On the Voting Time of the Deterministic Majority Process*

Dominik Kaaser[1], Frederik Mallmann-Trenn[2], and Emanuele Natale[3]

1 **University of Salzburg, Salzburg, Austria**
dominik@cosy.sbg.ac.at
2 **École normale supérieure, Paris, France, and Simon Fraser University, Burnaby, Canada**
fmallman@sfu.ca
3 **Sapienza Università di Roma, Rome, Italy**
natale@di.uniroma1.it

Abstract

In the deterministic binary majority process we are given a simple graph where each node has one out of two initial opinions. In every round, each node adopts the majority opinion among its neighbors. It is known that this process always converges in $\mathrm{O}(|E|)$ rounds to a two-periodic state in which every node either keeps its opinion or changes it in every round.

It has been shown by Frischknecht, Keller, and Wattenhofer (2013) that the $\mathrm{O}(|E|)$ bound on the convergence time of the deterministic binary majority process is even for dense graphs tight. However, in many graphs such as the complete graph the process converges in just a constant number of rounds from any initial opinion assignment.

We show that it is NP-hard to decide whether there exists an initial opinion assignment for which it takes more than k rounds to converge to the two-periodic state, for a given integer k. We then give a new upper bound on the voting time of the deterministic binary majority process. Our bound can be computed in linear time by carefully exploiting the structure of the potential function by Goles and Olivos. We identify certain modules of a graph G to obtain a new graph G^{Δ}. This new graph G^{Δ} has the property that the worst-case convergence time of G^{Δ} is an upper bound on that of G. Our new bounds asymptotically improve the best known bounds for various graph classes.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases distributed voting, majority rule

Digital Object Identifier 10.4230/LIPIcs.MFCS.2016.55

1 Introduction

Distributed voting is a fundamental problem in distributed computing. We are given a network of players modeled as a graph. Each player in the network starts with one initial opinion out of a set of possible opinions. Then the voting process runs either synchronously in discrete rounds or asynchronously according to some activation mechanism. During these rounds in the synchronous case, or upon activation in the asynchronous case, the players are allowed to communicate with their direct neighbors in the network with the main goal to eventually agree on one of the initial opinions. If all nodes agree on one opinion, we say this

* Research was supported by Austrian Science Fund (FWF): P 27613

© Dominik Kaaser, Frederik Mallmann-Trenn, and Emanuele Natale;
licensed under Creative Commons License CC-BY
41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016).
Editors: Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier; Article No. 55; pp. 55:1–55:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

opinion *wins* and the process *converges*. Usually, voting algorithms are required to be simple, fault-tolerant, and easy to implement [22, 24].

In this paper, we study the *deterministic binary majority process* which is defined as follows. We are given a graph $G = (V, E)$ where each node has one out of two opinions. The process runs synchronously in discrete rounds where each node in every round computes and adopts the majority opinion among all of its neighbors. It is known that this process always converges to a two-periodic state. The *convergence time* of a given graph for a given initial opinion assignment is the time required until this two-periodic state is reached. In this work we improve the bounds on the convergence time for given initial opinions and then we analyze the *voting time* of the process, which is the maximum convergence time over all possible initial opinion assignments.

In distributed computing, various variants of the majority process are used in fault-tolerant distributed consensus algorithms. In the analysis of structures of large networks, the deterministic binary majority process has widespread applications in the study of so-called *influence networks* [15]. Early applications can be found in distributed databases [16]. Further fields include sensor networks [6], the analysis of opinions in social networks [32], social behavior in game theory [12], chemical reaction networks [14], neural and automata networks [18], and cells' behavior in biology [7]. Variants of the deterministic binary majority process have been used in the area of distributed community detection [38, 28, 10]. In this context, the proposed community detection protocols exhibit a convergence time which can be bounded by the voting time of the deterministic binary majority process.

Among its many probabilistic variants that have been previously considered, plenty of work concerns *randomized voting* where in each step every node is allowed to contact a random sample of its neighbors and updates its current opinion according to the majority opinion in that sample [1, 5, 9, 13, 22, 23, 29, 30, 31, 33].

In an algorithmic game theoretic setting, the deterministic binary majority process can be seen as the simplest *discrete preference games* [8]. In this game theoretic perspective, the existence of *monopolies* has been investigated [2]. A monopoly in a graph is a set of nodes which start with the same opinion and cause all other nodes to eventually adopt this opinion. In the distributed computing area, a lot of research has been done to find small monopolies, see for example [34]. It has also been shown that there exist families of graphs with constant-size monopolies [4]. More recently, classes of graphs which do not have small monopolies have been investigated [35].

Many of these results relate to the voting time of the deterministic binary majority process. It was proven independently by Goles and Olivos [20], and Poljak and Sůra [36] with the same potential function argument that the deterministic binary majority process always converges to a two-periodic state. They later (independently) refined and generalized the potential function argument in several directions [17, 19, 21, 37]. Their proof was popularized in the *Puzzled* columns of Communications of the ACM [41, 42]. Recently, the same problem has been studied on infinite graphs w.r.t. a given probability distribution on the initial opinion assignments [3]. In [40], the authors provide a bound on the number of times a node in a given bounded-degree graph changes its opinion. Both [3] and [40] also investigate the probability that in the two-periodic state all nodes hold the same opinion.

As for the maximum time it takes for the process to converge over all initial opinion assignments, Frischknecht et al. [15] note that the potential argument by Goles et al. [20, 36, 42] can be used to prove an $\mathrm{O}(|E|)$ upper bound. They furthermore show that this upper bound is tight in general, by designing a class of graphs in which the deterministic binary majority process takes at least $\Omega\left(|V|^2\right)$ rounds to converge from a given initial opinion

assignment. This construction has later been extended to prove lower bounds for weighted and multi-edges graphs by Keller et al. [27].

Once the process converges to the two-periodic state, each node stays either with its own opinion or changes its opinion in every round. A lot of attention has been given to the opinions to which the deterministic binary majority process converges. However, regarding the voting time, besides the $\mathrm{O}(|E|)$ upper bound that follows from the result by Goles et al. [20, 36, 42], no further upper bound on the voting time that holds for any initial opinion assignment has been proved. Still, one can observe that in many graphs the voting time is much smaller than $\mathrm{O}(|E|)$. For example, the voting time of the complete graph is one.

We show that for the deterministic binary majority process the question whether the voting time is greater than a given number is NP-hard. While for many generalizations of the deterministic binary majority process many decision problems are known to be NP-hard, at the best of our knowledge this is the first NP-hardness proof that does not require any additional mechanisms besides the bare majority rule of the deterministic binary majority process. However, as we show in the rest of the paper, it is possible to obtain upper bounds on the voting time which can be computed in linear time. A module of a graph is a subset of vertices S such that for each pair of nodes $u, v \in S$ it holds that $N(u) \setminus S = N(v) \setminus S$, where $N(u)$ denotes the set of neighbors of a node u. By carefully exploiting the structure of the potential function by Goles et al. we leverage the particular behavior that certain modules, which we call *families*, exhibit and prove that the voting time of a graph can be bounded by the voting time of a smaller graph that can be constructed in linear time by contracting suitable vertices.

We obtain a new upper bound that asymptotically improves the previous $\mathrm{O}(|E|)$ bound on graph classes which are characterized by a high number of modules that are either cliques or independent sets. An example for such graphs is the Turán graph $T(n, r)$, formed by partitioning a set of n vertices into r subsets of (almost) equal sizes and connecting two vertices by an edge whenever they belong to different subsets. For the convergence time of the Turán graph $T(n, r)$ we obtain an $\mathrm{O}\left(r^2\right)$ bound, compared to the previously best known bound of $\mathrm{O}\left(n^2\right)$. Also, for the convergence time of full d-ary trees we get an $\mathrm{O}(|V|/d)$ bound, compared to $\mathrm{O}(|V|)$ originating from the $\mathrm{O}(|E|)$ bounds. Further examples include the clique and the star graph, for which our bound gives a constant $\mathrm{O}(1)$ convergence time. Our bound relies on a well-known graph contraction technique based on identifying *equivalent nodes*. This technique is used in other related disciplines as well, including parallel and distributed computing. See, for example, the notion of *identical nodes* in the work by Sarıyüce et al. [39].

1.1 Preliminaries

We are given a graph $G = (V, E)$ and an initial opinion assignment defined as follows.

▶ **Definition 1.** An opinion assignment f_t in round $t \geq 0$ is a function $f_t : V \to \{0, 1\}$ which assigns for each $v \in V$ one out of two possible opinions. We will also denote opinion 1 as *white* and opinion 0 as *black*. The opinion assignment at time $t = 0$ is called *initial opinion assignment*.

The deterministic binary majority process can be defined as follows. Let v be an arbitrary but fixed vertex and $N(v)$ the set of neighbors of v. To compute $f_{t+1}(v)$ the node v computes the majority opinion of all of its neighbors in $N(v)$. In the case of a tie the node behaves *lazily*, that is, v stays with its own opinion. Otherwise, there is a *clear majority* and the node adopts the majority opinion. This leads to the following definition.

▶ **Definition 2.** Let $G = (V, E)$ be a graph and let f_0 be an initial opinion assignment such that $f_0 : V \to \{0, 1\}$. The deterministic binary majority process is the series of opinion assignments that satisfy the rule

$$f_{t+1}(v) = \begin{cases} 0 & \text{if } |\{u \in N(v) : f_t(u) = 0\}| > |\{u \in N(v) : f_t(u) = 1\}| \\ 1 & \text{if } |\{u \in N(v) : f_t(u) = 0\}| < |\{u \in N(v) : f_t(u) = 1\}| \\ f_t(v) & \text{otherwise.} \end{cases}$$

Note that the pair (G, f_0) completely determines the behavior of the system according to the majority process. We now define the main object of this work, the voting time.

▶ **Definition 3.** Given a graph $G = (V, E)$ and any initial opinion assignment f_0 on V, the *convergence time* $\mathfrak{T}$ of the majority process on G w.r.t. f_0 is $\mathfrak{T} = \mathfrak{T}(G, f_0) = \min\{t : \forall v \ f_{t+2}(v) = f_t(v)\}$. The *voting time* of G is defined as $\max\limits_{f_0 \in \{0,1\}^V} \mathfrak{T}(G, f_0)$.

Observe that $\mathfrak{T}$ is indeed the number of steps until the process converges to a two-periodic state. This holds since the process is completely determined by the current opinion assignment. Thus $f_{t+2}(v) = f_t(v)$ also implies that $f_{t+3}(v) = f_{t+1}(v)$ for all nodes v.

In the following we assume without loss of generality that G is connected. For disconnected graphs the deterministic binary majority process runs independently in each connected component. Therefore, the resulting upper bounds on the voting time can be replaced by the maximum over the corresponding bounds in the individual connected components of G.

1.2 Our Contribution

First we define the *voting time decision problem* VTDP and show that it is NP-complete.

▶ **Definition 4** (voting time decision problem VTDP)**.** For a given graph G and an integer k, is there an assignment of initial opinions such that the voting time of G is at least k?

▶ **Theorem 5.** *Given a general simple graph G,* VTDP *is NP-complete.*

In Section 3 we extend known approaches to derive upper bounds on the voting time, which are tight for general graphs. In Section 3.2, we identify the following subsets of nodes that play a crucial role in determining the voting time of the deterministic binary majority process.

▶ **Definition 6.** A set of nodes S is called a *family* if and only if for all pairs of nodes $u, v \in S$ we have $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. We say that a family S is *proper* if $|S| > 1$.

The set of families of a graph forms a partition of the nodes into equivalence classes. Our main contribution is a proof that the voting time of the deterministic binary majority process is bounded by that of a new graph obtained by contracting its families into one or two nodes, as stated in the following theorem.

▶ **Definition 7.** Given a graph $G = (V, E)$, its *asymmetric graph* $G^\Delta = (V^\Delta, E^\Delta)$ is the subgraph of G induced by the subset $V^\Delta \subseteq V$ constructed by contracting every family of odd-degree non-adjacent nodes to one node, and any other proper family to two nodes.

Let in the following V_{even} be the set of even-degree vertices in V and, analogously, let V_{odd} be the set of odd-degree vertices. Based on above definition of G^Δ, we give the following bound on the voting time.

▶ **Theorem 8.** *Given any initial opinion assignment on a graph $G = (V, E)$, the voting time of the deterministic binary majority process is at most*

$$1 + \min\left\{|E^{\Delta}| - \frac{|V^{\Delta}_{odd}|}{2}, \frac{|E^{\Delta}|}{2} + \frac{|V^{\Delta}_{even}|}{4} + \frac{7}{4} \cdot |V^{\Delta}|\right\} .$$

Furthermore, this bound can be computed in $\mathrm{O}(|E|)$ *time.*

As mentioned before, this bound becomes $\mathrm{O}\left(r^2\right)$ for the Turán graph $T(n, r)$ and $\mathrm{O}(|V|/d)$ for d-ary trees. Finally, in Section 3 we also give some insight into further interesting computational properties of the deterministic binary majority process. For example, we disprove a monotonicity of the convergence time w.r.t. the potential function and argue that the voting time is not, at least straightforwardly, bounded by the diameter of the graph.

2 NP-Completeness

If it was possible to efficiently compute the worst-case voting time, there would have been not much interest in investigating good upper bounds for it. In this section, we show that this is unlikely to be the case. We prove Theorem 5 by reducing 3SAT to the voting time decision problem. Given $\Phi \in$ 3SAT, we construct a graph $G = G(\Phi)$ such that the deterministic binary majority process on G simulates the evaluation of Φ. The graph G consists of h layers where $h = 3 + 4 \cdot n$. The first layer represents an assignment of the variables in Φ, the remaining layers represent Φ and ensure that the assignment of variables in Φ is valid. We will show that if Φ is satisfiable, then there exists an initial assignment of opinions for which the convergence time is exactly $h + 1$. If, however, Φ is not satisfiable, then any assignment of opinions will result in a convergence time strictly less than $h + 1$. We now give the formal proof.

Let $\Phi \in$ 3SAT be a Boolean formula in 3-conjunctive normal form. Let n be the number of variables of Φ. Let m be the number of clauses of Φ. The Boolean formula is of the form $\Phi = (l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge \cdots \wedge (l_{m,1} \vee l_{m,2} \vee l_{m,3})$, where $l_{i,j} \in \{x_1, \overline{x}_1, x_2, \overline{x}_2, \cdots, x_n, \overline{x}_n\}$ is a literal for $1 \leq i \leq m$ and $1 \leq j \leq 3$.

We construct a graph G to simulate the evaluation of Φ as follows. Let $\ell = 10 \cdot (m+n)+1$. The graph consists of several layers. On the first layer, we place so-called literal cliques of size ℓ, and on the layers above we place the gates. In our reduction, we use OR-gates, an AND-gate, and 2/3-gates. Each gate consists of one or several nodes. Additionally, we have two so-called *mega-cliques* K_{white} and K_{black} of size ℓ.

Let g be an arbitrary but fixed gate. We denote a node on a layer below g which does not belong to g but is connected to g as *input node* to g. Additionally, we will denote a node that belongs to g and is connected to another gate on a layer above g as *output node* of g.

In the following, we assume that opinion 1, white, corresponds to Boolean TRUE and 0, black, corresponds to FALSE. The main idea of the construction is to show that an *activation signal* is transmitted from the bottom up through all layers. If the current assignment of opinions on the literal cliques corresponds to a satisfying assignment of Boolean values to Φ, then the process requires $h + 1$ steps. The main purpose of the OR-gates and the AND-gate is to evaluate Φ. The 2/3-gates check whether the opinion assignment to literal nodes is valid. That is, we need to enforce that the corresponding literal nodes for x_i and $\overline{x}_i$ are of opposite colors for every variable x_i of Φ. If either this condition is violated and variables x_i exist for which $x_i = \overline{x}_i$ or the current assignment of opinions on the literal cliques does not corresponds to a satisfying assignment of Boolean values to Φ, the construction enforces that the process stops prematurely after strictly fewer than $h + 1$ steps.

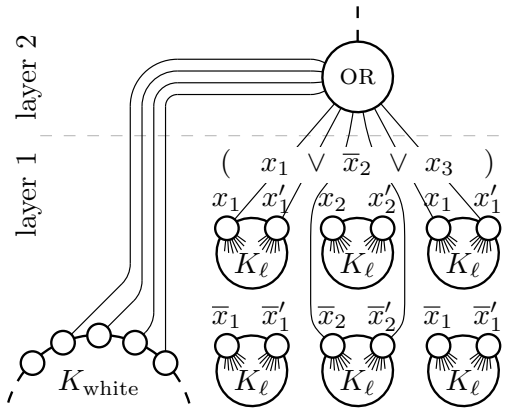

Figure 1 Literal nodes and OR-gate.

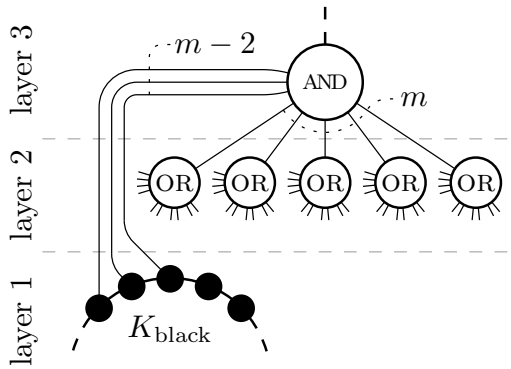

Figure 2 AND-gate.

Layer 1: Literal Cliques. We represent each variable x_i with two cliques, one for x_i and one for $\overline{x}_i$. Each clique has a size of ℓ which is defined above. Note that ℓ is odd. Additionally, we distinguish three so-called *representative nodes* in each of these cliques. Furthermore, we add two cliques of size ℓ to the graph which we call mega-cliques. Intuitively, these mega-cliques represent the Boolean values TRUE and FALSE. We will show that they cannot have the same color in order to achieve a long convergence time. The mega-cliques are used in all other gates.

Layer 2: Parallel OR-Gates. The OR-gates are placed on layer 2 and consist of one node v which is also the output node. There is one OR-gate for every clause. Fix a clause $(l_{j,1} \vee l_{j,2} \vee l_{j,3})$. Input nodes are three pairs of nodes (v_1, v_1'), (v_2, v_2'), and (v_3, v_3'), where (v_1, v_1') are two representative nodes of the literal clique for $l_{j,1}$, (v_2, v_2') are representatives of $l_{j,2}$, and (v_3, v_3') are representatives of $l_{j,3}$. That is, for each literal in the clause we connect the OR-gate on layer 2 to two of the three representative nodes of the corresponding literal clique on layer 1. The output node v is additionally connected to 4 nodes of the K_{white} mega-clique. Intuitively, we use the OR-gates to verify that for each clause at least one literal is TRUE. All clauses are evaluated simultaneously using an OR-gate for each clause. The OR-gate is shown in Figure 1.

Layer 3: AND-Gate. There is exactly one AND-gate on layer 3. This AND-gate consists of one output node denoted u_0, which has the following input nodes. It is connected to every output node of the OR-gates on layer 2 and to $m-2$ distinct nodes of the K_{black} mega-clique. Intuitively, the AND-gate is used toverify that every clause is satisfied. It is shown in Figure 2.

Layers 4 to 3 + 4n: 2/3-Gates. The 2/3-gates consist of a path v_1, v_2, v_3, and v_4. Each node of this path is connected to two distinct nodes of the K_{white}. The output node of the gate is v_4. The node v_1 of the first 2/3-gate on layer 4 is connected to the AND-gate on layer 3. The node v_1 of each of the following 2/3-gates is connected to the node v_4 of the previous 2/3-gate. Additionally, the input node of the i-th 2/3-gate is connected to three distinct nodes of the literal clique representing x_i and to three distinct nodes of the literal clique representing $\overline{x}_i$ on layer 1. The output node of the final 2/3-gate is connected to K_{black}. An example is shown in Figure 3. The 2/3-gates are used to verify that we do not have variables x_i in Φ for which the literal cliques of x_i and $\overline{x}_i$ have the same color. Observe that 2/3-gates span over 4 layers, and we have n such 2/3-gates.

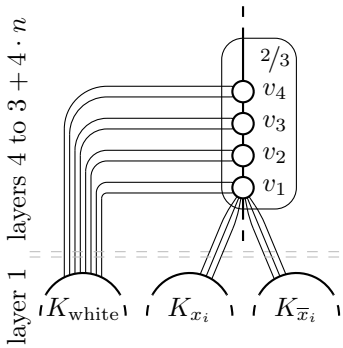

Figure 3 The 2/3-gate.

Literal cliques, OR-gates, and the AND-gate use only one layer, and 2/3-gates span over 4 layers. Therefore, the total number of layers is $h = 3 + 4 \cdot n$, which results from one layer for the literal cliques, one layer for the OR-gates, one layer for the AND-gate, and $4 \cdot n$ layers containing n concatenated 2/3-gates. A detailed example for such a graph G is given in [26]. Based on above description of G we prove the following lemmas, which are then used to show Theorem 5.

▶ **Lemma 9.** *If Φ is satisfiable, then there exists an assignment of opinions such that the convergence time in G is at least $h + 1$.*

To show Theorem 9, we construct an initial opinion assignment for which the gates change from black to white one layer after the other, assuming Φ is satisfiable. The full proof can be found in [26].

It remains to show that if Φ is not satisfiable, then the voting time in G is strictly less than $h + 1$. Recall that the voting time is the maximum of the convergence time over all possible initial opinion assignments.

▶ **Lemma 10.** *If Φ is not satisfiable, then there is no assignment of opinions such that the convergence time in G is at least $h + 1$.*

Before we prove this lemma, we establish several auxiliary lemmas which require the following definitions. Let u_0 denote the output node of the AND-gate. Consider the graph G' induced by the nodes of the AND-gate and the nodes of the 2/3-gates. Let u_i be the node at distance i to u_0 in G'. We observe that G' is a path $u_0, \ldots, u_\kappa$ consisting of the $4 \cdot n + 1$ top layers of the graph G. Consequently, $\kappa = 4 \cdot n$ and u_i is the i-th node on this path.

▶ **Definition 11** (Stable Time)**.** We define the *stable time* $s(v)$ for any node $v \in V$ to be the first time step such that v does not change its opinion in any subsequent time step $t' > s(v)$ over all possible initial configurations. That is,

$$s(v) = \min \left\{ t : \forall f_0 \in \{0,1\}^V \;\; \forall t' \geq t \quad f_{t'}(v) = f_t(v) \right\} \quad .$$

Accordingly, let for any subset $V' \subseteq V$ be $s(V')$ defined as $s(V') = \max \{s(v) : v \in V'\}$.

In the following, let V_K be the set of nodes of all cliques in $G(\Phi)$, that is, the nodes contained in the literal cliques and in the mega-cliques on layer 1. Furthermore, let $V_{K^{\text{R}}}$ be the set of representatives of the cliques and $V_{K^-} = V_K \setminus V_{K^{\text{R}}}$. That is, every clique K on layer 1 consists of $K^- \cup K^{\text{R}}$. Finally, let V_{OR} be the set of all output nodes of OR-gates. The following lemma shows that the layers become stable one after the other.

▶ **Lemma 12.** *It takes at most* 3 *time steps for the layers* 1 *and* 2 *consisting of literal cliques and* OR*-gates to become stable. Precisely, we have* (i) $s(V_{K^-}) = 1$, (ii) $s(V_{K^{\text{R}}}) = 2$, *and* (iii) $s(V_{\text{OR}}) = 3$.

The above lemma gives bounds on the stable time of layers 1 and 2. In the following, we argue that whenever a node changes its opinion in any step t after time step 3, it will not change its color in any subsequent time step $t' \geq t$ any more. We therefore define the so-called *activation time* of a node $v \in G'$ as follows.

▶ **Definition 13** (Activation Time)**.** Let c be the color of the K_{black} mega-clique at time 2 and let f_0 be an arbitrary but fixed initial opinion assignment. We define the *activation time* of a node $v \in G'$ to be the first time step after time step 3 in which the node v adopts opinion c. That is, $a(v) = \min\{t \geq 3 : f_t(v) = c\}$. If v does not change its color after time step 3 we write $a(v) = 3$.

We now use the above definition to state the following lemma, which describes that every node $u_i \in G'$ with $i \geq 1$ changes its color at most once after time step 3. Note that this covers the nodes of the 2/3-gates.

▶ **Lemma 14.** *Let f_0 be an arbitrary but fixed initial opinion assignment. Let t be the activation time w.r.t. f_0 of the node $u_i \in G'$ with $i \geq 1$ such that $t = a(u_i)$. Then for all $t' \geq t$ we have $f_{t'}(u_i) = f_t(u_i)$.*

Proof. By Theorem 12, all nodes $u \in V_{K^{\text{R}}}$ are stable at $t' \geq 2$. We now distinguish two cases.

Case 1: $i \mod 4 \neq 1$. Observe that u_i can only change its color at time $t = a(u_i)$, if it had a different color than K_{white} in the previous round. This holds, since every node u_i with $i \mod 4 \neq 1$ has the same number of connections to K_{white} than to nodes in $V \setminus K_{\text{white}}$. Since furthermore the process behaves lazy, any node u_i which has the same color as K_{white} cannot change its opinion back to the opposite color any more.

Case 2: $i \mod 4 = 1$. The node u_i is a v_1 node of the j-th 2/3-gate with $j = \lceil i/4 \rceil$. Therefore it is connected to three representatives of each literal clique for x_j and $\overline{x}_j$. The literal representatives of x_j and $\overline{x}_j$ are stable at time $t' \geq 2$. Now if x_j and $\overline{x}_j$ have the same color c, then u_i has $6 > |N(u_i)|/2$ edges to nodes of color c. Therefore, the node does not change its color any more after time step 3. That is, we have $a(u_i) = 3$ and also $f_{t'}(u_i) = c$ for any consecutive time step $t' \geq 3$. If, however, x_j and $\overline{x}_j$ do not have the same color, these edge *cancel* each other out and the color of node u_i is determined by u_{i-1}, u_{i+1}, and K_{white}. Therefore, the same argument as in the first case holds. ◀

In the following we examine the behavior of layer 3 which contains only the AND-gate. Recall that u_0 is the output node of the AND-gate. The next lemma describes the following fact. The AND-gate u_0 can only change its color in a time step $t \geq 4$ if u_1 changed its color in time step $t - 1$. After this change at time t, the node u_0 cannot change its color again.

▶ **Lemma 15.** *Let f_0 be an arbitrary but fixed initial opinion assignment and let furthermore t be the round after node u_1 has been activated such that $t = a(u_1) + 1$. For all consecutive rounds $t' \geq t$ we have $f_{t'}(u_0) = f_t(u_0)$. That is, the* AND*-gate does not change its opinion any more once the node u_1 has become stable.*

The following lemma implies that in order to reach a convergence time of $h + 1$ the gates on the path $u_0, \ldots, u_\kappa$ in G' have to activate one after the other starting with u_0 at time 4. Recall that $\kappa = 4 \cdot n$.

▶ **Lemma 16.** *Let f_0 be an arbitrary but fixed initial opinion assignment and let $u_i \in G'$ be a node with $0 \leq i \leq \kappa$. If $a(u_i) < i + 4$ w.r.t. f_0, then $\mathfrak{T}(G(\Phi), f_0) < h + 1$.*

In the following two lemmas, we enforce that initial opinion assignments which do not represent valid assignments of Boolean values to literal cliques result in premature termination of the deterministic binary majority process in $G(\Phi)$. An assignment is called *illegal* if there exist literal cliques such that the majority of x_i and the majority of $\overline{x}_i$ have the same initial color.

▶ **Lemma 17.** *For any illegal initial opinion assignment f_I to $G(\Phi)$, the convergence time $\mathfrak{T}(G(\Phi), f_I)$ is strictly less than $h + 1$.*

▶ **Lemma 18.** *If after two time steps K_{white} and K_{black} have the same color, the process stops after strictly fewer steps than $h + 1$.*

From above lemmas we conclude that Theorem 10 holds, and together with Theorem 9, Theorem 10 yields Theorem 5. The full proofs can be found in [26].

Proof of Theorem 5. It is easy to see that VTDP is in NP. Furthermore, we can polynomially reduce 3SAT to VTDP. The correctness proof of the reduction follows from Theorem 9 and Theorem 10. Therefore we conclude that VTDP is NP-complete. ◀

3 Bounds on the Voting Time

Since the problem is NP hard, we cannot hope to calculate the voting time of a graph efficiently. Nevertheless, in this section we show, that it is possible to obtain non-trivial upper bounds on the voting time that are easy to compute. This section is dedicated to proving our upper bound on the voting time, Theorem 8. The main contribution of this theorem is the influence of symmetry which is studied in Section 3.2.

We start by giving a formal version of the potential function argument [20, 36] as conceived in [42]. In the following we assume that each edge in $\{x, y\} \in E$ can be replaced by two directed edges (x, y) and (y, x). The main idea is based on so-called *bad arrows* defined as follows.

▶ **Definition 19.** Let $G = (V, E)$ be a graph with initial opinion assignment f_0. Let v denote an arbitrary but fixed node and $u \in N(v)$ a neighbor of v. Let t denote an arbitrary but fixed round. The directed edge (v, u) is called *bad arrow* if and only if the opinion of u in round $t + 1$ differs from the opinion of v in round t. We will also denote the bad arrows which have their tail at round $t = 0$ as *initial bad arrows*.

Intuitively, each of these directed edges (v, u) can be seen as *advice* given from v to u in the voting process. In the case of a bad arrow the advice was not followed by u since it has a different opinion in the following round than v. Observe that each bad arrow is incident at exactly two nodes and thus we say it is *outgoing* in the node at its tail and incoming in the node at its head. An example of such a bad arrow can be seen in Figure 4.

▶ **Theorem 20.** *Let $G = (V, E)$ be a graph which contains only vertices of odd degree. The voting time of the deterministic binary majority process on G is at most $1 + W_{bad}$ where W_{bad} is an upper bound on the number of initial bad arrows for any initial opinion assignment on G. In particular, the voting time of G is at most $2 \cdot |E| + 1$.*

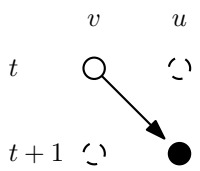

Figure 4 A *bad arrow* from node v to node u in round t.

The idea of the proof is to define a potential function ϕ_t that is strictly monotonically decreasing over the time. Let f_0 be any initial opinion assignment. The potential function ϕ_t is simply the number of bad arrows defined in Theorem 19, that is

$$\phi_t = \phi_t(G, f_t) = |\{(v, u) \in E : f_{t+1}(u) \neq f_t(v)\}| \quad .$$

For the full proof, see [26]. Note that in Theorem 20 it is assumed that all nodes of the graph have odd-degree. In the following we show how to remove this assumption.

▶ **Definition 21.** Let $G = (V, E)$ be a graph. The graph $G^* = (V, E^*)$ is the graph obtained by adding a self loop to every node of even degree in G. More formally,

$$E^* = E \cup \bigcup_{v \in V_{\text{even}}} (v, v) \quad .$$

From the definition it follows that $|E^*| = |E| + |V_{\text{even}}|$.

▶ **Theorem 22.** *The voting time of the deterministic binary majority process on any graph $G = (V, E)$ is at most $1 + W_{bad}$, where W_{bad} is an upper bound on the number of initial bad arrows in G^*.*

The proof is based on the fact that for every node $v \in V$ the sequence of opinions, $(f_t(v))$, is exactly the same for the deterministic binary majority process in G as for the deterministic binary majority process in G^*. For the full proof, see [26].

Observe, that while the number of bad arrows is used in the potential function, the convergence time is, however, not monotone w.r.t. the number of initial bad arrows.

▶ **Lemma 23.** *The convergence time is not monotone w.r.t. the number of initial bad arrows.*

The upper bound on the voting time considered in [27] follows from the $2 \cdot |E|$ upper bound on the number of bad arrows of Theorem 20. Clearly, this result can be improved by a factor of 2 by simply applying the observation that the number of initial bad arrows in G^* is at most $|E| - |V_{odd}|/2$. Therefore, from Theorem 22 we obtain the following corollary.

▶ **Corollary 24.** *The voting time of the deterministic binary majority process on any graph $G = (V, E)$ is at most $1 + |E| - |V_{odd}|/2$.*

▶ Remark. Theorem 24 is tight for general graphs up to an additive constant of 1. Indeed, consider a path with an initial opinion assignment on which the opinions alternate except for the last two nodes, which share the same opinion.

Suppose that, instead of specifying the initial opinion assignment, we decide in advance what bad arrows are there. We can do that by deciding for each ordered pair (u, v) for which $\{u, v\} \in E$ whether we want to have a bad arrow going from u to v. We formalize this notion by means of the following definitions.

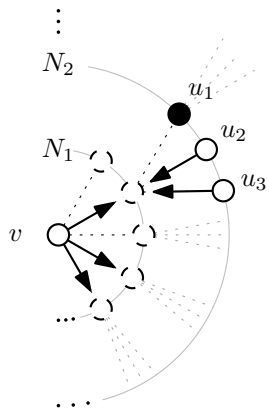

Figure 5 The opinions of each second neighborhood are uniquely determined.

▶ **Definition 25.** Let $G = (V, E)$ be a graph and $\beta : V \times V \to \{0, 1\}$ denote a characteristic function on $V \times V$. Then β is a bad arrows assignment on G if there exists an opinion assignment f on G that determines β such that β is the indicator function of the bad arrows we have on G w.r.t. the opinion assignment f.

In proving upper bounds on the voting time we consider the bad arrows assignment determined by the initial opinion assignment. One may wonder whether in doing so we are *losing information*. In the following lemma we show that, given a valid bad arrows assignment, we can reconstruct the initial opinion assignment up to exchanging black and white (and up to two more possibilities in bipartite graphs).

▶ **Lemma 26.** *Let G be a connected graph and let β be a valid bad arrows assignment on G. If the graph is not bipartite, there are exactly two opinion assignments, otherwise there are exactly four opinion assignments that determine β.*

Proof. Let $v \in V$ denote an arbitrary but fixed vertex. We now denote the set $\{v\}$ as N_0 and the set of direct neighbors of v as N_1 to define the i-th neighborhood N_i for $i \geq 2$ as

$$N_i = \left(\bigcup_{u \in N_{i-1}} N(u) \right) \setminus \left(\bigcup_{j=1}^{i-1} N_j \right) .$$

We now show by an induction on $k = 0, 1, 2, \ldots$ that the colors of all nodes in $N_{2 \cdot k}$ are determined by the color of v. The base-case is trivial since for $k = 0$ we have $N_0 = \{v\}$. For the induction step we observe that according to the induction hypothesis the color of each node in $N_{2 \cdot k}$ is determined. We now observe that the color at time 1 of each node in $N_{2 \cdot k+1}$ is determined by β and the colors at time 0 of the nodes in $N_{2 \cdot k}$. Vice versa, also the colors at time 0 of nodes in $N_{2 \cdot (k+1)}$ are determined by β and the colors at time 1 of each node in $N_{2 \cdot k+1}$. This concludes the induction.

An example is shown in Figure 5. In this figure it is clear that v and, e.g., u_1 must have a different color, for the following reason. Since u_1 does not have a bad arrow to its neighbor in N_1, it has the same color in the next round as this neighbor. But this neighbor's color in the next round is different to the current color of v because of the bad arrow assignment.

Observe that from above induction the lemma follows immediately for bipartite graphs. We can fix the colors for two arbitrary nodes, one from each of the two sets of non-adjacent nodes, to determine all other nodes' colors. This gives us four possible opinion assignments

for a given bad arrow assignment β. If the graph is not bipartite there must exist a cycle of odd length. The opinion assignments for all nodes of this cycle are determined by β with the same argument as in above induction. Therefore, not only the colors of even neighborhoods N_{2k} are determined, but also of odd neighborhoods N_{2k+1}. This leaves us with exactly two possible initial opinion assignments, which concludes the proof. ◀

3.1 Improved Bounds for Dense Graphs

We observe that Theorem 24 is (almost) tight, and it gives us a voting time linear in the number of vertices for sparse graphs where $|E| = \mathrm{O}(|V|)$. However, for dense graphs with, e.g., $|E| = \Omega\left(|V|^2\right)$ there is room for improvement. Now the main goal in this following subsection is to reduce the dominant term of the voting time even further, which leads us to the following theorem which is formally shown in [26].

▶ **Theorem 27.** *Let $G = (V, E)$ be a graph. For any initial opinion assignment f_0 on G, the convergence time of the deterministic binary majority process is at most* $1 + \frac{|E|}{2} + \frac{|V_{\text{even}}|}{4} + \frac{7}{4} \cdot |V|$.

▶ Remark. One might intuitively assume that the voting time is bounded by the diameter of the network. However, this is not true, at least straightforwardly, as there exist graphs G where the convergence time w.r.t. a given initial opinion assignments f_0 is asymptotically larger than the diameter of the network, that is, $\mathfrak{T}(G, f_0) \gg \mathrm{diam}(G)$.

3.2 The Influence of Symmetry

We observe that the majority process is much faster on graphs that exhibit certain types of symmetry, such as the star graph, the complete graph and many other graphs in which several nodes share a common neighborhood. We investigate this feature of the process to further improve the bounds obtained so far. We recall that a set of nodes S is called a *family* if and only if for all nodes $u, v \in S$ we have $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. The key fact is that these nodes of any family will behave in a similar way after the first step.

▶ **Definition 28.** Let $\mathrm{fam}(u)$ denote the family of u. We write $u \sim v$ if $\mathrm{fam}(u) = \mathrm{fam}(v)$.

▶ **Lemma 29.** *The relation $\sim$ defines an equivalence class. In particular, all nodes in the same family either form a clique or a stable set, and they all have the same degree in G.*

▶ **Corollary 30.** *For any graph G, its asymmetric graph G^{Δ} is well-defined.*

Proof. According to Theorem 29, the set of families is a partition of the nodes of G. By construction of G^{Δ}, every family S in G is replaced by one or two nodes in G^{Δ}. Therefore, there is a bijection between the families in G and the corresponding node or pair of nodes in G^{Δ}. Hence G^{Δ} is well-defined. ◀

We are now ready to prove Theorem 8.

▶ Remark. While we show that for the voting time we have $\max_f \mathfrak{T}(G^{\Delta}, f) \geq \max_f \mathfrak{T}(G, f)$, in general it is not the case that $\mathfrak{T}(G^{\Delta}, f) \geq \mathfrak{T}(G, f)$ for every opinion assignment f. A formal statement along with a counterexample is given in [26].

Proof of Theorem 8. Let v and v' be two nodes of the same family $\mathrm{fam}(v) = \mathrm{fam}(v')$, having the same color at time t. Since v and v' observe the same opinions in their respective neighborhood, v and v' will also have the same color anytime after t. It follows that if at some time t there is a bad arrow going from v to some neighbor u (or from u to v), then

there will also be a bad arrow from v' to u (or from u to v'). In particular, this implies that whenever the number of bad arrows adjacent to v is decreased by some amount c, also the identical number of bad arrows adjacent to v' will be decrease by the same amount c.

Recall the proofs of Theorem 24 and Theorem 27. An estimate of the voting time is obtained by upper bounding the number of bad arrows that can possibly disappear during the process. The main argument is the following. It suffices to only consider the bad-arrows adjacent to v in G^{Δ}, since the corresponding bad arrows adjacent to v' will disappear whenever those adjacent to v do.

Let v and v' be two nodes with $\mathrm{fam}(v) = \mathrm{fam}(v')$ having a different color at time t. We can divide every such family that contains nodes of different opinions into two sets S_0 and S_1 according to their initial opinion in the first round. Note that all nodes in either set behave identically. In particular, an adjacent bad arrow from a node u to all nodes of either set disappears at the same time. Since there is bijection between the families of G and the pairs of nodes and singletons of G^{Δ}, and by applying Theorem 24 and Theorem 27 we can bound the voting time by bounding the bad arrows in G^{Δ}. This yields the first part of the claim. Using [11], one can obtain the modular decomposition of G in $\mathrm{O}(|E|)$ time steps. In another $\mathrm{O}(|E|)$ time steps one can select from the modular decomposition those modules that form a family, using that all nodes of a family have the same degree. Hence, G^{Δ} can be constructed in linear time. ◀

Acknowledgement. We would like to thank our supervisors Petra Berenbrink, Andrea Clementi, Robert Elsässer, and Claire Mathieu for helpful discussions and important hints.

References

1. D. Aldous and J. Fill. Reversible Markov Chains and Random Walks on Graphs, 2002. Unpublished. `http://www.stat.berkeley.edu/~aldous/RWG/book.html`.
2. V. Auletta, I. Caragiannis, D. Ferraioli, C. Galdi, and G. Persiano. Minority Becomes Majority in Social Networks. In *Proc. WINE '15*, pages 74–88, 2015.
3. I. Benjamini, S.-O. Chan, R. O'Donnell, O. Tamuz, and L.-Y. Tan. Convergence, unanimity and disagreement in majority dynamics on unimodular graphs and random graphs. *CoRR*, abs/1405.2486, 2014. URL: `http://arxiv.org/abs/1405.2486`.
4. E. Berger. Dynamic Monopolies of Constant Size. *Journal of Combinatorial Theory, Series B*, 83(2):191–200, 2001.
5. S. Brahma, S. Macharla, S.P. Pal, and S.K. Singh. Fair Leader Election by Randomized Voting. In *Proc. ICDCIT '04*, pages 22–31, 2004.
6. F. Bénézit, P. Thiran, and M. Vetterli. Interval consensus: From quantized gossip to voting. In *Proc. ICASSP '09*, pages 3661–3664, 2009.
7. L. Cardelli and A. Csikász-Nagy. The Cell Cycle Switch Computes Approximate Majority. *Scientific Reports*, 2(656), 2012.
8. F. Chierichetti, J.M. Kleinberg, and S. Oren. On Discrete Preferences and Coordination. In *Proc. EC '13*, pages 233–250, 2013.
9. C. Cooper, R. Elsässer, H. Ono, and T. Radzik. Coalescing Random Walks and Voting on Connected Graphs. *SIAM Journal on Discrete Mathematics*, 27(4):1748–1758, 2013.
10. G. Cordasco and L. Gargano. Community Detection via Semi–Synchronous Label Propagation Algorithms. In *Proc. BASNA '10*, pages 1–8, 2010.
11. A. Cournier and M. Habib. A New Linear Algorithm for Modular Decomposition. In *Proc. 19th Colloquium on Trees in Algebra and Programming (CAAP '94)*, pages 68–84, 1994.
12. X. Deng and C. Papadimitriou. On the Complexity of Cooperative Solution Concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.

13 P. Donnelly and D. Welsh. Finite particle systems and infection models. *Mathematical Proceedings of the Cambridge Philosophical Society*, 94(01):167–182, 1983.

14 David Doty. Timing in Chemical Reaction Networks. In *Proc. SODA '14*, pages 772–784, 2014.

15 S. Frischknecht, B. Keller, and R. Wattenhofer. Convergence in (Social) Influence Networks. In *Proc. DISC '13*, pages 433–446, 2013.

16 D. Gifford. Weighted Voting for Replicated Data. In *Proc. SOSP '79*, pages 150–162, 1979.

17 E. Goles. Local Graph Transformations Driven by Lyapunov Functionals. *Complex Systems*, 3(1):173–184, 1989.

18 E. Goles and S. Martínez. *Neural and Automata Networks*. Kluwer, 1990.

19 E. Goles and A.M. Odlyzko. Decreasing Energy Functions and Lengths of Transients for Some Cellular Automata. *Complex Systems*, 2(5):501–507, 1988.

20 E. Goles and J. Olivos. Periodic behaviour of generalized threshold functions. *Discrete Mathematics*, 30(2):187–189, 1980.

21 E. Goles-Chacc, F. Fogelman-Soulié, and D. Pellegrin. Decreasing energy functions as a tool for studying threshold networks. *Discrete Applied Mathematics*, 12(3):261–277, 1985.

22 Y. Hassin and D. Peleg. Distributed Probabilistic Polling and Applications to Proportionate Agreement. *Information and Computation*, 171(2):248–268, 2001.

23 R. Holley and T. Liggett. Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model. *The Annals of Probability*, 3(4):643–663, 1975.

24 Barry W. Johnson, editor. *Design & Analysis of Fault Tolerant Digital Systems*. Addison-Wesley, 1989.

25 D. Kaaser, F. Mallmann-Trenn, and E. Natale. Brief Announcement: On the Voting Time of the Deterministic Majority Process. In *Proc. DISC '15*, 2015.

26 D. Kaaser, F. Mallmann-Trenn, and E. Natale. On the Voting Time of the Deterministic Majority Process. *CoRR*, abs/1508.03519, 2015. URL: `http://arxiv.org/abs/1508.03519`.

27 B. Keller, D. Peleg, and R. Wattenhofer. How Even Tiny Influence Can Have a Big Impact! In *Proc. FUN '14*, pages 252–263, 2014.

28 K. Kothapalli, S. Pemmaraju, and V. Sardeshmukh. On the Analysis of a Label Propagation Algorithm for Community Detection. In *Proc. ICDCN '13*, pages 255–269, 2013.

29 N. Lanchier and C. Neuhauser. Voter model and biased voter model in heterogeneous environments. *Journal of Applied Probability*, 44(3):770–787, 2007.

30 T. Liggett. *Interacting Particle Systems*. Springer, 1985.

31 F. Mallmann-Trenn. Bounds on the voting time in terms of the conductance. Master's thesis, Simon Fraser University, 2014. Master's thesis. `http://summit.sfu.ca/item/14502`.

32 E. Mossel and O. Tamuz. Opinion Exchange Dynamics. *CoRR*, abs/1401.4770, 2014. URL: `http://arxiv.org/abs/1401.4770`.

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

34 D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science*, 282(2):231–257, 2002.

35 D. Peleg. Immunity against Local Influence. In *Language, Culture, Computation. Computing - Theory and Technology*, volume 8001 of *LNCS*, pages 168–179. Springer, 2014.

36 S. Poljak and M. Sůra. On periodical behaviour in societies with symmetric influences. *Combinatorica*, 3(1):119–121, 1983.

37 S. Poljak and D. Turzík. On an application of convexity to discrete systems. *Discrete Applied Mathematics*, 13(1):27–32, 1986.

38 U. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

39 A. Sarıyüce, E. Saule, K. Kaya, and U. Çatalyürek. Shattering and Compressing Networks for Betweenness Centrality. In *Proc. SDM '13*, pages 686–694, 2013.

40 O. Tamuz and R. J. Tessler. Majority Dynamics and the Retention of Information. *Israel Journal of Mathematics*, 206(1):483–507, 2015.

41 P. Winkler. Puzzled: Delightful Graph Theory. *Communications of the ACM*, 51(8):104, 2008.

42 P. Winkler. Puzzled: Solutions and Sources. *Communications of the ACM*, 51(9):103, 2008.