BRICS

Basic Research in Computer Science

BRICS RS-00-8 Mustafa & Pekeč: Democratic Consensus and the Local Majority Rule

Democratic Consensus and the Local Majority Rule

Nabil H. Mustafa
Aleksandar Pekeč

BRICS Report Series **RS-00-8**

ISSN 0909-0878 **May 2000**

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

http://www.brics.dk
ftp://ftp.brics.dk
This document in subdirectory RS/00/8/

Democratic Consensus and the Local Majority Rule

Nabil H. Mustafa* Aleksandar Pekeč†

Abstract

In this paper we study a rather generic communication/ coordination/ computation problem: in a finite network of agents, each initially having one of the two possible states, can the majority initial state be computed and agreed upon (i.e., can a democratic consensus be reached) by means of iterative application of the local majority rule. We show that this task is failure-free only in the networks that are nowhere truly local. In other words, the idea of solving a truly global task (reaching consensus on majority) by means of truly local computation only (local majority rule) is doomed for failure.

We also show that even well connected networks of agents that are nowhere truly local might fail to reach democratic consensus when the local majority rule is applied iteratively. Structural properties of democratic consensus computers, i.e., the networks in which iterative application of the local majority rule always yields consensus in the initial majority state, are presented.

1 Introduction

Attempting to solve a complex problem by a simultaneous coordinated activity of local agents is an idea that arises naturally in a variety of contexts. For example, this idea is fundamental in frameworks as diverse as distributed computing and neural networks. While methods of local

*Department of Computer Science, Duke University, Durham, NC 27708, USA, nabil@cs.duke.edu

†The Fuqua School of Business, Duke University, Durham, NC 27708, USA, pekec@duke.edu

computation and decision-making are often effective in dealing with complex tasks, the successful implementation of such methods often raises a new breed of problems related to coordination and communication of local agents.

In this paper we study a rather generic communication/ coordination/ computation problem: in a finite network of agents, each initially having one of the two possible states, can the majority initial state be computed and agreed upon by means of local computation only? Our simple model assumes bidirectional communication among agents (agent i knows the agent j's state if and only if agent j knows agent i's state) and a synchronous, discrete time, democratic local decision-making procedure (an agent changes its state at time $t+1$ if and only if the majority of agents it communicates with are in the opposite state at time t). We describe the architecture of networks that are always capable of reaching the consensus on the majority initial state of its agents. In particular, we show that, for any truly local network of agents, there are instances in which the network is not capable of reaching such consensus. Thus, every local computation approach that requires reaching consensus among agents' results is not failure-free.

A precise formulation of the model will be given in the next section. Informally, the vertices of a graph $G = (V, E)$ represent the agents and the edges of G represent all (bidirectional) communication links between pairs of agents. Initially, at time $t = 0$, each agent is in one of the two possible states, e.g., colored red or blue (voted Yes or No, having value 0 or 1, ...). Then the local majority rule is applied synchronously and iteratively: an agent has different colors at time t and $t+1$ if and only if the agent's color at time t is not a majority color in the agent's neighborhood in G at time t. We call this discrete time, memoryless, synchronous dynamic process, *local majority process* on G.

The local majority process (and some of its natural extensions) has been studied in frameworks as diverse as social influence [H59, F56, D74, PS83, PT86a, PT86b] and neural networks [GO81, G080, G86, GM90]. Recently, the local majority process has reappeared (under the name *polling process*) in several papers motivated by certain distributed computing problems [P98, B99, FLLPS98, FLLPS99, H98, HP99, HP00, LPS99, NIY99, NIY00]. In fact, Peleg [P96b] points out several areas of distributed computing in which our model could be relevant.[1] These are

[1]We believe that the potential applicability of the local majority process goes beyond classical distributed computing problems. For example, anyone interested

areas that revolve around the idea of eliminating the damage caused by failed processors, or at least restricting their influence, by maintaining replicated copies of crucial data and performing a simple voting procedure among the participating processors whenever faults occur, with the goal of adopting the values stored at the majority of the processors as the correct data. The work of this flavor can be found in classical problems of agreement and consensus [AW98, LSP82, B87, DPPU88], system-level diagnosis [S86, P96a, DP96], distributed database management [D85, H84], quorum systems [G79, GB85, SB94, PW95, W96], and fault-local mending [KP95a, KP95b].

To give a concrete example, suppose that all processors in a distributed network collectively store some value and suppose that this value is distorted in some of the processors (distortions could be due to various reasons, even due to fundamental imprecise nature of floating point operations). The goal is to restore the correct value in all of the processors by means of local communication only, in particular, by triggering the local majority process. For example, if stored distortions are due to a rounding error (rounding up or down), a desirable feature would be for all processors to accept the rounded value which is stored in the majority of processors. Which network structures allow for successful restoration of the (global) majority value in all of the processors?

A natural question to ask is when does the local majority process ensure that all agents reach a consensus on the initial majority state? We will say that G is a *democratic consensus computer* (d.c.c.) if, for any set of initial states (there are 2^n such sets), the local majority process simultaneously brings all agents into the state that was the initial majority state. Note that, according to the local majority process, once all agents are in the same state, no agent will change its state everafter. All of the recent papers dealing with the local majority process and its modifications [P98, B99, FLLPS98, FLLPS99, H98, HP99, HP00, LPS99, NIY99, NIY00] investigated how badly could the local majority process (and its variations) miscalculate the initial majority (on a specific class of graphs)[2] In contrast to these results, we are interested in G which are

in data aggregation by means of local computation/communication only should be interested in this model (at least as a starting point towards possible more complex models).

[2]For example, Berger [B99] has shown that for every n there exists a G on at least n vertices and the set of states such that only 18 vertices are in one state and the rest are in the other, yet the local majority process forces all vertices to simultaneously end up in the initial minority state.

immune to miscalculations in the local majority process, i.e., the focus of this paper are democratic consensus computers and investigation of their structure.

Since being a democratic consensus computer is seemingly a very strong property, one would expect that a sort of an impossibility theorem holds. As will be shown, the situation is not that simple and the full characterization of d.c.c.'s remains an open problem. However, our results demonstrate in several ways that the non-locality is inherent property of every d.c.c.. Thus, reaching consensus on the majority is a truly non-local task in the sense that a most natural local computation procedure is failure-free only if computing local majority is essentially as complex as computing global majority.

As already mentioned, the local majority process is precisely formulated in the next section. Furthermore, we review some known properties of the model and formally define the class of graphs that we call democratic consensus computers. We end Section 2 by stating and proving several basic properties of democratic consensus computers.

In Section 3 we explore the structure of democratic consensus computers. For example, in this section we show that every such d.c.c. must have a trivial min-cut, a non-unique max-cut, diameter at most four and that for any vertex v in a d.c.c. the set of vertices that are neighbors of v or neighbors of the neighbors of v is a majority-making set (i.e., has more than half of the vertices of G).

In Section 4 we study highly connected graphs, i.e., those with the minimum degree of $n-3$ and show that if such a graph is a d.c.c., then there must exist a "truly global" vertex which we call a *master* (that is, a vertex connected to every other vertex in the graph). Furthermore, we give full characterization of democratic consensus computers with $\delta(G) \geq n-3$ and, as a byproduct, show that there exists d.c.c.'s on n vertices, n odd, with exactly k masters for every positive k except for $k = (n-3)/2$.

Some generalizations of our model and relaxations of the definition of democratic consensus computer are presented and discussed in Section 5. This includes emulation results showing how our model can be used to study seemingly more complex models.

In Section 6 we discuss some assumptions of our model and try to illustrate why our model is a most natural one to study.

We close the paper with a brief summary of our results and directions for further research.

2 Democratic Consensus Computers

A standard graph theoretic notation is used throughout the paper. Cardinality of the set S is denoted by $|S|$ and the complement of the set is denoted by S^c. $G = (V, E)$ denotes an undirected, simple, finite graph G with the vertex set V, $|V| = n$, and the edge set E (i.e. $E \subseteq \{S \subseteq V : |S| = 2\}$). We say that the vertices u and v are adjacent or *neighbors* in G if and only if $\{u, v\} \in E$. We will abuse the notation and sometimes denote an edge $\{i, j\}$ by (i, j). The *neighborhood* of a vertex v in the set $S \subseteq V$ is the set of the neighbors of v that are in S, $N_S(v) := \{s \in S : (v, s) \in E\}$. Note that $N_S(v) = N_V(v) \cap S$. The *degree* of a vertex v in S, denoted as $deg_S(v)$ is the number of neighbors of v that are in S, i.e. $deg_S(v) = |N_S(v)|$. In the rest of the text, we will omit the subscript when $S = V$, i.e., we will refer to $N_V(v)$ as $N(v)$ and to $deg_V(v)$ as $deg(v)$. The maximum degree of a vertex in G is denoted by Δ, where $\Delta = \Delta(G) = \max\{deg(v) : v \in V\}$ and the minimum degree is denoted by δ, where $\delta = \delta(G) = \min\{deg(v) : v \in V\}$.

Given a pair of nonempty $S, T \subseteq V$, let $E(S, T) = \{\{s, t\} \in E : s \in S, t \in T\}$. Recall that $G = (V, E)$ is bipartite with bipartition $S \cup S^c = V$ if $E(S, S^c) = E$. A pair of nonempty sets S, S^c defines a (S, S^c)-*cut* represented by $E(S, S^c)$. A cut is *trivial* if either S or S^c is a one element set. An (S, S^c)-cut is a *min-cut* if $|E(S, S^c)| \le |E(T, T^c)|$ for all pairs of nonempty sets $T, T^c \subset V$. Similarly, (S, S^c)-cut is a *max-cut* if $|E(S, S^c)| \ge |E(T, T^c)|$ for all pairs of nonempty sets $T, T^c \subset V$.

A graph $H = (V', E')$ is a *subgraph* of G, denoted as $H \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$. We'll also use the notation $G \backslash H = (V, E \backslash E')$. The following graphs on n vertices are denoted in the standard way: the complete graph K_n, the path P_n, and the cycle C_n. A $K_k \subseteq G$ is a *clique* (or a k-*clique*) in G. The *complement* of G is denoted by $G^c = (V, E^c) = K_n \setminus G$. G is *connected* if for every pair of vertices $u, v \in V$, there exists a path $P \subseteq G$ containing both u and v. Otherwise, G is *disconnected*. A *connected component* H of G is a maximal connected subgraph $H \subseteq G$. The *distance* between two vertices u and v in G, denoted as $dist(u, v)$, is the smallest k for which there exists $P_{k+1} \subseteq G$ that contains both u and v (might not be defined in a disconnected graph). The *diameter* of a connected graph G is $diam(G) = max\{dist(u, v) : u, v \in V\}$.

Some non-standard terminology: a vertex v is a *master* if $deg(v) = n - 1$ (i.e., v is adjacent to every other vertex). We also say that v is a k-*master* if $deg(v) = n - 1 - k$ (i.e., v is adjacent to all but k other vertices). Note that 0-master and master are equivalent notions, and we will use them interchangeably throughout the rest of the text.

In our model all agents and communication links in the system are represented by a graph G in a natural way. That is, the vertices of G are in a one-to-one correspondence with the agents and the edges of G correspond to adjacency relation among the agents.

A *coloring* of the graph G, $c^t : V \to \{0, 1\}$ defines an assignment of binary values (colors) to the vertices of G at time t. We use the notation $c^t_v := c^t(v)$ to denote the color of a vertex v at time t. The notation $sum(c^t) := \sum_{v \in V} c^t_v$ will also be useful. A color which is assigned to more than $|V|/2$ vertices at a time t is called the *majority color* of the coloring c^t and denoted by $maj(c^t)$. Thus, $maj(c^t) = 1$ if and only if $sum(c^t) > n/2$ and $maj(c^t) = 0$ if and only if $sum(c^t) < n/2$. Note that $maj(c^t)$ is not defined if $|V|$ is even and c^t defines an equipartition of V, i.e., if $sum(c^t) = n/2$. A coloring c^t is a *consensus* if it is constant, i.e. if all the vertices of G have the same binary values (colors). Thus, c^t is a consensus if and only if $c^t_v = maj(c^t)$ for all $v \in V$. We will sometimes abuse the notation and write $c^t = 0$ or $c^t = 1$ for consensus in color 0 and 1, respectively. Another abuse of notation is $(1 - c^t)$ denoting the coloring obtained from c^t by changing the color of every vertex, i.e., for every $v \in V$ and coloring c^t, $(1 - c^t)(v) = 1 - c^t(v)$.

Note that in our model $c^t(v)$ corresponds to the state of agent, represented by v, at time t.

The main object of our study is the *local majority process* $LMP(G, c^0)$, a discrete time process on G that is based on the iterative application of the local majority rule. The process is completely defined by G and the *initial coloring* c^0. For every $t = 0, 1, 2, \ldots$, the coloring c^{t+1} is derived by applying the *local majority rule* on $N(v)$ for each vertex in G:

$$c^{t+1}_v = \begin{cases} c^t_v & \text{if } |\{w \in N(v) : c^t_w = c^t_v\}| \geq |N(v)|/2 \\ 1 - c^t_v & \text{if } |\{w \in N(v) : c^t_w \neq c^t_v\}| > |N(v)|/2 \end{cases} \tag{1}$$

The local majority rule simply states that, at the next discrete time step, the color assigned to a vertex v will be the color of the majority of its neighbors. Note that an even degree vertex will retain its color whenever exactly half (or more) of its neighbors have the same color. The above rule also implies that the local majority rule is executed simultaneously for all the vertices. The change from c^t to c^{t+1} is called a *global update* of G at time $t + 1$, while the change of the color of a particular vertex v from c^t_v to c^{t+1}_v is called a *local update*. We say that there is a *majority switch* at time $(t + 1)$ if $maj(c^t) \neq maj(c^{t+1})$.

Note that if c^t is a consensus, then $c^{t+k} = c^t$ for all positive integers k. If, for some positive integer t, c^t is a consensus, then we say that G *reaches consensus* for c^0. If G reaches consensus c^t for c^0 and $c^t = maj(c^0)$, then we say that the $LMP(G, c^0)$ *correctly computes the initial majority* and that G admits a *democratic consensus* for the initial coloring c^0. A graph G is a *democratic consensus computer* (or a *d.c.c.* in short) if, for every c^0 (there are 2^n such colorings), $LMP(G, c^0)$ correctly computes the initial majority. In other words, G is a d.c.c. if G admits democratic consensus for all of the 2^n possible initial colorings. Note that for every graph with even number of vertices there exists a c^0 where $maj(c^0)$ is not defined. Thus, G can be a democratic consensus computer only if it has an odd number of vertices.

Assumption.
Throughout the rest of the paper we assume that n is odd.

Our first observation about democratic consensus computers is the following proposition.

Proposition 2.1 *Let G be a d.c.c. and let c^0 be an initial coloring of G. Then there are no majority switches for $LMP(G, c^0)$, i.e. $maj(c^t) = maj(c^0)$ for $t = 0, 1, 2, \ldots$.*

Proof: Suppose that there exists t^* such that $maj(c^{t^*}) \neq maj(c^0)$. Define another initial coloring $d^0 = c^{t^*}$, and observe that $d^t = c^{t^*+t}$.

Since G is a d.c.c., the local majority process reaches consensus on the $maj(c^0)$. In other words, there exists a t_0 such that $c^t = maj(c^0)$ for every $t \geq t_0$. Thus, for $t \geq t_0 - t^*$, $maj(d^t) = maj(c^{t+t^*}) = maj(c^0) \neq maj(c^{t^*}) = maj(d^0)$. Therefore G is not a d.c.c. since it does not admit democratic consensus for d^0. A contradiction. $\square$

The related research in the area of neural networks and models of social influence was geared towards finding properties of the local majority process, rather than towards finding specific graphs (i.e., network architectures) having certain desirable properties. In particular, we use results about the behavior of the sequence $c^0, c^1, c^2, \ldots$. There are only 2^n possible colorings and c^{t+1} is a function of G and c^t, thus the sequence $c^0, c^1, c^2, \ldots$ must become periodic, i.e., there exists positive integers t_0 and k such that $c^{t+k} = c^t$ for every $t \geq t_0$. Obviously, the period k and t_0 are not larger than 2^n. Somewhat surprisingly, the period can be only one or two and there exists t_0 smaller than $|E|$. We first state the original result from neural network literature[3] and then show how this result applies to our model.

Theorem 2.2 (Goles-Olivos [GO81], Goles [G86]) *Let $A = [a_{ij}]$ be a $n \times n$ matrix and $b \in \mathbf{R}^n$. For any $c^0 \in \{0,1\}^n$ define a dynamic process by $c_i^{t+1} = p[Ac^t - b]_i$ where $p(x) = 1$ if $x \geq 0$ and $p(x) = 0$ if $x < 0$.*

If A is symmetric, there exists t_0 such that $c^t = c^{t+2}$ for all $t \geq t_0$. Furthermore, if A is integer valued and $b = A(1,\ldots,1)^T$, then t_0 can be chosen so that $t_0 \leq |(\sum_{i,j}|a_{ij}|) - n|/2$

Corollary 2.3 *Consider the sequence $c^0, c^1, c^2, \ldots$ defined by the local majority process on G with initial coloring c^0, $LMP(G, c^0)$. Then there exists $t_0 < |E|$ such that $c^t = c^{t+2}$ for every $t \geq t_0$.*

Proof: Let A be a slightly modified adjacency matrix of G, i.e., let $A = [a_{uv}]$ be defined with

$$a_{uv} = \begin{cases} 1 & \text{if } \{u,v\} \in E \\ 1 & \text{if } u = v \text{ and } deg(v) \text{ is even} \\ 0 & \text{otherwise.} \end{cases}$$

It is straightforward to check that, for any c^0, a dynamic process from Theorem 2.2 with A as defined is exactly $LMP(G, c^0)$. Note that A is a zero-one symmetric matrix: $a_{uv} = a_{vu}$ since $\{u,v\} = \{v,u\}$. Further note that $\sum_{u,v}|a_{uv}| = 2|E| + |\{v : deg(v) \text{ is even}\}|$. Thus, t^0 from Theorem 2.2 can be chosen so that $t_0 \leq |E| + |\{v : deg(v) \text{ is even}\}|/2 - |V| < |E|$. $\Box$

Many of our results will be based on the *"period is at most two"* property.

Next we show that a monotonicity property with respect to the structure of the coloring holds in the local majority process. As the next Lemma shows, if at time t the color of some set of vertices is changed

[3] Theorem 2.2 is not the most general result. Various variations can be found in a rather comprehensive collection of results related to dynamic behavior of neural and automata networks by Goles and Martinez [GM90]. The *period is either one or two* property holds in models beyond the symmetric neural network model. For example, dynamical systems with more general threshold functions and allowing for more than two possible colors are studied in [PS83, PT86a, PT86b], while sufficient conditions for the property in the case of LMP on infinite graphs were studied in [M94a, M94b, M94c].

from $1-i$ to i and colors of all other vertices remain the same, then, at any later time $t \geq t'$ the number of vertices of color i is as at least as large as it would be without the change that was executed at time t.

Lemma 2.4 *Let $V_i(c^t) = \{v \in V : c_v^t = i\}$, $i = 0, 1$, where c^t is a coloring of $G = (V, E)$. If there exists $i \in \{0, 1\}$ and colorings c^t and $d^{t'}$ such that $V_i(c^t) \subseteq V_i(d^{t'})$ then $V_i(c^{t+k}) \subseteq V_i(d^{t'+k})$ for $k = 0, 1, 2, \ldots$.*

Proof: By induction on k. If $k = 0$ there is nothing to prove. Suppose $V_i(c^{t+k}) \subseteq V_i(d^{t'+k})$. We have to show that, for every $v \in V$, $c_v^{t+k+1} = i \Rightarrow d_v^{t'+k+1} = i$. It follows from the assumption that $N(v) \cap V_i(c^{t+k}) \subseteq N(v) \cap V_i(d^{t'+k})$ and, in particular, $c_v^{t+k} = i \Rightarrow d_v^{t'+k} = i$. Hence, if $c_v^{t+k+1} = i$ because $|N(v) \cap V_i(c^{t+k})| > |N(v)|/2$, then $|N(v) \cap V_i(d^{t'+k})| > |N(v)|/2$ also, and $d_v^{t'+k+1} = i$. If $c_v^{t+k+1} = i$ because $c_v^{t+k} = i$ and $|N(v) \cap V_i(c^{t+k})| = |N(v)|/2$, then $d_v^{t'+k} = i$ and $|N(v) \cap V_i(d^{t'+k})| \geq |N(v)|/2$ which shows that $d_v^{t'+k+1} = i$. $\square$

According to the definition, in order to check whether G is a democratic consensus computer, one would have to check whether G admits democratic consensus for all 2^n possible initial colorings c^0. However, because of the monotonicity property described in Lemma 2.4, it suffices to consider only colorings c^0 such that $sum(c^0) = (n+1)/2$. (There are $\binom{n}{(n+1)/2} = O(2^n/\sqrt{n})$ such colorings).

Theorem 2.5 *Suppose G admits democratic consensus for any c^0 such that $sum(c^0) = n/2 + 1$. Then G is a democratic consensus computer.*

Proof: By symmetry, if G admits democratic consensus for all c^0 with $maj(c^0) = 1$, then G admits democratic consensus for all c^0 with $maj(c^0)$ $= 0$ also. (If G does not admit democratic consensus for c^0 with $maj(c^0) =$ 0, then G does not admit democratic consensus for $1 - c^0$ also. Note that $maj(1 - c^0) = 1 - maj(c^0) = 1$.)

Suppose d^0 such that $maj(d^0) = 1$, i.e., $sum(d^0) \geq (n+1)/2$. Thus, there exists a c^0 with $sum(c^0) = (n+1)/2$ such that c^0 and d^0 satisfy conditions of Lemma 2.4 with $i = 1$ (e.g., construct c^0 from d^0 by changing color of any $sum(d^0) - sum(c^0)$ vertices w such that $d_w^0 = 1$). By assumption, G admits democratic consensus for c^0. Thus, using terminology of Lemma 2.4, there exists t such that $V_1(c^t) = V$ and, by the lemma, $V_1(c^t) \subseteq V_1(d^t)$. Therefore, d^t is a consensus with $maj(d^t) = 1$ which shows that G admits democratic consensus for d. $\square$

Remark. Unfortunately, it is not true that adding an edge to or deleting an edge from a democratic consensus computer G preserves the property *"democratic consensus computer"*. In other words, if G is a d.c.c., $G+e$ might not be. Similarly, if G is not a d.c.c, $G-e$ could be. For example, consider

$$(K_n)^c \subset K_n \setminus P_{n-1} \subset K_n \setminus P_{(n+1)/2} \subset K_n$$

where n is odd. We later show that $(K_n)^c$ is not a d.c.c. (Corollary 3.3), $K_n \setminus P_{n-1}$ is a d.c.c. (Theorem 4.12), $K_n \setminus P_{(n+1)/2}$ is not a d.c.c ((b) of Proposition 3.1), and that K_n is a d.c.c. ((a) of Proposition 3.1). Thus, the graph property *"democratic consensus computer"* is **not** monotone in the sense that addition or deletion of an edge in G preserves the property.

We close this section by showing that masters in G compute majority instantly, i.e., the color of a master at time $t+1$ is $maj(c^t)$. (Larger the difference between the majority and minority color of c^t, smaller degree of v is needed to ensure $c_v^{t+1} = maj(c^t)$.)

Proposition 2.6 *If v is a master in G, then $c_v^{t+1} = maj(c^t)$. More generally, if v is a k-master in G and $|sum(c^t) - n/2| \geq (k+1)/2$, then $c_v^{t+1} = maj(c^t)$.*

Proof: Recall that a vertex v is a k-master if $deg(v) = n-(k+1)$. Note that $|\{w \in V : c_w^t = 1 - maj(c^t)\}| \leq deg(v)/2$ implies $c^{t+1}(v) = maj(c^t)$ (at time t, at most half of v's neighbors have color $1 - maj(c^t)$; if each of the two colors is the color of exactly half of v's neighbors, then no other vertex has color $1 - maj(c^t)$ and, in particular, $c_v^t = maj(c^t)$ and the local majority process ensures $c_v^{t+1} = c_v^t = maj(c^t)$). In order to complete the proof note that $|sum(c^t) - n/2| \geq (k+1)/2$ is equivalent to $|\{w \in V : c_w^t = 1 - maj(c^t)\}| \leq (n-(k+1))/2$. □

3 Structural properties

Let's start by presenting a class of graphs that are d.c.c. and a class of graphs that are not d.c.c..

Proposition 3.1
(a) *A graph G with more than $n/2$ masters is a democratic consensus computer.*
(b) *A graph G with exactly $(n-1)/2$ masters is not a democratic consensus computer.*

Proof: First suppose that G has more than $n/2$ masters. Then, by Proposition 2.6, for any c^0 and any master $v \in V$, $c_v^1 = maj(c^0)$. Since there are more than $n/2$ masters, it follows that $maj(c^1) = maj(c^0)$. Thus, $c_v^2 = maj(c^1) = maj(c^0)$ (first equality follows from Proposition 2.6 with $t = 1$). Since every vertex w is adjacent to all the masters, it follows that every $w \in V$ that is not a master is connected to more than $n/2$ masters. Thus, $c_w^2 = maj(c^0)$ because masters are the majority of w's neighbors and, as already observed, $c_v^1 = maj(c^0)$ for every master v. Hence, c^2 is the consensus in color $maj(c^0)$ and (a) follows.

In order to prove (b), let G be a graph with exactly $(n-1)/2$ masters and let $c_v^0 = 0$ if v is master and $c_w^0 = 1$ if v is not a master. Note that $maj(c^0) = 1$. Every $w \in V$ that is not a master is connected to all $(n-1)/2$ masters (all having color 0 at time t=0) and is connected to at most $(n+1)/2 - 2 = (n-3)/2$ vertices that are not masters (w is not connected to itself and to at least one more vertex u because w is not a master; u is not a master either because it is not connected to w). Thus, $c_w^1 = 0$ and $maj(c^1) = 0 \neq maj(c^0)$. Hence, by Proposition 2.1, G is not a democratic consensus computer. $\Box$

Next we give a characterization of democratic consensus computers which indicates a way towards a static representation in the form of existence of a particular partition of the vertices of G.

Theorem 3.2 *G is* **not** *a democratic consensus computer if and only if at least one of the following holds:*
(a) *There exists c^0 such that $maj(c^0) \neq maj(c^1)$*
(b) *There exists a partition of V into four sets A_0, A_1, B_0, B_1 satisfying*

1. $|B_0||B_1| = 0 \Rightarrow |A_0||A_1| = 1$,

2. *For every $v \in V$ and $i = 0, 1$:*

$$v \in A_i \Rightarrow deg_{A_i}(v) - deg_{A_{1-i}}(v) \geq |deg_{B_i}(v) - deg_{B_{1-i}}(v)|$$

3. *For every $v \in V$ and $i = 0, 1$:*

$$v \in B_i \Rightarrow deg_{B_{1-i}}(v) - deg_{B_i}(v) > |deg_{A_i}(v) - deg_{A_{1-i}}(v)|$$

Proof: Suppose G is not a democratic consensus computer. If G admits a consensus for every possible initial coloring c^0, there must exist d^0 for which G does not admit a democratic consensus, i.e., there exists d^0

and t such that d^t is a consensus and $maj(d^0) \neq maj(d^t)$. Obviously, in the sequence $d^0, d^1, \ldots, d^t$, there exists $t' < t$ such that $maj(d^{t'}) \neq maj(d^{t'+1})$. Thus, (a) holds for $c^0 := d^{t'}$.

Thus, we may assume that there exists c^0 for which G does not admit a consensus. By Corollary 2.3 there exists t such that $c^t = c^{t+2}$. For $i = 0, 1$ define $A_i := \{v \in V : i = c_v^t = c_v^{t+1}\}$ and $B_i := \{v \in V : i = c_v^t \neq c_v^{t+1}\}$. Note that A_0, A_1, B_0, B_1 partition V and that *1.* must hold since neither c^t nor c^{t+1} is a consensus. Since for every $v \in A_i$, $c_v^t = c_v^{t+1}$,

$$deg_{A_i}(v) + deg_{B_i}(v) \geq deg_{A_{1-i}}(v) + deg_{B_{1-i}}(v).$$

Similarly, for every $v \in A_i$, $c_v^{t+1} = c_v^{t+2}$ implies (because $\{w : c_w^{t+1} = i\} = A_i \cup B_{1-i}$)

$$deg_{A_i}(v) + deg_{B_{1-i}}(v) \geq deg_{A_{1-i}}(v) + deg_{B_i}(v).$$

These two inequalities imply *2.* In the same manner, it follows that for every $v \in B_i$, $c_v^t \neq c_v^{t+1}$ implies

$$deg_{A_{1-i}}(v) + deg_{B_{1-i}}(v) > deg_{A_i}(v) + deg_{B_i}(v)$$

and that $c_v^{t+1} \neq c_v^{t+2}$ implies

$$deg_{A_i}(v) + deg_{B_{1-i}}(v) > deg_{A_{1-i}}(v) + deg_{B_i}(v).$$

Hence, *3.* follows from these two inequalities.

Conversely, suppose that (a) holds. Then, by Proposition 2.1, G is not a d.c.c.

Finally, suppose that (b) holds. Define $c_v^0 := i$ for $v \in A_i \cup B_i$, $i = 0, 1$. Note that *3.* implies that either none or both sets B_0 and B_1 must be nonempty (otherwise, if B_i is not empty and B_{1-i} is empty, the lefthand side of inequality in *3.* would be less than or equal to zero for a $v \in B_i$ thereby automatically violating the inequality). If both B_0 and B_1 are empty, *1.* implies that both A_0 and A_1 are not empty. Hence, in either case, c^0 is not a consensus because $A_i \cup B_i \neq \emptyset$ for $i = 0, 1$. Note that *2.* implies that $c_v^1 = c_v^0$ for $v \in A_i$ and that *3.* implies that $c_v^1 \neq c_v^0$ for $v \in B_i$, $i = 0, 1$. Furthermore, *2.* also implies that $c_v^2 = c_v^1$ for $v \in A_i$ and *3.* implies that $c_v^2 \neq c_v^1$ for $v \in B_i$, $i = 0, 1$. Hence, $c^2 = c^0$ and the sequence $c^0, c^1, c^2, \ldots$ never admits a consensus. Thus, G is not a democratic consensus computer. $\square$

Theorem 3.2 indicates possible ways of adding edges to G that is not a democratic consensus computer so that the new graph is still not a democratic consensus computer. For example, if (b) holds for G and if there exists four edges defining a 4-cycle

$$E' := \{(w, x), (x, y), (y, z), (z, w)\}$$

such that $E' \cap E = \emptyset$, then it is straightforward to check that (b) remains to hold for $G' = (V, E \cup E')$ provided that one of the following is true: (i) $w, x \in A_0$ and $y, z \in A_1$, (ii) $w, x \in B_0$ and $y, z \in B_1$, (iii) $w \in A_0, x \in B_0, y \in A_1$ and $z \in B_1$.

Special cases of Theorem 3.2 help identify large classes of graphs that are not d.c.c.'s. and provide insights into the structure of graphs that are d.c.c.'s.

Corollary 3.3 *Let G be bipartite or disconnected. Then G is not a democratic consensus computer.*

Proof: First suppose that G is disconnected. Let $A_0 \neq V$ be one of G's connected components. Set $A_1 := V \setminus A_0$ and $B_0 = B_1 = \emptyset$. Note that this partition satisfies conditions *1., 2., 3.* from (b) in Theorem 3.2. Thus, G is not a democratic consensus computer.

Next suppose that G is a connected bipartite graph with bipartition $B_0 \cup B_1 = V$. Set $A_0 = A_1 = \emptyset$ and note that *1., 2., 3.* from (b) in Theorem 3.2 hold. Thus, G is not a democratic consensus computer. $\square$

Corollary 3.4 *Let G be a democratic consensus computer.*
(a) *Every min-cut in G is trivial.*
(b) *G does not have a unique max-cut.*

Proof: First note that if (S, S^c) is a non-trivial min-cut, then for every $v \in S$, $deg_S(v) \geq deg_{S^c}(v)$ (if not, then (S, S^c) is not a min-cut since the cut $(S \setminus \{v\}, S^c \cup \{v\})$ has lesser edges) and for every $v \in S^c$, $deg_{S^c}(v) \geq deg_S(v)$ (if not, then (S, S^c) is not a min-cut since the cut $(S \cup \{v\}, S^c \setminus \{v\})$ has lesser edges). Thus, setting $A_0 = S, A_1 = S^c$ and $B_0 = B_1 = \emptyset$ gives a partition from (b) in Theorem 3.2. Hence, G is not a democratic consensus computer.

Similarly, note that in a unique max-cut (S, S^c) for every $v \in S$, $deg_S(v) < deg_{S^c}(v)$ (if not, then (S, S^c) is not a max-cut since the cut $(S \setminus \{v\}, S^c \cup \{v\})$ has more edges), and for every $v \in S^c$, $deg_{S^c}(v) < deg_S(v)$ (if not, then (S, S^c) is not a max-cut since the cut $(S \cup \{v\}, S^c \setminus \{v\})$ has more edges). Thus, setting $B_0 = S, B_1 = S^c$ and $A_0 = A_1 = \emptyset$

gives a partition from (b) in Theorem 3.2. Hence, G is not a democratic consensus computer. $\square$

The last corollary indicates that democratic consensus computers are highly connected graphs (in the sense that having only trivial min-cuts and many max cuts could be taken as a good indication of high level of connectivity). The following theorem and its corollary provide another confirmation of this claim.

Theorem 3.5 *Let G be a democratic consensus computer. Then for every $v \in V$*

$$|\bigcup_{w \in N(v)} N(w)| \geq n/2. \tag{2}$$

Proof: First note that we can assume that G is connected (by Corollary 3.3) and that $n > 2$.

Suppose (2) does not hold for some $v \in V$. Let $u \in V$ be a vertex of the minimum degree among all vertices v for which (2) is violated. Let c^0 be such that $c^0_v = 1$ for every $v \in \bigcup_{w \in N(u)} N(w)$ and such that $sum(c^0) = (n+1)/2$. Note that $c^0_u = 1$ and that $maj(c^0) = 1$. Let d^0 be such that $d^0_v \neq c^0_v$ if and only if $v = u$ (i.e., the only difference between c^0 and d^0 is in the color of u). Note that $sum(d^0) = (n-1)/2$ and thus

$$maj(d^0) = 0 \neq 1 = maj(c^0). \tag{3}$$

Observe that for all $v \notin N(v) \cup \{u\}$, $w \in N(v) \Rightarrow c^0_w = d^0_w$, and hence $c^1_v = d^1_v$. Further observe that $c^1_u = d^1_u = 1$ because the color of all neighbors of u is 1 in both c^0 and d^0 (and u has at least one neighbor since G is connected). Finally observe that by the choice of u and the fact that G is connected and $n > 2$, $deg(v) \geq 2$ for all $v \in N(u)$. Since the color of all neighbors of v other than u is 1 in both c^0 and d^0, it follows that $c^1_v = d^1_v$ for $v \in N(u)$. Hence, $c^1 = d^1$ and thus, because of (3), either $maj(c^1) \neq maj(c^0)$ or $maj(d^1) \neq maj(d^0)$. In either case, it follows from Proposition 2.1 that G is not a democratic consensus computer. $\square$

The theorem shows that democratic consensus computers are nowhere truly local since the second neighborhood of any vertex contains a majority of the vertices of V. Hence, the local majority process always reaches a consensus on the initial majority color only if the local majority rule is **nowhere** local. Hence, the theorem can be viewed as a sort of an impossibility result.

Corollary 3.6 *If G is a democratic consensus computer then $diam(G) \leq 4$.*

Proof: Follows immediately from (2) because for any two vertices $u, v \in V$,

$$(\bigcup_{w \in N(u)} N(w)) \bigcap (\bigcup_{w \in N(v)} N(w)) \neq \emptyset.$$

$\square$

Exhaustive computer aided search confirmed that $diam(G) \leq 2$ for every democratic consensus computer on at most 13 vertices. We conjecture that a much stronger statement is true (also confirmed to hold for $n \leq 13$ by an exhaustive search method).

Master Conjecture. Every democratic consensus computer contains a master.

This is a rather strong conjecture because it implies that a necessary condition for reaching democratic consensus is the existence of a vertex connected to all the other vertices, thereby annihilating any notion of local computation. In the next section we'll show that the master conjecture holds for graphs G with $\delta(G) \geq n-3$. Note that intuitively, such graphs should be considered as prime candidates for a counterexample to the conjecture since all of the vertices in these graphs are either masters or very close to being masters (i.e., 0-masters, 1-masters, or 2-masters). Thus our result that the master conjecture holds for graphs with $\delta(G) \geq n-3$ provides strong evidence towards the truth of the master conjecture.

4 The case of $\delta(G) \geq n-3$.

In the first part of this section we show that the Master Conjecture holds for G with $\delta(G) \geq n-3$. Moreover, in the second part we give complete characterization of democratic consensus computers with $\delta(G) \geq n-3$. We close the section by demonstrating that, for every n and positive $k \neq (n-1)/2$, there exists a democratic consensus computer whose number of masters is exactly k.

A direct consequence of Proposition 2.6 is that the only colorings c^0 for which G might not admit a democratic consensus are the tight ones, i.e., c^0 such that $sum(c^0) = (n+1)/2$. (The case $sum(c^0) = (n-1)/2$ is symmetric).

Proposition 4.1 *If $\delta(G) \geq n-3$, then G admits democratic consensus for every c^0 such that $sum(c^0) \geq (n+3)/2$.*

Proof: Note that every $v \in V$ is either a master, a 1-master, or a 2-master. Thus, by Proposition 2.6, $c^1_v = maj(c^0)$ for every $v \in V$. □

If $\delta(G) \geq n-3$, then G^c has a very simple structure since $\Delta(G^c) = (n-1) - \delta(G) \leq (n-1) - (n-3) = 2$. In other words, a connected component of G^c is a single vertex, a path, or a cycle. The decomposition of G^c into its connected components $H_1 = (V_1, E^c_1)$, $H_2 = (V_2, E^c_2)$, ..., $H_m = (V_m, E^c_m)$ [4] will be used throughout this section and we will often abuse the notation and identify $V(H)$ with H whenever such notation will be unambiguous (e.g., we will often say that the connected components of G^c define a partition of V).

Another convenient property of G with $\delta(G) \geq n-3$ is that every vertex in G is either a master, a 1-master, or a 2-master. Thus, the following lemma gives a complete boolean formula representation of local updates for colorings c^t with $sum(c^t) = (n+1)/2$.

Lemma 4.2 *Let c^t such that $sum(c^t) = (n+1)/2$.*
(a) *If v is a master, then $c^{t+1}_v = 1$.*
(b) *If v is a 1-master, then $c^{t+1}_v = 1 - c^t_v c^t_w$ where w is the unique vertex not adjacent to v.*
(c) *If v is a 2-master, then $c^{t+1}_v = 1 - c^t_u c^t_w$ where u and w are the two vertices not adjacent to v.*

Proof: Since $maj(c^t) = 1$, (a) follows directly from Proposition 2.6.

If v is a 1-master then $V \setminus N(v) = \{v, w\}$, so

$$|\{u \in N(v) : c^t_u = 1\}| = \frac{n+1}{2} - c^t_v - c^t_w$$

Note that $c^{t+1}_v = 0$ if and only if $|\{u \in N(v) : c^t_u = 1\}| < |N(v)|/2 = (n-2)/2$ ($deg(v) = n-2$ is odd, so tie is impossible). But the last inequality holds if and only if $c^t_v = c^t_w = 1$. Thus, (b) holds.

[4] In other words, V_i, $i = 1, \ldots m$ are pairwise disjoint, $V_1 \cup \ldots \cup V_m = V$, and $E^c_1 \cup \ldots \cup E^c_m = E^c$

Similarly, if v is a 2-master then $V \setminus N(v) = \{v, u, w\}$, so

$$|\{u \in N(v) : c_u^t = 1\}| = \frac{n+1}{2} - c_v^t - c_u^t - c_w^t$$

First suppose $c_v^t = 0$. Then, $c_v^{t+1} = 0$ if and only if $|\{u \in N(v) : c_u^t = 1\}| \leq |N(v)|/2 = (n-3)/2$ and this is true if and only if $c_u^t = c_w^t = 1$. Thus, (c) holds if $c_v^t = 0$. Finally, suppose $c_v^t = 1$. Then, $c_v^{t+1} = 0$ if and only if $|\{u \in N(v) : c_u^t = 1\}| < |N(v)|/2 = (n-3)/2$ and, again, this is true if and only if $c_u^t = c_w^t = 1$. Thus, (c) also holds if $c_v^t = 1$. □

This lemma allows us to track action of the local majority process on G. We define an *auxiliary graph* $AG = (V, E(AG))$. Edges of AG are defined by formulas from (b) and (c) from the lemma:

$$\begin{aligned} E(AG) \quad = \quad & \{\{v, w\} : d_G(v) = n-2, \{v, w\} \notin E(G)\} \\ & \cup \\ & \{\{u, w\} : \exists v, d_G(v) = n-3, \{v, u\}, \{v, w\} \notin E(G)\}. \end{aligned}$$

Thus, $E(AG)$ is in one to one correspondence with the set of all vertices of G which are not masters. Note that AG has a rather simple structure: all of its connected components are cycles, each corresponding to a connected component of G^c as follows (this is a direct consequence of the definition of AG):

- If a connected component $H \subset G^c$ is a path, say $v_1, v_2, \ldots, v_l$ (i.e., $\{v_i, v_{i+1}\} \in E(G^c)$, $i = 1, \ldots l-1$), then $V(H)$ defines a cycle C_H that is a connected component in AG:
  If l is even, then the adjacent vertices in C_H are

  $$v_1, v_2, v_4, v_6, \ldots, v_{l-2}, v_l, v_{l-1}, v_{l-3} \ldots, v_3, v_1.$$

  If l is odd, then adjacent vertices in C_H are

  $$v_1, v_2, v_4, v_6, \ldots, v_{l-1}, v_l, v_{l-2}, v_{l-4}, \ldots, v_3, v_1.$$

- If a connected component $H \subset G^c$ is an odd cycle $v_1, v_2, \ldots v_{2k+1}, v_1$ (i.e., $\{v_i, v_{i+1}\} \in E(G^c)$, $i = 1, \ldots 2k+1$, and $\{v_{2k+1}, v_1\} \in E(G^c)$), then $V(H)$ defines a cycle C_H that is a connected component in AG:

  $$v_1, v_3, \ldots, v_{2k+1}, v_2, v_4, \ldots, v_{2k}, v_1.$$

- If a connected component $H \subset G^c$ is an even cycle $v_1, v_2, \ldots, v_{2k}, v_1$ (i.e., $\{v_i, v_{i+1}\} \in E(G^c)$, $i = 1, \ldots 2k$, and $\{v_{2k}, v_1\} \in E(G^c)$), then $V(H)$ defines two disjoint cycles $C_H = C1_H \cup C2_H$ that are connected components in AG:

$$v_1, v_3, \ldots, v_{2k-1}, v_1 \text{ and } v_2, v_4, \ldots, v_{2k}, v_2.$$

Lemma 4.3 *Let c^t such that $sum(c^t) = (n+1)/2$. Let H be a connected component of G^c on l vertices, $l \geq 2$. Let $S = \{v \in H : c^t_v = 1\}$. Then*

$$|\{v \in H : c^{t+1}_v = 1\}| \geq l - |S| \tag{4}$$

Furthermore, the equality holds in (4) if and only if one of the following holds: (i) $|S| = 0$, (ii) $|S| = l$, (iii) H is an even cycle and $c^t_v \neq c^t_w$ whenever $\{v, w\} \in E(G^c)$.

Proof: First note that, by Lemma 4.2 and by definition of AG,

$$|\{v \in H \colon c^{t+1}_v = 1\}| = \sum_{v \in H} c^{t+1}_v = \sum_{\{u,w\} \in C_H \subset AG} (1 - c^t_u c^t_w) = |H| - \sum_{\{u,w\} \in C_H \subset AG} c^t_u c^t_w.$$

Thus, it remains to show that

$$|S| \geq \sum_{\{u,w\} \in C_H \subset AG} c^t_u c^t_w. \tag{5}$$

Note that

$$\sum_{\{u,w\} \in C_H \subset AG} c^t_u c^t_w = |\{\{u, w\} \in E_{AG}(C_H) : u, w \in S\}| = |E(C_H[S])|$$

where $C_H[S]$ denotes the induced subgraph of C_H, i.e., the maximal subgraph of C_H on the vertex set $S \subset V(C_H)$.

If $|S| = 0$, $|E(C_H[S])| = 0$, and (5) holds with equality. Thus, (4) holds with equality.

If $|S| = l$, then $C_H[S] = C_H$ and $|E(C_H[S])| = |E(C_H)| = l$ since C_H is a cycle or a union of two disjoint cycles. Thus, if $|S| = l$, (4) also holds with equality.

If H is an even cycle, then $C_H = C1_H \cup C2_H$. Furthermore, $S = V(C1_H)$ or $S = V(C2_H)$ if and only if vertices of H are colored alternately along the cycle H (i.e., as described in (iii) in the statement of the lemma). In either case, $|E(C_H[S])| = |S|$ and (4) again holds with equality.

If neither (i) nor (ii) nor (iii) holds, then $C_H[S]$ contains an acyclic component and any possible cyclic component of C_H must be a cycle.[5] Thus, $|E(C_H[S])| \leq |S| - 1$ and

$$|\{v \in H : c_v^{t+1} = 1\}| \geq l - |S| + 1 \tag{6}$$

$\square$

Several simple consequences of this lemma will be useful in the analysis that follows. For example, if a connected component of G^c that is not an isolated vertex is monochromatic for some c^t, then every vertex in H will switch color.

Lemma 4.4 *Let c^t be a coloring of G, $\delta(G) \geq n-3$, such that $sum(c^t) = (n+1)/2$. Let $H = (V_H, E_H)$ be a connected component of G^c with $|V_H| \geq 2$. Suppose that $c_v^t = c_w^t$ for every $v, w \in V_H$. Then $c_v^{t+1} = 1 - c_v^t$ for every $v \in V_H$.*

Proof: Let $|V_H| = l$. If $c_v^t = 1$ for all $v \in V_H$, then result follows from Lemma 4.4 with $|S| = l$. If If $c_v^t = 0$ for all $v \in V_H$, then result follows from Lemma 4.4 with $|S| = 0$. $\square$

The next lemma presents an opposite scenario: if colors assigned by c^t, $sum(c^t) = (n+1)/2$, alternate along an even cycle that is a connected component of G^c, then no vertex on that cycle will switch color.

Lemma 4.5 *Let c^t be a coloring of G, $\delta(G) \geq n-3$, such that $sum(c^t) = (n+1)/2$. Let $C_{2k} \subset G^c$ be a connected component in G^c. Suppose the colors assigned by C^t alternate along the cycle: if u is adjacent to v in C_{2k} then $c_u^t = 1 - c_v^t$. Then $c_v^{t+1} = c_v^t$ for every $v \in C_{2k}$.*

Proof: Every $v \in C_{2k}$ is a 2-master and, by (c) of Lemma 4.2, $c_v^{t+1} = 1 - c_u^t c_w^t = 1 - (1 - c_v^t)$, because in C_{2k}, v is adjacent to both u and w. $\square$

The preceding lemmas indicate a way to construct c^0 yielding a complete switch, i.e., $c^1 = 1 - c^0$. Obviously, all masters must be colored with a minority color in order to switch. If all the other connected components of G^c are monochromatic (with some even cycles possibly being

[5] In fact the only possibility for a cyclic component is when H is an even cycle

colored as described in the previous lemma), and if the resulting coloring c^0 is a tight majority coloring on G (i.e., $sum(c^t) = (n+1)/2$), then, as shown in the next lemma, $c^1 = 1 - c^0$ (except on the even cycles where $c^{t=1} = c^t$), and G is not a democratic consensus computer.

Lemma 4.6 *Let $\delta(G) \geq n-3$. Let $H_1 = (V_1, E_1^c)$, $H_2 = (V_2, E_2^c)$, ..., $H_m = (V_m, E_m^c)$ be the connected components of G^c. Suppose there exist i and j, $1 \leq i < j \leq m$, such that*
(i) $|V_k| = 1 \Rightarrow k \leq i$,
(ii) $m \geq k > j \Rightarrow H_k$ is an even cycle,
(iii) $|V_1| + |V_2| + \ldots + |V_i| + 1 = |V_{i+1}| + \ldots + |V_j|$.
Then G is not a democratic consensus computer.

Proof: For $v \in V_k$, set $c_v^0 = 0$ if $k \leq i$ and set $c_v^0 = 1$ if $i < k \leq j$. If $j < m$, then the remaining vertices lie on even cycles in G^c. Color each H_k alternately, i.e., as described in the statement of Lemma 4.5. Note that, by (iii),

$$\begin{aligned} |\{v \in V : c_v^0 = 0\}| &= \textstyle\sum_{k=1}^i |V_k| + \frac{1}{2}\sum_{k=j+1}^m |V_k| \\ &= (\textstyle\sum_{k=i+1}^j |V_k|) - 1 + \frac{1}{2}\sum_{k=j+1}^m |V_k| \\ &= |\{v \in V : c_v^0 = 1\}| - 1. \end{aligned}$$

Thus, $sum(c^0) = (n+1)/2$ and $maj(c^0) = 1$.

If v is a master $c_v^1 = maj(c^0) = 1 = 1 - c_t^0$ (the last equality holds because $\{v\} = H_k$ for some k and $k \leq i$ by (i)). If v is not a master, then $v \in H_k$ for some $k \leq m$ such that $|H_k| \geq 2$. If $k \leq j$, then $c_v^1 = 1 - c_v^0$ by Lemma 4.4. If $k > j$ then $c_v^1 = c_v^0$ by Lemma 4.5. Therefore, $c_v^1 = 1 - c_v^0$ if $v \in V_1 \cup \ldots \cup V_j$ and $c_v^1 = c_v^0$ if $v \in V_{j+1} \cup \ldots \cup V_m$. Thus,

$$\begin{aligned} |\{v \in V : c_v^1 = 1\}| &= \textstyle\sum_{k=1}^i |V_k| + \frac{1}{2}\sum_{k=j+1}^m |V_k| \\ &= (\textstyle\sum_{k=i+1}^j |V_k|) - 1 + \frac{1}{2}\sum_{k=j+1}^m |V_k| \\ &= |\{v \in V : c_v^1 = 0\}| - 1. \end{aligned}$$

So, $maj(c^1) = 0 \neq maj(c^0)$ and G is not a democratic consensus computer by Proposition 2.1. □

For any $k = 0, 1, \ldots (n-1)/2$, it is straightforward to construct a G with k masters satisfying conditions of Lemma 4.6. For example, if $k = 0$, take G such that connected components of G^c are $P_{(n-1)/2}$ and $P_{(n+1)/2}$. If $k > 0$, G whose connected components are its masters, P_{k+1} and C_{n-2k-1} is such example. Thus, there exist G with $\delta(G) \geq n-3$

which are not d.c.c. and having exactly k masters for every $k < (n+1)/2$. (Recall that, by Proposition 3.1, every G with at least $(n+1)/2$ masters is a d.c.c.)

A similar construction to that of Lemma 4.6 yields a class of graphs with a unique master that are not d.c.c.'s

Lemma 4.7 *Let $\delta(G) \geq n-3$ and let v_0 be the unique master in G. Let $H_1 = \{v_0\}$, $H_2 = (V_2, E_2^c)$, ..., $H_m = (V_m, E_m^c)$ be the connected components of G^c. Suppose there exist i and j, $1 \leq i \leq j \leq m$, such that*
(i) $m \geq k > j \Rightarrow H_k$ is an even cycle,
(ii) $|V_2| + |V_3| + \ldots + |V_i| = |V_{i+1}| + \ldots + |V_j|$ (assuming the empty summation on both sides of the equation when $i = j = 1$).
Then G is not a democratic consensus computer.

Proof: Define c^0 as in the proof of Lemma 4.6 except for v_0. Set $c^0_{v_0} = 1$. Observe that $c^1_v = 1 - c^0_v$ for $v \in V_2 \cup \ldots \cup V_j$ (by Lemma 4.4), that $c^1_{v_0} = maj(c^0) = 1 = c^0_{v_0}$ (by Proposition 2.6), and that $c^1_v = c^0_v$ for $v \in V_{j+1} \cup \ldots \cup V_m$ (by Lemma 4.5). Note that $sum(c^1) = (n+1)/2$ because of (ii). Repeating the same observation, we get $c^2_v = 1 - c^1_v = 1-(1-c^0_v) = c^0_v$ for $v \in V_2 \cup \ldots \cup V_j$ (by Lemma 4.4), $c^2_{v_0} = maj(c^1) = 1 = c^0_{v_0}$ (by Proposition 2.6), and $c^2_v = c^1_v = c^0_v$ for $v \in V_{j+1} \cup \ldots \cup V_m$ (by Lemma 4.5). Thus $c^2 = c^0$ and $c^0, c^1, c^2, \ldots$ is periodic with period at most two. Since c^0 is not a consensus, G is not a democratic consensus computer. □

For example, $K_n \setminus C_{n-1}$ is not a d.c.c. because it satisfies the conditions of the lemma with $i = j = 1$ and $m = 2$.

In order to prove that the Master Conjecture holds in the case $\delta(G) \geq n-3$, we need yet another lemma. In what follows we will say that $v_1, v_2, \ldots v_k$ form a path P_k if v_i is adjacent to v_{i+1} in P_k for $i = 1, \ldots, (k-1)$. Similarly, we will say that $v_1, \ldots v_k$ form a cycle C_k if $v_1, \ldots, v_k$ form a path $P_k \subseteq C_k$ and v_1 is adjacent to v_k in C_k.

Lemma 4.8 *Let c^t be a coloring of G, $\delta(G) \geq n-3$, such that $sum(c^t) = (n+1)/2$. Let $v_1, v_2, \ldots, v_k$ form $H \subset G^c$, a connected component in G^c on $k \geq 3$ vertices. Suppose that there exists a $j < k/2$ such that $c^t_{v_i} = i \bmod 2$ for $i \leq 2j+1$. If $2j+1 < k$, also suppose that $c^t_{v_i} = c^t_{v_{2j+2}}$ for $i > 2j+1$.*

Then $c^{t+1}_{v_i} = c^t_{v_i}$ for $i \leq 2j+1$ and $c^{t+1}_{v_i} = 1 - c^t_{v_i}$ for $i > 2j+1$.

Proof: Since $\delta(G) \geq n-3$, H is a path or a cycle. Using (b) and (c) of Lemma 4.2, observe that $c^{t+1}_{v_i} = c^t_{v_i}$ for $i \leq 2j+1$ (since each v_i such that $c^t_{v_i} = 0$ has both non-neighbors of color 1, while each v_i such that $c^t_{v_i} = 1$ has at least one non-neighbor of color 0) and that $c^{t+1}_{v_i} = 1 - c^t_{v_i}$ for $i > 2j+1$ (if $c^t_{v_{2j+2}} = \ldots = c^t_{v_k} = 0$, then each such v_i has a non-neighbor of color 0; if $c^t_{v_{2j+2}} = \ldots = c^t_{v_k} = 1$, then each such v_i has all non-neighbors of color 1 because $c^t_{v_1} = c^t_{v_{2j+1}} = 1$). $\square$

Theorem 4.9 *Let G such that $\delta(G) \geq n-3$. If G is a democratic consensus computer, then G contains a master.*

Proof: Suppose G does not contain a master. We'll show that G is not a democratic consensus computer. Let $H_1 = (V_1, E^c_1)$, $H_2 = (V_2, E^c_2)$, ..., $H_m = (V_m, E^c_m)$ be the connected components of G^c. Since G does not contain a master, $|V_l| \geq 2$, $l = 1, \ldots m$. Choose an index i such that

$$|V_1| + \ldots + |V_i| \leq (n-1)/2 < |V_1| + \ldots + |V_i| + |V_{i+1}|.$$

If $|V_1| + \ldots + |V_i| = (n-1)/2$, then the conditions of Lemma 4.6 are satisfied with i, and with $j = m$. Therefore, in this case, G is not a democratic consensus computer.

For the rest of the proof we may assume that $|V_1| + \ldots + |V_i| < (n-1)/2$. We may also assume that $|V_1| + \ldots + |V_{i-1}| + |V_i| + (|V_{i+1}|/2) > (n-1)/2$. (If not, then $(|V_{i+1}|/2) + |V_{i+2}| + |V_{i+3}| + \ldots + (|V_m|) > (n-1)/2$ and we could map l to $m+1-l$, i.e. H_l becomes H_{m+1-l}, $l = 1, \ldots, m$.) Note that these imply that $|V_i| \geq 3$.

Let $v_1, v_2, \ldots, v_k$ form H_{i+1} and let

$$j = (n-1)/2 - (|V_1| + \ldots + |V_{i-1}| + |V_i|) \tag{7}$$

Note that $j < k/2$. Set

$$c^0_v = \begin{cases} 0 & v \in V_1 \cup V_2 \cup \ldots \cup V_i \\ i \mod 2 & v_i, \quad i = 1, \ldots, 2j+1 \\ 1 & v_i, \quad i = 2j+2, \ldots, k \\ 1 & v \in V_{i+2} \cup V_{i+3} \cup \ldots \cup V_m \end{cases}$$

Note that $sum(c^0) = (n+1)/2$. By Lemma 4.4, $c^1_v = 1 - c_v$ for every $v \notin V_{i+1}$. By Lemma 4.8, $c^1_{v_i} = 1 - c^0_{v_i}$ for $i = 2j+2, \ldots, k$ and $c^1_{v_i} = c^0_{v_i}$ for $i = 1, \ldots 2j+1$. Thus, only j vertices colored by 0 and only $j+1$ vertices colored by 1 do not switch color. Hence, $sum(c^1) = |V_1| + \ldots + |V_i| + (j+1) = (n-1)/2 + 1 = (n+1)/2$ (the second equality follows from (7)).

Repeating the same argument for

$$c_v^1 = \begin{cases} 1 & v \in V_1 \cup V_2 \cup \ldots \cup V_i \\ i \mod 2 & v_i, \quad i = 1, \ldots, 2j+1 \\ 0 & v_i, \quad i = 2j+2, \ldots, k \\ 0 & v \in V_{i+2} \cup V_{i+3} \cup \ldots \cup V_m \end{cases}$$

we conclude that $c^2 = c^0$. Thus, $c^0, c^1, c^2, \ldots$ has period two. Therefore, G is not a democratic consensus computer. $\square$

Next we turn to G, $\delta(G) \geq n-3$, which contain masters. Because of Proposition 3.1, the only remaining cases are graphs with k masters, $k = 1, 2, \ldots (n-3)/2$. We have already demonstrated two conditions that would immediately classify such G as not being d.c.c. (Lemma 4.6 and Lemma 4.7). As the next theorem shows, these are the only two obstacles.

Theorem 4.10 *Let G, $\delta(G) \geq n-3$, contain exactly k masters, $1 \leq k \leq (n-3)/2$. G is not a d.c.c. if and only if G satisfies conditions of Lemma 4.6 or conditions of Lemma 4.7.*

Proof: We only have to prove necessity. (Sufficiency follows from Lemma 4.6 and Lemma 4.7.) We will show that, for any c^0, G either admits a democratic consensus for c^0 or satisfies conditions of either Lemma 4.6 or conditions of Lemma 4.7. By Proposition 4.1 we may assume $sum(c^0) = (n+1)/2$.

Let H_1, H_2, ..., H_l be the connected components of G^c that are not isolated vertices. Let c^0 be a coloring of G, $sum(c^0) = (n+1)/2$. For $i = 0, 1$ define

$$m(i) = |\{v \in G : v \text{ is a master}, c_v^0 = i\}|$$

and

$$h_j(i) = |\{v \in H_j : c_v^0 = i\}|,$$

$j = 1, \ldots, l$. Note that $k = m(0) + m(1)$, $|H_j| = h_j(0) + h_j(1)$, $(n+1)/2 = sum(c^0) = m(1) + \sum_{j=1}^{l} h_j(1)$ and that $(n-1)/2 = m(0) + \sum_{j=1}^{l} h_j(0)$.

Furthermore, for $j = 1, \ldots, l$, let $\alpha_j = 0$ if H_j satisfies (i), (ii), or (iii) from Lemma 4.3; otherwise let $\alpha_j = 1$. In this notation, by Lemma 4.3, we have

$$|\{v \in H_j : c_v^1 = 1\}| \geq |H_j| - h_j(1) + \alpha_j = h_j(0) + \alpha_j.$$

Thus, taking into account that $c_v^1 = 1$ for every master v (Proposition 2.6),

$$sum(c^1) \geq m(0) + m(1) + \sum_{j=1}^{l}(h_j(0) + \alpha_j).$$

Thus,

$$\begin{aligned} sum(c^1) - sum(c^0) &\geq m(1) + m(0) + \textstyle\sum_{j=1}^{l} h_j(0) + \sum_{j=1}^{l} \alpha_j - sum(c^0) \\ &\geq m(1) + (n-1)/2 + \textstyle\sum_{j=1}^{l} \alpha_j - (n+1)/2 \\ &\geq m(1) - 1 + \textstyle\sum_{j=1}^{l} \alpha_j \end{aligned}$$

Therefore, $sum(c^1) < sum(c^0)$ if and only if $m(1) = 0$ and $\alpha_j = 0$ for all $j = 1, \ldots, n$. Also, in this case, $sum(c^1) = (n-1)/2$, i.e., G is not a d.c.c. since $maj(c^1) = 0 \neq 1 = maj(c^0)$. Note that conditions of Lemma 4.6 are satisfied: list all components of G^c as follows: start with all masters and continue with all H such that $c_v^0 = 0$ for all $v \in H$; then continue by listing all H such that $c_v^0 = 1$ for all $v \in H$; if there are remaining components, these must be even cycles colored alternately along the cycle by c^0.

If $sum(c^1) > sum(c^0)$, then $sum(c^1) \geq (n+3)/2$ and, therefore, G admits democratic consensus for c^1 by Proposition 4.1(and thus for c^0 also since $maj(c^0) = maj(c^1)$).

Therefore, we may assume that $sum(c^1) = sum(c^0) = (n+1)/2$. This also means that we may assume that

$$1 - m(1) = \sum_{j=1}^{l} \alpha_j.$$

Since the righthand side is nonnegative, $m(1) = 0$ or $m(1) = 1$. Note that we may assume that $m(1) = 1$. (For otherwise, $m(1) = 0$ implies $m(0) > 0$ since G has at least one master and that there is exactly one H_j such that $\alpha_j = 1$. Note that $d^0 = c^1$ has $sum(d^0) = (n+1)/2$, unchanged α_j for every j, and, because of Proposition 2.6, $m(0) = 0$ and $m(1) > 0$. Thus, we can replace c^0 by d^0 and ensure that $m(1) \neq 0$. This can be done because G admits democratic consensus for c^0 if and only if G admits democratic consensus for d^0.)

Because $m(1) = 1$, we know that every $\alpha_j = 0$, i.e., H_j is either colored monochromatically by c^0 or H_j is an even cycle colored alternately by c^0. If, in addition, $m(0) = 0$, G has only one master and the conditions of Lemma 4.7 are satisfied: list all components of G^c as follows: start with the unique master, continue with all H such that $c_v^0 = 0$ for every $v \in H$, and then continue by listing all H such that $c_v^0 = 1$ for every $v \in H$; if there are any vertices in H left, these must be even cycles colored alternately.

Therefore, it remains to consider the case $sum(c^0) = sum(c^1) = (n+1)/2$, $m(1) = 1$, $m(0) \geq 1$ and $\alpha_j = 0$ for every H_j. Note that

$$\begin{aligned} sum(c^2) &= m(0) + m(1) + \textstyle\sum_{j=1}^{l} h_j(1) \geq 1 + m(1) + \sum_{j=1}^{l} h_j(1) \\ &= 1 + sum(c^0) = (n+3)/2. \end{aligned}$$

Thus, by Proposition 4.1, c^3 is a consensus in color 1. □

Remark. The proof of Theorem 4.10 shows that if G, with $\delta(G) \geq n-3$, is a d.c.c., then c^3 is the democratic consensus, i.e., the local majority process reaches consensus in at most three steps.

Theorem 4.11 *Let G be a graph with k masters and with $\delta(G) \geq n-3$. G is not a democratic consensus computer if and only if one of the following holds: (i) $k = 0$, (ii) $k = (n-1)/2$, (iii) G satisfies conditions Lemma 4.6, (iv) G satisfies the conditions of Lemma 4.7.*

Proof: Theorem 4.9, Proposition 2.6, and Theorem 4.10. □

Note that Theorem 4.10 can be used to define various classes of d.c.c.'s and non-d.c.c.'s. We close this section by observing just one additional class of d.c.c's that was mentioned in Section 2.

Theorem 4.12 *Let $k \geq 1$, $k \neq (n-1)/2$. There exist a democratic consensus computer with exactly k masters. In particular, $K_n \setminus P_{n-k}$ is a democratic consensus computer.*

Proof: If $k \geq (n+1)/2$, the result follows from Proposition 2.6. If $1 \leq k \leq (n-3)/2$, the connected components of $G = K_n \setminus P_{n-k}$ are k single element components corresponding to masters in G and P_{n-k}. Note that G does not satisfy conditions of Lemma 4.6, nor conditions of Lemma 4.7. Thus, by Theorem 4.10, G is a democratic consensus computer. □

5 Generalizations and Relaxations

A simple generalization of the local majority process would allow vertex v to have some resistivity towards color switch. Formally, for a nonnegative integer k_v, we define a k_v-*local majority rule* for vertex v:

$$c_v^{t+1} = \begin{cases} c_v^t & \text{if } |\{w \in N_v : c_w^t = c_v^t\}| \geq |N(v)|/2 + k_v \\ 1 - c_v^t & \text{if } |\{w \in N_v : c_w^t \neq c_v^t\}| > |N(v)|/2 + k_v \end{cases} \qquad (8)$$

The value k_v is called the *resistivity value* of vertex v and we call the graph $G = (V, E)$ together with the set of vertex resistivities $\{k_v : v \in V\}$ a *varied-resistivity* graph. Similarly, the process defined by (8) is called the local majority process with resistivities. Note that the local majority process with resistivities where $k_v = 0$, $v \in V$, is exactly the local majority process. As the next theorem shows, introducing vertex resistivities does not bring additional difficulties: the local majority process with resistivities can be simulated by the (standard) local majority process.

Theorem 5.1 *Let $G(V, E)$ be a varied-resistivity graph, with the vertex set $V = \{v_1, \ldots, v_n\}$, and the corresponding resistivities $R = \{k_{v_1}, \ldots, k_{v_n}\}$. The local majority process with resistivities on the varied-resistivity graph G can be simulated by the local majority process on some graph $G'(V', E')$.*

Proof: We will derive the graph $G'(V', E')$ iteratively. Let the graph $G_0(V_0, E_0) = G(V, E)$, where the vertices of G_0 are labeled differently, i.e. $V_0 = \{v_1^1 = v_1, \ldots, v_n^n = v_n\}$. The edges are the same. Also, let c_0 be the coloring defined on G, and let the corresponding coloring for G_0 be c_0^0.

We now show how to derive G_i from G_{i-1}. The vertices of the graph $G_{i-1}(V_{i-1}, E_{i-1})$ are $V_{i-1} = \{v_1^1, \ldots, v_1^m, \ldots, v_n^1, \ldots, v_n^m\}$, where $k_{v_l^j} = k_{v_l^{j'}}$ for all $1 \leq j, j' \leq m$ and for all l. If $k_{v_i^j} = 0$, then we set $G_i = G_{i-1}$, i.e. $V_i = V_{i-1}$ and $E_i = E_{i-1}$. Otherwise, we do the following.

For ease of notation, we refer to $k_{v_i^j}$ as k_i, since all v_i^j vertices have the same resistivity by assumption. For each vertex $v_l^j \in V_{i-1}$, we add k_i new vertices. We shall denote these k_i new vertices for each vertex $v_l^j \in V_{i-1}$ as v_l^d, where $d = (m + (j-1)k_{v_l^j} + 1), \ldots, (m + jk_{v_l^j})$. We shall call v_l^j the *parent* vertex of all such k_i new v_l^d vertices. More formally, $V_i = V_{i-1} \cup_{u=1}^{k_i} \{v_l^{m+(j-1)k_i+u} : v_l^j \in V_{i-1}\}$. The resistivity values of all the vertices are reassigned as

$$k_{v_l^j} = \begin{cases} 0 & \text{if } \ l = i, \text{ or } v_i^{j'} \text{ is parent of } v_l^j, \text{ for any } j' \\ k_{v_l^j} & \text{else if } \ l \neq i, \text{ or } v_i^{j'} \text{ is not a parent of } v_l^j, \text{ for all } j' \end{cases} \qquad (9)$$

That explains all the vertices we add to the graph. We add the edges to the new graph as follows. First, for each vertex v_i^d, $(m + (j-1)k_i) \leq$

$d \leq (m + jk_i)$, we add the edge (v_i^d, v_i^j), where v_i^j is the parent vertex of the vertex v_i^d. For each pair of vertices v_i^d, v_i^e, for all $(m+(j-1)k_i) \leq d < e \leq (m + jk_i)$, we add the edges (v_i^d, v_i^e) to E_i. Then, for every pair of vertices $v_l^d, v_{l'}^d$ such that the parent of v_l^d is v_l^j and the parent of $v_{l'}^d$ is $v_{l'}^j$, we add the edge $(v_l^d, v_{l'}^d)$ to E_i whenever $(v_l^j, v_{l'}^j) \in E_{i-1}$. More formally,

$$\begin{aligned} E_i &= E_{i-1} \cup (\{(v_i^d, v_i^j) : (m + (j-1)k_i) \leq d \leq (m + jk_i)\}) \\ &\qquad\qquad \cup \\ &\quad (\cup_{j=1}^{m} \{(v_i^d, v_i^e) : (m + (j-1)k_i) \leq d < e \leq (m + jk_i)\}) \\ &\qquad\qquad \cup \\ &\quad (\cup_{u=1}^{k_i} \{(v_l^{m+(j-1)k_i+u}, v_{l'}^{m+(j'-1)k_i+u}) : (v_l^j, v_{l'}^{j'}) \in E_{i-1}). \end{aligned}$$

This completes the construction of the new graph. Note that $|V_i| = k_i|V_{i-1}|$. Also, by the assignment of resistivities, $k_{v_i^j} = 0$ for all j.

We now extend the coloring c_{i-1}^0 for the coloring c_i^0 of the new graph simply by making the color of each new vertex as the color of its parent vertex. More formally, $c_i^0 = \{c_i^0(v_l^{m+(j-1)k_i+u}) = c_{i-1}^0(v_l^j) : 1 \leq u \leq k_i, 1 \leq j \leq m\}$.

Note that now c^1 according to the local update procedure can be found on G_i by extending c^1 from V_{i-1} to V_i, or by applying the local update on c^0 in G_i directly.

Finally, note that G_n has $d|V|$ vertices, where $d = \prod_{u=1}^{n} k_u$. Also, since for the graph G_i, $k_j = 0$ for all $j \leq i$, G_n has the resistivity values of all vertices as zero. At each step, G_i simulates G_{i-1}, and thus G_n simulates $G_0 = G$. $\square$

Another natural generalization of our model would be to associate weights to the edges of G. The weight a_{uv} associated to the edge $\{u, v\}$ could be interpreted as the strength of the relationship between u and v and used as the relative importance of the info provided by $u \in N(v)$ at time t for v's decision on its color c_v^{t+1} at time $t+1$. In other words, the *weighted local majority process* is defined by

$$c_v^{t+1} = 1 - c_v^t \Leftrightarrow \left(\sum_{u \in N(v): c_u^t = 1 - c_v^t} a_{uv} \right) > \frac{1}{2} \sum_{u \in N(v)} a_{uv}.$$

Note that we may assume that all a_{uv} are rational numbers because G is finite[6]. Furthermore, we may assume that all weights a_{uv} are integer

[6] There exists $\epsilon > 0$ such that replacing each weight a_{uv} by a_{uv}^*, $a_{uv} - \epsilon < a_{uv}^* \leq a_{uv}$, yields the same process for any initial coloring c^0.

valued[7]. If all weights happen to be nonnegative, the weighted local majority process can be simulated by the local majority process on the corresponding multigraph $G^M = (V, E^M)$ where E^M is the multiset of edges of G with multiplicity of an edge $\{u, v\}$ given by a_{uv}. In order to properly apply (1) in this case, several definitions have to be adjusted: $N(v)$ is the multiset of the vertices adjacent to v where a vertex u appears with multiplicity a_{uv}. A degree of a vertex v, $deg(v)$ is the cardinality (taking into account multiplicities) of $N(v)$. Most of our results readily generalize in the multigraph framework.

Combining both generalizations yields the *local weighted majority process with resistivities* which is defined by:

$$c_v^{t+1} = 1 - c_v^t \Leftrightarrow \left(\sum_{u \in N(v): c_u^t = 1 - c_v^t} a_{uv} \right) > k_v + \left(\sum_{u \in N(v): c_u^t = c_v^t} a_{uv} \right). \quad (10)$$

In other words, v changes its color at time $t+1$ if and only if at least k_v more than the weighted majority of its neighbors are colored by $1 - c_v^t$ at time t. As already noted, it is safe to assume that edge weights a_{uv} are integral, and thus, without loss of generality we may assume that resistivities k_v are integral also[8]. If k_v is nonnegative, this process can be simulated by the local majority process on the multigtraph G^M defined as in the previous paragraph with addition of k_v loops to each vertex v. Note that v belongs to $N(v)$ with multiplicty k_v in such a multigraph.

Hence, because of the multigraph simulation, it is straightforward to generalize most of the presented results in the framework that allows for nonnegative edge weights a_{uv} and for nonnegative vertex weights k_v. Note that the weighted local majority process with resistivities can be described in terms of the process from the statement of Theorem 2.2 with $b = A(1, \dots, 1)^T$ where nondiagonal entries of the integer valued A are a_{uv} if $\{u, v\}$ is an edge and $a_{uv} = 0$ if $\{u, v\}$ is not an edge and where the diagonal entries of A are $a_{vv} = k_v + 1$ if there exists c^t which would turn the righthand side of (10) into equality, and $a_{vv} = k_v$ otherwise. Conversely, if A is a nonnegative matrix and b such that, for every $i = 1, \dots n$, the interval $(b_i - a_{ii}, b_i)$ contains no elements from

$$S = \{\sum_{j \in J} a_{ij} : J \subseteq \{1, 2 \dots, n\} \setminus \{i\}\}$$

[7]Multiplying all weights by a scalar λ yields the same process. Set λ to the common denominator of all weights.

[8]Any k_v can be replaced by an integral k_v^* resulting in no change in the coloring process.

and $b_i \geq (\max S)/2$, then it is straightforward to define G and non-negative edge and vertex weights such that the weighted local majority process with resistivities on G is equivalent to the dynamic process in the symmetric neural network model (i.e., the process described in the statement from Theorem 2.2). Finally, note that there are instances of the general symmetric neural network process, e.g., a weighted local majority process with resistivities where any of the edge weights or vertex resistivities are negative, that cannot be simulated by the multigraph simulation approach to the local majority process. (Allowing negative weights and resistivities could be viewed as a technical generalization but not necessarily as a natural one. If at least one of these parameters is negative, there exists v and c^0 such that v switches its color at time $t = 1$ from $c_v^0 =$ the majority color of its neighborhood at time $t = 0$, to $c_v^1 = 1 - c_v^0 =$ the mi! nority color of its neighborhood at time $t = 0$. A rule that allows for a switch from the local majority to the local minority hardly qualifies as an acceptable majority computation rule.)

In the next section we further discuss some basic assumptions of our model and try to illustrate why our model is a natural one to analyze. In the rest of this section we present an approach toward relaxing the notion of democratic consensus computer.

In view of our results showing that democratic consensus computers are nowhere truly local, one might want to know how likely is a network of agents G to admit a democratic consensus. Here we single out and then combine two possible ways to measure this. On the one hand, one might be interested only in the colorings where the difference between majority and minority is substantial, i.e., we may only require that G admits democratic consensus only for c^0 such that $sum(c^0) \geq k$ (where an integer $k \geq n/2$ is a numerical expression of "substantial majority"). On the other hand, one might allow for occasional failures requiring that G admits democratic consensus for substantial proportion p of initial colorings c^0 that are of interest. These two approaches motivate the following definition of the (p, k)-weak democratic consensus computer that is defined for $0 \leq p \leq 1$ and integer $k > n/2$. G is a (p, k)-*weak democratic consensus computer* if G ad! mits a democratic consensus for at least $p|C(k)|$ colorings $c^0 \in C(k) = \{c^0 : sum(c^0) \geq k\}$. Note that, for odd n, d.c.c. is equivalent to $(1, (n+1)/2)$-weak d.c.c. Also note that, if G is a (p, k)-weak d.c.c. and (p', k')-weak d.c.c., then $k \leq k' \Rightarrow p \leq p'$. (This follows from Lemma 2.4.) Further note that, by Proposition 2.6, any G with $\delta(G) \geq 2(n-k)$ is a $(1, k)$-weak democratic consensus computer. In fact, we believe that the following generalization of the master conjecture holds.

Generalized Master Conjecture. Every $(1, k)$-weak democratic consensus computer contains a $(2k - n - 1)$-master.

6 Discussion of model assumptions

As already noted, one might think that our model is not a realistic one nor a natural one to study because of several assumptions that we have made. In this section we discuss model assumptions and hopefully illustrate why our model is a natural one to study.

Choice of the neighborhood and tie-breaking rule. One might consider our choice of the tie-breaking rule and the definition of the neighborhood somewhat arbitrary. For example, why not redefine $N(v)$ by including v itself in $N(v)$ and/or modify the tie-breaking rule so that $c_v^{t+1} = 1 - c_v^t$ whenever $|\{w \in N(v) : c_w^t = 1\}| = |N(v)|/2$ (i.e., require that v switches color in the case of a tie in the neighborhood). Note that adopting both changes in our model does not change the process. Also note that the proposed modification of the tie-breaking rule (without redefining $N(v)$) allows no democratic consensus computers[9]. Other modifications can be found in the literature (e.g., several variants are studied in [P96b]).

Since it is impossible to discuss all possible creative proposals for the modifications of our model, let's discuss some properties that a reasonable model should have, and then show that our model is the only one satisfying these properties.

The least one should expect from a local majority process is that every vertex v should be able to update its color c_v^t so that c_v^{t+1} is the majority color among the colors it is aware of, i.e., c_v^{t+1} should be the majority color on $N(v) \cup \{v\}$ at time t. In the case where the majority is not defined, the update should reflect that ambivalence, i.e., if there is a tie among colors that v is aware of at time t, then $c_v^{t+1} = 1 - c_v^t$. In other words, c_v^{t+1} should be computed to reflect the majority situation in $N(v) \cup \{v\}$ at time t (majority is 0 or 1 or ambivalent) because v has no information about possible existence of vertices that are not itself nor in $N(v)$. Thus, if v happens to be a master, thereby having no reasons

[9] Let $c_v^0 = 1$ if and only if $v \in S$ where (S, S^c) is a max-cut in G. If the tie-breaking rule is redefined as described, then $c^1 = 1 - c^0$ (because $deg_{S^c}(v) \geq deg_S(v)$ for every $v \in S$).

for faulty computations of global majority, c_v^{t+1} will correctly signal the global majority[10]. Note that the stated properties uniquely define

$$c_v^{t+1} = f(\{c_w^{t+1} : w \in N(v) \cup \{v\}\})$$

and (1) is a way to represent c^{t+1}. Therefore, if the goal is to define a local update step satisfying outlined properties, the only choice is the local update used in our model.

Bidirectional communication. The bidirectional nature of the relationship among the agents played a crucial rule in our analysis. For example, even the basic *"period is one or two"* property does not hold when G is allowed to be a directed graph. For example, if $\vec{C_n}$ is a directed cycle on n vertices and c^0 is the coloring assigning 1 to only one vertex and 0 to the remaining $n-1$ vertices, $c^0, c^1, c^2, \ldots$ is periodic with period n. Thus, allowing for nonsymmetric relationships yields to periods of any possible length. In order to generalize presented results, one would have to take into account the possibility of periods longer than two.

Memoryless property. The memoryless property of the local majority process might seem unreasonable in many applications. In this paper we investigated iterative use of the local majority rule as the simplest local approach to the problem of determining global majority. Limited computational power of the agents due to the memoryless property of the process and the agent's ability to calculate and communicate the local majority in the form of a one-bit information is of central importance in our analysis. Empowering agents with memory would bring the problem closer to the standard distributed computing framework. Design and analysis of possibly more successful and more complicated protocols of the distributive computing flavor is beyond the scope of this paper. Here we only note that the problem of determining majority becomes trivial if all agents are aware of the network structure. (If G is disconnected, there is no way to communicate between two connected components. If G is connected, the in! fo about c_v^0 can be propagated through the network. This could be repeated for all n vertices which would allow all agents to learn c^0 and thus $maj(c^0)$). Thus, the interesting protocols would be those defined for agents that have no unique IDs and have no info about the network.

Static network structure. The static nature of the network of agents is another critical property of the local majority process. It is

[10]If, for a master v, $c_v^{t+1} = 1 - c_v^t$, one has to check c_v^{t+2} to determine if the color switch indicated ambivalence or the choice of the global majority color.

possible that allowing for network dynamics in the form of protocols that simultaneously control changes in c^t and the structure of the network at the time t (e.g., changing the weight of an edge; adding/deleting an edge) might yield efficient protocols. This seems to be a fundamentally different model than the one studied here.

Synchronous vs. asynchronous updates. Synchronous updates make the local majority process less restrictive than it would be with possible asynchronous update protocols. If the local majority process is modified in a way that an infinite sequence $v_1, v_2, \ldots$ of vertices from G is given and that the only update of c^t at time t occurs at vertex v_t according to the local majority rule (1) while $c_v^{t+1} = c_v^t$ for all $v \neq v_t$, then no G except the complete graph on odd number of vertices can be a democratic consensus computer. (First note that v_1 must be a master with $deg(v) = n - 1$ even to ensure that $maj(c^0) = maj(c^1)$ for all colorings c^0. Thus, by induction, all vertices appearing in the sequence must be masters with even degree. If a vertex v does not appear in the sequence, then G cannot be a d.c.c. because an update at v will never occur.)

Deterministic vs. stochastic model. The presented model is purely deterministic and there are several aspects of the model that call for stochastic modification. For example, it would be interesting to see what effect would replacing the local update (1) by the stochastic update rule

$$\mathtt{P}(c_v^{t+1} = i) = \frac{1}{deg(v)} |\{w \in N(v) : c_w^t = i\}|$$

have on the conclusions drawn from the model. (The same rule is used in [HP99, HP00, NIY99, NIY00].)

Also, allowing for asynchronous updates where the next vertex to be updated is selected at random could yield interesting results. However, one has to be aware that stochastic rules allow for a nonzero probability of not admitting a democratic consensus.

Also, allowing for asynchronous updates where the next vertex to be updated is selected at random could yield interesting results. However, one has to be aware that stochastic rules allow for a nonzero probability of not admitting a democratic consensus.

Number of colors. One might consider generalizing the model by allowing k possible colors, i.e., allowing $c^t : V \to \{0, 1, \ldots, k-1\}$. Proper definition of the tie-breaking rule is an inherent problem of this generalization. If $k > 2$, it is possible that c_v^t is a minority color in $N(v) \cup \{v\}$ and that there is more than one majority color in $N(v)$. Then any choice for

c_v^{t+1} will favor one of the majority colors even though info available, c_w^t, $w \in N(v)$ does not. Regardless of the definition of the tie-breaking rule that would hopefully generalize the one used in our model, understanding the case $k = 2$ is a prerequisite for understanding models allowing $k > 2$ colors. (Clearly, if G is a democratic consensus computer when k colors are possibly present, G is also a democratic consensus computer when $k' < k$ are possibly present.)

A generalization that would be more along the lines of our approach would be to allow $c^t : V \to \mathbf{R}$, define the dynamic process by

$$c_v^{t+1} = \sum_{w \in N(v)} c_w^t$$

and stating that G admits a democratic consensus for c^0 if there exists a t such that $sign(c_v^t) = sign(sum(c^0))$ for every $v \in V$. A minimalistic version of this generalization would be to allow $c^t : V \to \{-1, 0, 1\}$, define define the dynamic process by

$$c_v^{t+1} = sign(\sum_{w \in N(v)} c_w^t)$$

and stating that G admits a democratic consensus for c^0 if there exists a t such that c^t is a consensus with $(c_v^t) = sign(sum(c^0))$ for every $v \in V$. (Note that the local majority process differs from this generalization only in the tie-breaking rule: if initial colorings are restricted to $c^0 : V \to \{-1, 1\}$, then $c_v^1 = 0$ if and only if the number of 1's and -1's is equal in $N(v)$.) For both generalizations it is straightforward to generate results that show that G cannot be a democratic consensus computer if there exists a partition of G, similar to that described in (b) of Theorem 3.2. For example, no bipartite graph can be a democratic consensus computer in either of the two generalizations.

7 Conclusions and Directions

The main result of this paper is that failure-free computation of democratic consensus by iterative applications of the local majority rule is possible only in the networks that are nowhere truly local (Theorem 3.5). In other words, the idea of solving a truly global task (reaching consensus on majority) by means of truly local computation only (local majority rule) is doomed for failure.

However, even well connected networks of agents that are nowhere truly local might fail to reach democratic consensus when iteratively applying the local majority rule. We have investigated the properties of democratic consensus computers, i.e., the networks in which iterative application of the local majority rule always yields consensus in the initial majority state.

There are several directions that might be of potential interest. One such direction that was not of our interest here is to determine the complexity of the decision problem:

> **DCC**. Input is a finite graph G. Is G a democratic consensus computer?

Clearly, DCC is in co-NP because of Theorem 3.2 and it is very likely that DCC is co-NP complete. However, subclasses of DCC are in P: for example, if the input in DCC is restricted to G with $\delta(G) \geq n-3$, then it is not hard to construct a polynomial time dynamic programming algorithm that counts the number of masters and then, if necessary, determines whether conditions of either Lemma 4.6 or Lemma 4.7 are satisfied, i.e., determines, using Theorem 4.11, whether G is a d.c.c. or not.

The direction that would be more along the lines of our work would be quest for the full characterization of democratic consensus computers. We have made a first step towards possible characterization theorem by characterizing democratic consensus computers for networks that are almost complete in the sense that every agent does not communicate with at most two other agents (Theorem 4.11). A simpler task would be to determine interesting properties of democratic consensus computers that fail short of characterization. For example, we have shown, by the exhaustive computer aided search, that in every democratic consensus computer on at most 13 agents there exists an agent that communicates with all other agents. In fact, we conjecture that every democratic consensus computer G contains a master, i.e., there exists $v \in V(G)$ such that $d(v) = |V(G)| - 1$ (see Master Conjecture in Section 3). We have shown that this conjecture ho! lds for almost complete networks, i.e. networks that are in a way most natural candidates for a counterexample to the conjecture (Theorem 4.9). However, the Master Conjecture remains open.

References

[AW98] H. Attiya and J. Welch, *Distributed Computing: Fundamentals, Simulations and Advanced Topics.* McGraw-Hill, England, 1998.

[B99] E. Berger, Dynamic monopolies of constant size, *Manuscript.* http://xxx.lanl.gov/abs/math/9911125.

[B87] G. Bracha, An $o(\log n)$ expected rounds randomized Byzantine generals algorithm, *Journal of the ACM*, 910-920, 1987.

[D85] S. B. Davidson, H. Garcia-Molina, D. Skeen, Consistency in partitioned networks, *ACM Computing Surveys*, vol 3, 341-370, 1985.

[D74] M. H. Degroot, Reaching a consensus, *Journal of the American Statistical Association*, vol 69, 167-182, 1974.

[DP96] K. Diks, D. Pelc, System diagnosis with smallest risk of error, *22nd Int. Workshop on Graph-Theoretic Concepts in Computer Science*, 1996.

[DPPU88] C. Dwork, D. Peleg, N. Pippenger, E. Upfal, Fault tolerance in networks of bounded degree, *SIAM J. of Computing*, vol 17, 975-988, 1988.

[FLLPS98] P. Flocchini, E. Lodi, F. Luccio, L. Pagli and N. Santoro, Irreversible dynamos in tori, *Proc. EUROPAR*, 554-562, 1998.

[FLLPS99] P. Flocchini, E. Lodi, F. Luccio, L. Pagli and N. Santoro, Monotone dynamos in tori, *Proc. 6th Colloq. on Structural Information & Communication Complexity*, 1999.

[F56] J. R. P. French, A formal theory of social power, *Psych. Review 63,* 181-194, 1956.

[GB85] H. Garcia-Molina, D. Barbara, How to assign votes in a distributed system, *J. of ACM*, vol 5, 32:4(841-860), 1985.

[G79] D. K. Gifford, Weighted voting for replicated data, *Proc. 7th Operating System Principles*, 150-159, 1979.

[G86] E. Goles, Positive automata networks, *Disordered Systems and Biological Organization*, Springer-Verlag, 101-112, 1986.

[GM90] E. Goles, S. Martinez, *Neural and Automata Networks*, Kluwer Academic Publishers, 1990.

[G080] E. Goles, J. Olivos, Periodic behavior of generalized threshold functions, *Discr. Math.*, vol 30, 187-189, 1980.

[GO81] E. Goles, J. Olivos, Comportement Periodique des Fonctions a Seuil Binaires et Applications, *Discr. App. Math.*, vol 3, 95-105, 1981.

[H59] F. Harary, A criterion for unanimity in French's theory of social power. In D. Cartwright, ed., *Studies in social power,* (Inst. Soc. Res., Ann Arbor, MI) 168-182, 1959.

[H98] Y. Hassin. Probabilistic local polling processes in graphs, *M.Sc. Thesis*, The Weizmann Institute, Rehovot, Israel, 1998.

[HP99] Y. Hassin and D. Peleg, Distributed probabilistic polling and applications to proportionate agreement, *Proc. 26th International Colloq. on Automata, Languages, and Programming*, 402-411, 1999.

[HP00] Y. Hassin and D. Peleg, Extremal bounds for probabilistic polling in graphs, *Proc. 7th International Colloq. on Structural Information and Communication Complexity*, 2000.

[H84] M. P. Herlihy, Replication Methods for Abstract Data Types, *PhD thesis*, MIT, 1984.

[KP95a] S. Kutten, D. Peleg, Fault-local distributed mending, *Proc. 14th ACM Symp. on Principles of Distributed Computing*, 1995.

[KP95b] S. Kutten, D. Peleg, Tight fault-locality, *Proc. 36th IEEE Symp. on Foundations of Computer Science*, 1995.

[LSP82] L. Lamport, R. Shostak, M. Pease, The Byzantine generals problem, *ACM Transactions of Programming Languages and Systems*, vol 4, 1982.

[LPS99] F. Luccio, L. Pagli, and H. Sanossian, Irreversible dynamos in butterflies, *Proc. 6th International Colloq. on Structural Information & Communication Complexity*, 1999.

[M94a] G. Moran, The r-majority vote action on 0-1 sequences, *Discr. Math.*, vol 132, 145-174, 1994.

[M94b] G. Moran, Parametrization for stationary patterns of the r-majority operators on 0-1 sequences, *Discr. Math.*, vol 132, 175-195, 1994.

[M94c] G. Moran, On the period two property of the majority operator in infinite graphs, *Trans. of American Mathematical Society*, 347:(5), 1649-1667, 1995.

[NIY99] T. Nakata, H. Imahayashi and M. Yamashita Probabilistic local majority voting for the agreement problem on finite graphs, *Proc. 5th Computing and Combinatorics Conference*, 330-338, 1999.

[NIY00] T. Nakata, H. Imahayashi and M. Yamashita A probabilistic local polling game on weighted directed graphs with an application to the distributed agreement problem, *Neworks*, to appear.

[P96a] D. Pelc, Efficient fault location with small risk, *Proc. 3rd Colloq. on Structural Information & Communication Complexity*, 1996.

[P96b] D. Peleg, Local majority voting, small coalitions and controlling monopolies in graphs: a review, *Proc. 3rd Colloq. on Structural Information & Communication Complexity*, 1996.

[P98] D. Peleg, Size bounds for dynamic monopolies, *Discrete Applied Math*, vol 86, 263-273, 1998.

[PW95] D. Peleg, A. Wool, The availability of quorom systems, *Information and Computation*, vol 12, 210-223, 1995.

[PS83] Poljak, S., M. Sura, On Periodical Behavior in Society with Symmetric Influences, *Combinatorica*, vol 3, 119-121, 1983.

[PT86a] Poljak, S., D. Turzik, On Pre-periods of discrete influence systems, *Disc. Appl. Maths.*, 1986, 33-39.

[PT86b] Poljak, S., D. Turzik, On an application of convexity to discrete systems, *Disc. Appl. Maths.*, 27-32, 1986.

[SB94] M. Spasojevic, P. Berman, Geometric voting as the optimal pessimistic scheme for managing replicated data, *IEEE Trans. on Parallel & Distributed Systems*, vol 5, 64-73, 1994.

[S86] G. F. Sullivan, The complexity of system-level fault diagnosis and diagosability, *PhD thesis*, Department of Computer Science, Yale Univ., 1986.

[W96] A. Wool, Quorom systems for distributed control protocols, *PhD thesis*, The Weizmann Institute, Rehovot, Israel, 1996.

Recent BRICS Report Series Publications

RS-00-8 Nabil H. Mustafa and Aleksandar Pekeč. *Democratic Consensus and the Local Majority Rule*. May 2000. 38 pp.

RS-00-7 Lars Arge and Jakob Pagter. *I/O-Space Trade-Offs*. April 2000. To appear in *7th Scandinavian Workshop on Algorithm Theory*, SWAT '98 Proceedings, LNCS, 2000.

RS-00-6 Ivan B. Damgård and Jesper Buus Nielsen. *Improved Non-Committing Encryption Schemes based on a General Complexity Assumption*. March 2000. 24 pp.

RS-00-5 Ivan B. Damgård and Mads J. Jurik. *Efficient Protocols based on Probabilistic Encryption using Composite Degree Residue Classes*. March 2000. 19 pp.

RS-00-4 Rasmus Pagh. *A New Trade-off for Deterministic Dictionaries*. February 2000.

RS-00-3 Fredrik Larsson, Paul Pettersson, and Wang Yi. *On Memory-Block Traversal Problems in Model Checking Timed Systems*. January 2000. 15 pp. Appears in Graf and Schwartzbach, editors, *Tools and Algorithms for The Construction and Analysis of Systems: 6th International Conference*, TACAS '00 Proceedings, LNCS 1785, 2000, pages 127–141.

RS-00-2 Igor Walukiewicz. *Local Logics for Traces*. January 2000. 30 pp.

RS-00-1 Rune B. Lyngsø and Christian N. S. Pedersen. *Pseudoknots in RNA Secondary Structures*. January 2000. 15 pp. To appear in *Fourth Annual International Conference on Computational Molecular Biology*, RECOMB '00 Proceedings, 2000.

RS-99-57 Peter D. Mosses. *A Modular SOS for ML Concurrency Primitives*. December 1999. 22 pp.

RS-99-56 Peter D. Mosses. *A Modular SOS for Action Notation*. December 1999. 39 pp. Full version of paper appearing in Mosses and Watt, editors, *Second International Workshop on Action Semantics*, AS '99 Proceedings, BRICS Notes Series NS-99-3, 1999, pages 131–142.