathcal{I}, \mathcal{F})$. This framework by itself does not forbid "unnatural" coding behaviors as discussed in the Introduction; however in the spirit of emphasizing algorithms that are simple and local, we restrict $\mathcal{I}$ and $\mathcal{F}$ to be functions that only depend on simple and local information of a node i. In particular, only the following arguments to either functions are considered: 1) f_i, the distribution of opinions in the neighborhood, where $f_i(b)$ and $f_i(r)$ represent the fractions of neighbors currently holding opinion *blue* and *red*, respectively; 2) i's intrinsic preferences w_i; 3) i's degree $d_i = |\mathcal{N}(i)|$.

3 The Classic and Simplest: the Voter Model

The voter model, which was introduced by Clifford and Sudbury [2] and Holley and Liggett [5], is a well-studied probabilistic stochastic process that models opinion diffusion on social networks in a most basic and natural way. It consists of a class of protocols that satisfy our criterion of being *simple and local*. In fact, this class of protocols is the simplest that we examine in this paper. A voter model protocol is one where in each round, a node i is picked uniformly at random from V, and i in turn picks one of his neighbors uniformly at random and adopts his opinion; it does not specify how the initialization is done. More formally,

Definition 1 (Voter Model). *The voter model is a class of protocols of the form $(\mathcal{I}, \mathcal{F})$ where $\mathcal{F}(f_i) = \alpha$ with probability $f_i(\alpha)$, $\forall\ \alpha \in \{b, r\}$.*

Importantly, the voter model is a class of protocols in which there are no indiviudal preferences present at all, and the only concern is with reaching unanimity (to either color). This is in sharp contrast to the DPP setting. However, we shall make use of some known results on the voter model, which we turn to now.

Let C_{vm} denote the random variable whose value is the time at which a consensus is reached in a voter model protocol. It can be shown that $E(C_{vm}) = O(\log(n) \max_{i,j} h_{ij})$, where h_{ij} is the expected hitting time of node j of a random walk starting from node i (see [1] for a proof of this). Also it is well-known that for any graph G with self-loops (i.e. $i \in \mathcal{N}(i)$), $h_{ij} = O(n^3)$ for any node i, j [11], so $E(C_{vm}) = O(n^3 \log(n))$. We summarize this in the following theorem.

Theorem 1 ([1]). *For any initialization, it takes $O(n^3 \log(n))$ time in expectation for all the n nodes to converge to a consensus opinion in a voter model protocol.*

Denote by π the stationary distribution of a random walk on G, i.e. $\pi(i) = d_i/2m$ for all $i \in V$ and $\pi(S) = \sum_{i \in S} d_i/2m$ for all $S \subset V$. The next theorem also largely follows from established results in literature. And we omit the proof here.

Theorem 2. *Let $S \subset V$ be the set of nodes initialized to opinion α in a voter model protocol, then after n^5 rounds the probability that an α-consensus is reached differs from $\pi(S)$ by an amount negligible in n.*

Theorem 1 and 2 allow us to conclude that after n^5 rounds into a voter model protocol, with high probability *some* consensus is reached. In particular, let $B, R \in V$ be the set of nodes initialized to *blue* and *red* respectively, the probability of reaching a b-consensus (resp., r-consensus) differs from $\pi(B)$ (resp., $\pi(R)$) by negligible amount. Recall our goal of solving DPP is to find an efficient protocol that converges to the collectively preferred consensus with high probability. Since the voter model does not even consider w_i, it is clear that it does not solve the DPP. (The voter model does not specify how initialization is done, however it is easy to prove that even if $\mathcal{I}$ is allowed to depend on w_i in an *arbitrary* way, no voter model protocol solves the DPP.)

Therefore, the logical next thing to consider in order to solve the DPP is to allow $\mathcal{F}$ to in addition depend on w_i. And this leads us to the natural extension of the classic voter model: the *biased voter model*.

4 The Biased Voter Model

Discussion from the previous section reveals that in order to solve the DPP, it is necessarily to allow $\mathcal{F}$ to depend on w_i in addition to f_i, so that how an individual changes his opinion is influenced by his neighbors as well as his own intrinsic preferences. A natural class of $\mathcal{F}$ that reflect an individual's preference (or *bias*) are those that let him assume his preferred opinion α with probability higher than $f_i(\alpha)$, which is the probability he assumes opinion α in the voter model. We call the resulting model the *biased voter model* and define it formally as follows.

Definition 2 (Biased Voter Model). *The biased voter model is a class of protocols of the form* $(\mathcal{I}, \mathcal{F})$ *where for some constant* $\epsilon > 0$*,*

$$P\{\mathcal{F}(f_i, w_i) = \alpha\} \begin{cases} \geq \min\{f_i(\alpha) + \epsilon, 1\} & \text{if } w_i(\alpha) > 1/2; \\ \leq \max\{f_i(\alpha) - \epsilon, 0\} & \text{otherwise.} \end{cases}$$

and $\mathcal{I}$ *is allowed to depend on* w_i *in an arbitrary way.*

Definition 2 is a generic one which only defines biased updating function $\mathcal{F}$ qualitatively without specifying how exactly it is computed. A natural choice is where each agent plays α with probability proportional to the product $f_i(\alpha)w_i(\alpha)$ [8]. In this model an agent balances their preferences with the behavior of their neighbors in a simple multiplicative fashion and we call this the *multiplicative* biased voter model.

We note the extension to the biased voter model in Definition 2 is fairly general in that $\mathcal{F}$ is allowed to include a broad class of local stochastic updating rules that reflect a node's preferences; and $\mathcal{I}$ is allowed to be *arbitrary* although it has to be independent of G. These seemingly provide us with a lot of power in the design of protocols; but perhaps surprisingly, in this section we prove that even this broad class of biased voting rules is insufficient to solve the DPP:

Theorem 3. *No biased voter model protocol* $(\mathcal{I}, \mathcal{F})$ *solves the DPP.*

The rest of this section is organized as follows. In Section 4.1, we prove a technical lemma about a certain Markov chain that can be represented by a line graph. We then use this lemma to prove Theorem 3 in 4.2, by constructing an example where for any voter model protocol $(\mathcal{I}, \mathcal{F})$, it either takes exponential time to converge to the globally preferred color, or convergence is to the globally non-preferred color, both violations of the DPP requirements.

4.1 A Markov Chain Lemma

Consider a Markov Chain on a line graph of n nodes, namely $s_1, s_2, ..., s_n$, where transition does not happen beyond adjacent nodes. In this subsection we want to show that if at any state s_i $(1 < i < n)$, the Markov chain is more likely to go 'backward' to state s_{i+1} than to go 'forward' to state s_{i-1}, then starting from state s_i (where $i \geq 2$), it takes exponential time in expectation to hit state s_1. While this is perhaps intuitive, we will need this result to be in a particular form for the later reduction.

Here are a couple of notations: Let $p_{i,j}$ $(i, j \in [n])$ be the transition probability from node i to j, by construction $p_{i,j} = 0$ if $|i - j| > 1$. Simplify notation by writing $p_i = p_{i,i-1}$ and $q_i = p_{i,i+1}$, which are the 'forward' and 'backward' transition probability, respectively. Define h_i to be the expected number of rounds for the process to hit state s_1 for the first time, given that it starts from state s_i. Let $\gamma_{max} = \max_{i \in \{2,3,...,n-1\}} \frac{q_i}{p_i}$ and $\gamma_{min} = \min_{i \in \{2,3,...,n-1\}} \frac{q_i}{p_i}$, we have the following lemma.

Lemma 1. *If $\gamma_{min} \geq 1+\epsilon$ for some constant $\epsilon > 0$, then h_i ($i \geq 2$) is exponential in n.*

Proof. We first claim that $h_i - h_{i-1} > \frac{\gamma_{min}^{n-i}}{p_n}$. To prove this claim, note h_i satisfies the following linear system

$$h_i = \begin{cases} 0 & (i=0) \\ 1 + p_i h_{i-1} + q_i h_{i+1} + (1-p_i-q_i)h_i & (2 \leq i \leq n-1) \\ 1 + p_n h_{n-1} + (1-p_n)h_n & (i=n) \end{cases}$$

It is clear $h_j - h_{j-1} > 0$ for all $j > 1$ as a process starting from state s_j has to hit s_{j-1} before hitting s_1. Let $h_j - h_{j-1} = \lambda_j$, combining it with $h_{j-1} = 1 + p_{j-1}h_{j-2} + q_{j-1}h_j + (1-p_{j-1}-q_{j-1})h_{j-1}$ gives $h_{j-1} - h_{j-2} = \frac{1+q_{j-1}\lambda_j}{p_{j-1}}$, which in turn implies $\lambda_{j-1} > \left(\frac{q_{j-1}}{p_{j-1}}\right)\lambda_j > \gamma_{min}\lambda_j$. Repeating this inductively gives $\lambda_i > \gamma_{min}^{n-i}\lambda_n$. Since $\lambda_n = \frac{1}{p_n}$, this proves the claim.

Immediately following from this claim, we have $h_2 = h_2 - h_1 > \frac{\gamma_{min}^{n-2}}{p_n} \geq (1+\epsilon)^{n-2}$ if $\gamma_{min} \geq 1+\epsilon$. Since $h_i > h_2$ whenever $i > 2$, this completes the proof. □

4.2 The Impossibility Result

Our goal in this subsection is to prove Theorem 3. To this end, first consider the biased voter model on the following 3-regular line graph.

A Line Graph. *G is a line graph of $2n$ nodes, where the left half prefers* blue *and the right half prefers* red. *The leftmost and rightmost node each has two self-loops and all the other nodes have one self-loop.*

We prove two lemmas (Lemma 2 and 3) about this particular setting, as a preparation for the proof of the main theorem.

Lemma 2. *For any biased voter model protocol $(\mathcal{I}, \mathcal{F})$, given that $\mathcal{I}$ results in an initialization where all nodes initialized to* blue *are to the left of all nodes initialized to* red, *it takes exponential time in expectation to reach a consensus on the line.*

Proof. We prove this by reducing this stochastic process to the Markov process described in Section 4.1. First observe that since we start with a coloring where all blues are to the left of all reds, this will hold as an invariant throughout the evolution of the whole process and the only way for the coloring to evolve is for the *blue* node adjacent to a *red* neighbor to convert to *red*, or for its *red* neighbor to convert to *blue*.

Therefore, we can always describe the state by a pair of integers $(b, 2n-b)$, where b is the number of blue-colored nodes. Now if we lump two states, $(b, 2n-b)$ and $(2n-b, b)$, into one, this model is exactly the Markov process (with $n+1$ states) described in Section 4.1 with $s_i = \{(i, 2n-i), (2n-i, i)\}$ for $i = \{0, 1, ..., n\}$.

And by definition of *biased voter model* and the way the Markov chain is constructed in the above, we have $p_i \leq \frac{1/3-\epsilon}{2n}$ and $q_i \geq \frac{1/3+\epsilon}{2n}$ ($i \in \{1, 2, ..., n-1\}$) for some constant $\epsilon > 0$. Therefore, $\gamma_{min} \geq \frac{1/3+\epsilon}{1/3-\epsilon} = 1+\delta$, for some constant $\delta > 0$. Invoking Lemma 1 shows that it takes exponential time to hit s_0 starting from state s_i (where $i \geq 1$). Therefore, it takes exponential time to reach a consensus given that one starts with a coloring where all nodes initialized to *blue* are to the left of all nodes initialized to *red*. □

Lemma 3. *For any biased voter model protocol $(\mathcal{I}, \mathcal{F})$, if $\mathcal{I}$ initializes a node i to his preferred opinion with positive probability, then it takes exponential time in expectation to reach a consensus on the line.*

Proof. Note $\mathcal{I}$ is independent of G, therefore whenever it initializes with positive probability, the probability is independent of n. In particular, the probability that $\mathcal{I}$ initializes the leftmost node to *blue and* the rightmost node to *red* is not negligible. Therefore we are through if we can show that *given* the leftmost node is initialized to *blue* and the rightmost to *red*, it takes exponential time to reach a consensus.

Lemma 2 does not differentiate between a b-consensus and a r-consensus. If we concern ourselves only with the outcome of, say a b-consensus, it can be shown that it still takes exponential time to reach given that we start from the same initialization described in Lemma 2 (i.e. all blues are to the right of all reds). We prove this by first observing that, conditioning on that a b-consensus is reached, the time taken is distributed exactly the same as in the modified stochastic process on the same $2n$-node line graph, with the only difference being making the leftmost node extremely biased towards *blue* so that it always votes for *blue* regardless of his neighbor's opinion. Therefore we only need to prove it takes exponential time for this modified process to reach a consensus (which can only be a *blue* one), and this follows from Lemma 2.

Of course by a similar argument we can show that starting from an initialization where all blues are to the left of all reds, it takes exponential time to reach a r-consensus.

Now consider the initialization where the leftmost node is *blue* and rightmost node is *red* and call this the case of interest. Compare it with the initialization where the leftmost node is *blue* and all the other $n-1$ nodes are *red*, the r-consensus time of this case is clearly upper bounded by that of the case of interest. By the above discussion, it takes exponential time to reach a r-consensus even when we start with only the leftmost node *blue*; therefore, it takes exponential time to reach a r-consensus for the case of interest. By the same argument, it also takes exponential time to reach a b-consensus for the case of interest. In sum this allows us to conclude that it takes exponential time to reach a consensus given that $\mathcal{I}$ initializes the leftmost node to *blue* and the rightmost to *red*. □

We are now ready to give a proof for Theorem 3.

Proof. (of Theorem 3) In Lemma 3, we have already shown that any biased voter model protocol $(\mathcal{I}, \mathcal{F})$ fails to solve the DPP (taking exponential time to converge) if we restrict $\mathcal{I}$ to the kind of initialization functions that initializes a node to its preferred opinion with positive probability. It is easy to see that $(\mathcal{I}, \mathcal{F})$ also fails for any $\mathcal{I}$ that does the opposite, in which case $\mathcal{I}$ initializes a node to his not-preferred opinion with probability 1: Simply construct a graph consists solely of nodes that prefer *blue*, and $\mathcal{I}$ initializes it to a r-consensus. Therefore, we conclude that *any* biased voter model protocol $(\mathcal{I}, \mathcal{F})$ fails to solve the DPP. □

Note since the line graph we construct above is 3-regular, we have actually shown a stronger version of Theorem 3: Even if we allow both $\mathcal{I}$ and $\mathcal{F}$ to depend on d_i, no protocol $(\mathcal{I}, \mathcal{F})$ can solve the DPP. We also note that a similar exponential convergence result can be shown for clique in certain settings.

5 A Protocol for DPP

Previous discussions establish the limitation of the classic voter model protocol and its natural extension to the biased voter model when it comes to solving DPP. We are thus interested in the question: What are the (ideally minimal) extensions to the biased voter model that are needed to obtain a simple, efficient and local protocol for solving the DPP?

In this section, we give one answer to this question, by providing a provable solution to the DPP that employs the following extensions:

1. Introduction of a third choice of opinion, *undecided*, or u for short;
2. Allowing initialization and evolution of a node's opinion be dependent on its degree in G;
3. Allowing multiple identical copies of the protocol to be run in G and having each node vote for the opinion (between *blue* and *red* only, and ignoring *undecided*) converged to more frequently among the multiple runs. This can be implemented by having a slightly more powerful schedule that after every n^5 steps, re-initializes each node.

It is interesting to note that at least two of these extensions — the ability to temporarily declare oneself undecided, and the notion of an election that is run in multiple phases — have obvious analogues in many actual political processes. In any case, we would argue that our protocol is natural in the sense that it obviously does not engage in any of the kind of coding or signalling schemes mentioned in the Introduction.

We give the protocol in Algorithm 1. This protocol consists of $T = poly(n)$ phases. In each phase, each node simultaneously and independently initializes his opinion to either b, r or u according to some probabilities before

Algorithm 1 A Simple and Local Voting Protocol

Each node i maintains an array R_i of size T
for $phase = 1$ to T **do**
  Each node i simultaneously and independently initializes its color to b, r and u with probability $\frac{w_i(b)}{d_i}$, $\frac{w_i(r)}{d_i}$ and $\frac{d_i-1}{d_i}$, respectively;
  Run the (standard) *voter model* process (on opinion b, r, u) for n^5 rounds
  Each node i records his last round opinion of this phase of the voter model process in $R_i[phase]$
end for
`// Each node now has a record of his local 'outcomes' of all` T `phases`
Each node i ignores all entries in R_i that record u; among the remaining entries, identify a majority between b and r, breaking ties arbitrarily
Each node i vote for this majority identified as his *final vote*

launching into the standard unbiased voter model process. These initialization probabilities are properly chosen so that the probability of reaching an α-consensus ($\alpha \in \{b, r\}$) is proportional to $\sum_{i \in V} w_i(\alpha)$. The introduction of the *undecided* opinion u is to allow individuals of high degree deliberately reduce their potentially strong influence on the outcome (We note that in a strategic or game-theoretic setting, high-degree individuals might of course exactly wish to exploit this influence, a topic we examine in Section 6). At the end of each phase, the standard unbiased voter model process is run for n^5 rounds and each node records his opinion in the last round as the 'outcome' of this phase. After T phases, each node ignores all phases where his local outcome is *undecided* and identify the majority between *blue* and *red* among the remaining outcomes; he then vote for this majority as his *final vote*.

We now proceed to prove that Algorithm 1 indeed solves the DPP. First we need the following lemma.

Lemma 4. *In each of the T phases of Algorithm 1, with error exponentially small in n, the probability of reaching a b-consensus, r-consensus and u-consensus are $\frac{\sum_{i \in V} w_i(b)}{2m}$, $\frac{\sum_{i \in V} w_i(r)}{2m}$ and $\frac{2m-n}{2m}$, respectively.*

Proof. We give proof for the case of a b-consensus, and the proof for r-consensus and u-consensus follows a similar argument.

Let $B \subseteq V$ be the set of nodes initialized to *blue*, and $P_b = \sum_{B \in 2^V} p(B)p(b \mid B)$ the probability that a single phase of Algorithm 1 results in a b-consensus. By Theorem 2 we have $\left|p(b \mid B) - \pi(B)\right| = o(c^{-poly(n)})$ for some constant c, or $p(b \mid B) = \pi(B) \pm o(c^{-poly(n)})$, therefore $P_b = \sum_{B \in 2^V} p(B)p(b \mid B) = \sum_{B \in 2^V} p(B)(\pi(B) \pm o(c^{-poly(n)})) = \sum_{B \in 2^V} p(B)\pi(B) \pm o(c^{-poly(n)}) = \sum_{k=0}^{2m} p(d(B) = k)\,(k/2m) \pm o(c^{-poly(n)}) = \frac{E(d_B)}{2m} \pm o(c^{-poly(n)}) = \frac{\sum_{i \in V} w_i(b)}{2m} \pm o(c^{-poly(n)})$. Therefore, we conclude that $\left|P_b - \frac{\sum_{i \in V} w_i(b)}{2m}\right|$ is negligible. □

Recall our goal is to let V converge to the collectively preferred consensus. And by definition of DPP one opinion is significantly preferred than the other, i.e. $|\sum_{i \in V} w_i(b) - \sum_{i \in V} w_i(r)| \geq \epsilon$ for some constant ϵ; this assumption turns out to be sufficient for Algorithm 1 to achieve this goal if we set $T = poly(n)$ sufficiently large.

Theorem 4. *Setting $T = O(n^3 \log(n))$ in Algorithm 1 solves the DPP.*

Proof. By Lemma 4 we have $P_b \geq \frac{\sum_{i \in V} w_i(b)}{2m} - o(c^{-poly(n)})$ and $P_r \leq \frac{\sum_{i \in V} w_i(r)}{2m} + o(c^{-poly(n)})$. Therefore the gap between P_b and P_r is at least $\frac{\epsilon}{2m} - o(c^{-poly(n)})$, so there exists a positive constant $\delta < \epsilon$ such that the gap between P_b and P_r is at least $\frac{\delta}{2m}$ whenever n is sufficiently large.

Let T_b and T_r be the number of b-consensuses and r-consensuses among the T trials, the bad event happens when $T_b < T_r$. For this bad event to happen, either event $T_b < \left(P_b - \frac{1}{3} \cdot \frac{\delta}{2m}\right) T$ or event $T_r > \left(P_r + \frac{1}{3} \cdot \frac{\delta}{2m}\right) T$ has to happen. Since $\frac{n}{4m} < P_b < \frac{n}{2m}$ and $P_r < \frac{n}{4m}$, by applying Chernoff bound, it can be shown that $T = O(n^3 \log(n))$ is sufficiently large to guarantee that both of the above two cases happen with negligible probability. □

Before closing this section, we note that there is an alternative protocol that is a natural variant of Algorithm 1. In this variation, we do not need to introduce the undecided opinion u, instead we make the degree of G, $d(G) = \max_{i \in V} d_i$, an input to $\mathcal{F}$. Now that each node is aware of $d(G)$, he can increase his own influence by (conceptually) adding $d(G) - d_i$ self-loops. When each node does so, the graph becomes regular and we can now simply have each node initialize to opinion $\alpha \in \{b, r\}$ with probability $w_i(\alpha)$ and then run the voter model protocol. Using essentially the same analysis it can be shown that this alternative protocol also solves the DPP if repeated sufficiently many times.

6 An ϵ-Nash Protocol for Democratic Primary Game

Our protocol for solving DPP assumes that each individual will actually follow the protocol honestly. However in a strategic setting, an individual may have incentives to deviate from the prescribed protocol. For example, a node i who prefers *blue* may deviate from Algorithm 1 in a way that increases the chance of reaching a *blue*-consensus, even when this consensus is not collectively preferred.

This naturally leads us to consider the Democratic Primary Game (DPG), which is an extension of DPP to the strategic, or game theoretic, setting. In DPG, a node with preference $(w_i(b), w_i(r))$ receives payoff $w_i(b)$ (resp., $w_i(r)$) if the game results in an unanimous global *blue*-consensus (resp., *red*-consensus) and payoff 0 if no consensus is reached. A solution to DPG is a protocol that solves the DPP (which must be *simple and local* and in polynomial time converge to the collectively preferred consensus with high probability) and at the same time is *strategy-proof*, i.e. each node honestly following the protocol constitutes a Nash equilibrium of the game. We note that DPG may also be viewed as a distributed, networked version of the classic "Battle of the Sexes" game, or as a networked coordination game [7].

In the rest of this section, we show the existence of a protocol that is an ϵ-approximate Nash equilibirum, or ϵ-Nash for short, of DPG. This means although a node can deviate unilaterally from this protocol and increases his expected payoff, the amount of this increase is at most ϵ and we show ϵ is negligible in n and can be made arbitrarily small. To this end, we need to make the following mild assumptions.

1. The removal of any node from G leaves the remaining graph connected. Formally, let G_{-i} be the graph induced by $V \backslash \{i\}$, we assume G_{-i} is connected for all $i \in V$.
2. The exclusion of any node does not change the collectively preferred consensus, and moreover, it still leaves a significant (constant) gap between $\sum_{j \in V(G_{-i})} w_j(b)$ and $\sum_{j \in V(G_{-i})} w_j(r)$.
3. Each node i is identified by a unique ID, $ID(i)$, which is an integer in $\{1, 2, ..., n\}$.

Our ϵ-Nash protocol consists of n runs of the non-Nash protocol Algorithm 1, each on a subgraph G_{-i}. Each run of Algorithm 1 polls the majority opinion of $V \backslash \{i\}$, which by assumption is the same as that of V; however by excluding i from participating, we prevent him from any manipulation of this particular run of the non-Nash protocol. When all the n runs of non-Nash is done, each node ends up with $n-1$ 'polls' and with high probability they should all point to the same collectively preferred consensus. In case it does not, it is strong evidence that some run(s) of the non-Nash protocol had been manipulated and the contingency plan is for each node to ignore all the polling results entirely and toss a (private) fair coin to decide whether to vote for *blue* or *red* — and this turns out to be a sufficient deterrent of unilateral deviation from the non-Nash protocol.

We note conceptually we are making yet another simple extension in the protocol's expressiveness by allowing it to be run on a subgraph G_{-i}. To implement this, it is important for each node i to be uniquely identified by his neighbors so that they know when to ignore i; and this is the reason we need assumption 3 listed above. We give this ϵ-Nash protocol in Algorithm 2 and claim the following theorem.

Theorem 5. *Algorithm 2 approximately solves DPG by being an ϵ-Nash equilibrium.*

Proof. Suppose each node follows the protocol faithfully, by our assumption that the exclusion of any node does not change the collectively preferred consensus, say *blue*, the n runs of Algorithm 1 must have all resulted

Algorithm 2 A Simple and Local Protocol that is ϵ-Nash

Each node i maintains an array E_i of size $n-1$
for *episode* $= 1$ to n **do**
  Let i be the node such that $ID(i) = episode$
  Run Line 1 - 8 of Algorithm 1 on G_{-i}
  Each node $j \in V \backslash \{i\}$ records in $E_j[episode]$ the majority between b and r he identifies on Line 8 of (this run of) Algorithm 1
end for
`// Each node has now participated in` $n-1$ `runs of Algorithm 1`
for all $i \in V$ **do**
  if both b and r are present in the $n-1$ entries of E_i **then**
    Tossing a private fair coin to decide between b and r, and vote for it as i's *final vote*
  else
    Vote for the only opinion present as i's *final vote*
  end if
end for

in a b-consensus with high probability. Therefore the final votes result in the collectively preferred b-consensus with high probability.

Now we examine why faithfully executing this protocol is an ϵ-Nash strategy for each node. For a node i that prefers *red* (i.e. the opinion not collectively preferred), assuming everyone else is following Algorithm 2, the expected payoff to i for doing the same is at least his payoff in a b-consensus minus a number negligible in n (because there is a negligible probability that no consensus is reached in the final voting even if every node follows Algorithm 2 faithfully). Now we consider what happens if he deviates. There are two stages during which i can deviate: the first or the second for-loop in Algorithm 2. i's effort during the first for-loop is obviously futile if none of the $n-1$ runs of Algorithm 1 is turned into a r-consensus, and in this case, with high probability all the n runs of Algorithm 1 result in a b-consensus. Therefore, i will have no incentive to deviate during the second for-loop because everyone else is going to vote for b.

Next consider the case where i successfully turns some of the global outcomes of Algorithm 1 into a r-consensus (i.e. all nodes identify r as the majority on Line 8 of Algorithm 1), then with high probability the n runs of Algorithm 1 result in both r-consensus and b-consensus because the single run of it without i participating results in a b-consensus with high probability. In this case, at least $n-2$ nodes out of $V \backslash \{i\}$ see both *blue* and *red* as outcomes from the $n-1$ runs of Algorithm 1 they each participated in and will vote for either b or r by tossing a private fair coin, which means the probability of reaching a b-consensus or r-consensus among them, independent of whatever strategy i adopts in the second for-loop, is $(\frac{1}{2})^{(n-2)}$. Therefore, no matter what strategy i adopts in the second for-loop, his expected payoff is negligible and obviously worse than what he would have gotten by not deviating. Therefore, we conclude that executing Algorithm 2 faithfully is actually a Nash strategy for i.

Now consider a node j who prefers a b-consensus. By the same discussion as above, Algorithm 2 results in a b-consensus in the final voting with high probability, therefore the expected payoff to j is at least his payoff in a b-consensus minus a negligible number. Therefore by deviating j can only hope to improve his expected payoff by a negligible amount. And this allows us to conclude that each node following Algorithm 2 faithfully constitutes an ϵ-Nash equilibrium for the game, where ϵ is a negligible number and can be made arbitrarily small. □

Finally, we note that it is possible for one to construct a distributed protocol that is a Nash equilibrium for DPG, by employing cryptographic techniques developed for distributed computation in a recent work by Kearns et al. [8]. Although the resulting protocol is highly distributed and uses only local information, its use of cryptographic tools, including the broadcast of public keys and secure multiparty function computations, violates our goal of finding simple protocols of the kind we have examined here.

Acknowledgement. We thank Tanmoy Chakraborty for helpful conversations on the work presented here.

References

1. D. Aldous and J. Fill. Reversible Markov Chains and Random Walks on Graphs. *Draft*, 1994.
2. P. Clifford and A. Sudbury. A model for spatial conflict. Biometrika, Vol. 60(3), 1973, 581C588.
3. E. Even-Dar and A. Shapira. A Note on Maximizing the Spread of Influence in Social Networks. In Proceedings of *WINE'07*, 2007.
4. M. Granovetter. Threshold Models of Collective Behavior. In *American Journal of Sociology*, Vol. 83(6), 1978, 1420C1443.
5. R.A. Holley and T.M. Liggett. Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model. Annals of Probability, Vol. 3, 1975, 643C663.
6. L. Lovász. Random Walks on Graphs: A Survey. Combinatorics, Paul Erdös is Eighty, Vol. 2 Jänos Bolyai Mathematical Society, Budapest, 1996, 353-398.
7. R. Luce and H. Raiffa. Games and Decisions. *Wiley*, 1957.
8. M. Kearns, J. Tan and J. Wortman, Network-Faithful Secure Computation. preprint, 2007.
9. M. Kearns and J. Wortman. Learning from Collective Behavior. In Proceedings of *COLT'08*, 2008.
10. J. Kleinberg. Cascading Behavior in Networks: Algorithmic and Economic Issues. In *Algorithmic Game Theory*. N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani eds., *Cambridge University Press*, 2007, 613C632.
11. R. Motwani and P. Raghavan. Randomized Algorithms. *Cambridge University Press*, 1996.
12. T. Schelling. Micromotives and Macrobehavior. *W.W. Norton*, 1978.
13. J. Zeleny. Working Together, Obama and Clinton Try to Show Unity. New York Times, June 28, 2008. Available online at http://www.nytimes.com/2008/06/28/us/politics/28unity.html.