Rational Fair Consensus in the $\mathcal{GOSSIP}$ Model

Andrea Clementi[1], Luciano Gualà[1], Guido Proietti[2], and Giacomo
Scornavacca[2]

[1]Università *Tor Vergata* di Roma,
`clementi/guala@mat.uniroma2.it`
[2] Università degli Studi dell'Aquila, `guido.proietti@univaq.it`,
`giacomo.scornavacca@graduate.univaq.it`

September 21, 2018

Abstract

The *rational fair consensus problem* can be informally defined as follows. Consider a network of n (selfish) *rational agents*, each of them initially supporting a *color* chosen from a finite set Σ. The goal is to design a protocol that leads the network to a stable monochromatic configuration (i.e. a consensus) such that the probability that the winning color is c is equal to the fraction of the agents that initially support c, for any $c \in \Sigma$. Furthermore, this fairness property must be guaranteed (with high probability) even in presence of any fixed *coalition* of rational agents that may deviate from the protocol in order to increase the winning probability of their supported colors. A protocol having this property, in presence of coalitions of size at most t, is said to be a *whp -t-strong equilibrium*.

We investigate, for the first time, the rational fair consensus problem in the $\mathcal{GOSSIP}$ communication model where, at every round, every agent can actively contact at most one neighbor via a *push/pull* operation. We provide a randomized $\mathcal{GOSSIP}$ protocol that, starting from any initial color configuration of the complete graph, achieves rational fair consensus within $O(\log n)$ rounds using messages of $O(\log^2 n)$ size, w.h.p. More in details, we prove that our protocol is a whp t-strong equilibrium for any $t = o(n/\log n)$ and, moreover, it tolerates worst-case permanent faults provided that the number of non-faulty agents is $\Omega(n)$. As far as we know, our protocol is the first solution which avoids any all-to-all communication, thus resulting in $o(n^2)$ message complexity.

1 Introduction

There is an increasing interest on algorithmic tasks performed by distributed systems that are formed by a finite set of selfish, *rational agents*. When the system does not provide for any central authority, the techniques for studying this kind of processes lie at the intersection of two scientific fields, *Distributed Computing* and *Algorithmic Game Theory*. Typically, there is a social task/decision

to be performed by the distributed system and, at the same time, every agent of the system has his own profit: The latter being a fixed function of the final configuration reached by the system. A feasible solution here consists in a protocol that not only computes the desired task but it even must result profitable for every rational agent: In other words, agents should not get any gain (according to their own profit functions) to deviate from the protocol's local rules. Protocols satisfying this property are said *Nash equilibria* or, even better, *t-strong (Nash) equilibria* when this robustness property is guaranteed even if agents can form a deviating *coalition* of size at most t [4]. Perhaps, this framework has been investigated for the first time in [13], where rational behaviour is analyzed in secret-sharing problems and multiparty computations. More recently, the impact of rational adversaries has been investigated on fundamental tasks in Distributed Computing such as *leader election*, *consensus*, and *wake-up* problems [2, 3, 12, 14].

Inspired by this new line of research, we here study the *rational fair consensus* problem [14] which can be informally defined as follows (see Section 2 for a formal definition). At the onset, every agent supports a *color* $c \in \Sigma$ where Σ is the color space. The goal of the system is to reach a stable monochromatic configuration where all agents support the same *winning* color and the probability that the winning color is c equals the fraction of agents initially supporting c, for any color $c \in \Sigma$.[1] Moreover, every rational agent u has his own profit function which is maximized whenever the winning color is the one he supported at the onset, while it is smaller when the winning color is any other color, and, finally, it is much smaller (and minimized) whenever the protocol fails to achieve consensus. More general classes of profit functions have been studied for rational fair consensus (e.g. [2]). The well-known *fair leader election problem* is the special case of the fair consensus problem where the color initially supported by each agent is his own ID.

In this setting, a *with high probability*[2] (for short, *w.h.p.*) *t*-strong equilibrium (see Def. 1) for the rational fair consensus problem is a protocol $\mathcal{P}$ that, given any initial color configuration $\vec{c}$ for the n agents, w.h.p. achieves fair consensus and, moreover, for any coalition C of size at most t and for any deviating strategy of C, there is at least one agent in C that, according to the deviating strategy, w.h.p. will not increase his chance to make his color win.

Several versions of consensus in presence of rational agents have been recently studied [2, 3, 12] but only few of them consider the fairness property. As far as we know, rational consensus has been studied only in the $\mathcal{LOCAL}$ communication model [8, 15] where, at every round, every agent can exchange messages with all his neighbors. In [2], Abraham et al. present a protocol for fair leader election that is an $(n-1)$-*resilient equilibrium* (*t*-resilient equilibrium is a stronger version than *t*-strong equilibrium, where no agent of the coalition will profit from a deviation [1]). However, their protocol is not robust against crash faults. A protocol achieving consensus is given in [12] in presence of a rational adversary that controls a proper subset of agents (this model is different from the ones studied in [2, 3, 14] and in this work). Their protocol does not guarantee fairness and assumes there are no crash faults. In [3], a protocol achieving $(n-1)$-resilient

---

[1]Notice that this *fairness* property is stronger than validity, the latter being required in the classic consensus problem - see Section 2.

[2]We will adopt here the standard notion in probabilistic algorithms: An event is said to hold with high probability if its probability is $1 - \frac{1}{n^{\Omega(1)}}$.

equilibrium[3] is provided for fair leader election, while, in presence of crash faults, the protocol is shown to be (only) a Nash equilibrium. This protocol does not work for the rational fair consensus problem and, in general, we emphasize that, even though any protocol for fair leader election can be easily transformed to one for fair consensus, if agent's rational behaviour is considered - i.e. if the "rational" versions of such two problems are considered -, then this reduction is no longer true [14]. Further results are presented in [7] for some versions of rational consensus in models which depart significantly from ours.

More recently, Halpern and Vilaca [14] studied the rational fair consensus problem in the $\mathcal{LOCAL}$ communication model assuming that agents have unique IDs. They study the problem in the complete graph and in presence of dynamic patterns of crash faults. It is first shown that if the adversary can adaptively choose the initial configuration and the dynamic fault-pattern (i.e. a worst-case, dynamic adversary), then no protocol can achieve a Nash equilibrium for rational fair consensus. Then, they consider a much weaker, natural adversary on the complete graph: The adversarial fault pattern is chosen randomly according to some distribution π. They prove that, if π satisfies some reasonable conditions, then it is possibile to design a protocol achieving a Nash equilibrium for rational fair consensus provided that the overall number of faulty agents is smaller than $n-1$. Their protocol is not "light-weight" [15] since it has to take care about random dynamic fault patterns and it requires $\Omega(n^2)$ messages. The authors also claim that its robustness against coalitions is quite hard to analyze and thus it is not included in the paper.

Our contribution. Recently, there has been strong interest in the design of algorithms for several versions of consensus problems in network models that severely restrict communication and computation [5, 6, 10]: This both for efficiency considerations and because such models capture aspects of the way consensus is reached in social networks, biological systems, and other domains of interest in network science. From the point of view of computation, the restrictive setting is to assume that each node only has polylogarithmic size of memory available, while, as for communication, this bound is also required to link bandwidth available in each round. Finally, the number of interactions a node can open in one round are severely constrained. These constraints are well-captured by the synchronous $\mathcal{GOSSIP}$ model [6, 9, 10, 8, 16, 17]: At every round, every node can actively push or pull a (short) message (say, of polylogarithmic size) with at most one of his neighbors. Notice that, in every round, a node can receive more than one message but the number of active links is always $O(n)$: A *per-round* communication pattern that can be considered definitely reasonable in several real network applications.

A major point is that all the previous protocols for rational (fair) consensus [2, 3, 14] heavily rely on broadcast operations made by every (non faulty) agent: In the complete graph, every agent directly communicates some piece of information (e.g. his own ID) to all the agents. It turns out that the number of exchanged messages is $\Omega(n^2)$.

Achieving Nash equilibria without the use of all-to-all operations is a major technical issue we want to investigate in this paper.

As in the work of Halpern and Vilaca [14], we consider a complete network of n agents, each of them having a unique ID which is an integer in the set

---

[3]Actually, in [3], the obtained property is improperly named as t-strong equilibrium.

$[n] := \{1, \ldots, n\}$. We consider the $\mathcal{GOSSIP}$ communication model and, concerning rational fair consensus, we assume each agent u initially knows his ID, his initial opinion $c_u \in \Sigma$ and the network size n. When two nodes communicate, despite their selfish behaviour, they cannot cheat each other about their ID's. This condition is more than reasonable in several network scenarios, it is assumed in the previous works (e.g. [14]) and it definitely does not make the problem easy. Taking in mind the strong negative result obtained by Halpern and Vilaca [14] about worst-case dynamic agent faults, we explore a weaker kind of adversary, the *worst-case permanent* one: At the very beginning, every agent can be either active or faulty and we assume this initial setting can be managed by a worst-case adversary. After this setting, then no further action of the adversary is allowed. We again remind that rational fair consensus in any model allowing only sparse communication patterns has never been studied so far, even in the fault-free case: This communication constraint essentially makes previous solutions of little use. Moreover, the presence of this static adversary introduces further issues to take care about: a rational active agent can pretend to be a faulty node in some rounds, and hence the protocol must be robust also against this kind of (potentially profitable) deviations.

We provide a $\mathcal{GOSSIP}$ protocol that, starting from any initial color configuration, achieves rational fair consensus in $O(\log n)$ rounds using local memory and messages of $O(\log^2 n)$ size, w.h.p., thus resulting in $O(n \log^3 n)$ overall communication complexity. We prove that our protocol is a whp-t-strong equilibrium for any $t = o(n/\log n)$ and, moreover, it tolerates worst-case permanent faults provided that the number of active agents is $\Omega(n)$. We remark that the known previous protocols [2, 3, 14], on the complete graph, use $\Omega(n^2)$ messages and local memory of size $\Omega(n)$. It is always possible to simulate a $\mathcal{LOCAL}$ protocol over the $\mathcal{GOSSIP}$ model thanks to the general technique introduced in [8]. However, this approach would yield exponentially larger message size and it is not clear whether the so-obtained simulation achieves any kind of equilibrium w.r.t. selfish behaviour.

To the best of our knowledge, our protocol is the first efficient solution for rational fair consensus on the $\mathcal{GOSSIP}$ model and, thus, it represents a first evidence of the fact that a short sequence of sparse communications patterns (each pattern formed by n push/pull operations) suffices to reach this kind of equilibria. We believe this result might open interesting directions in the design of more scalable solutions in real network applications where fair consensus in presence of selfish agents is a crucial issue [18].

2 Preliminares

We consider a complete graph $G([n], E)$ of n nodes, each of them having a unique *label* in $[n] = \{1, \ldots, n\}$, and we adopt the synchronous $\mathcal{GOSSIP}$ model: at every round, each node can make either a *pull* or a *push* operation with one of his neighbors. The choice of the neighbor can be made *uniformly at random* (for short, u.a.r.). At the onset, every node u knows n and how to communicate with every other node over a *secure channel*: during a communication over the edge $\{u, v\}$, the two nodes are aware about the label of his peer and the exchanged message is private (this is fully in the line of the related previous works [2, 3] and, moreover, it well reflects the real scenarios inspiring the $\mathcal{GOSSIP}$ model,

such as peer-to-peer and opportunistic networks).

The classic *consensus problem* in presence of *unknown, permanent node-faults* can be defined as follows. At round $t = 0$, every node is either in the *active* state or in the *faulty* state and let $\mathcal{A}$ be the subset of active nodes. The permanent faults are chosen by a worst case adversary that knows the protocol. A node, starting in the faulty state, will remain quiescent for all the process while each active node $u \in \mathcal{A}$ supports a color $c_u \in \Sigma$ (Σ being a shared set of colors). A protocol solves the *consensus task* if all the following conditions are met:

- *Termination*: Every active node gets into a final state within a finite number of rounds.

- *Agreement*: When all active nodes have reached a final state they will support the same color c. We say that c is the winning color.

- *Validity*: The winning color c must be a *valid* one, i.e., a color which was initially supported by at least one active node.

In the *fair-consensus* task [15, 14] the *validity* property is replaced by a stronger, probabilistic property.

- *Fairness*: The probability that a color $c \in \Sigma$ is the winning one is equal to the fraction of active nodes that initially support c.

We remark that, initially, every node only knows his label and his state (active or not) while he knows nothing about the other nodes. It is only assumed that the (unknown) set $\mathcal{A}$ has linear size, i.e. $|\mathcal{A}| = \Theta(n)$. A well-studied special case of the above task is the *fair leader election* where every node initially support his own ID as a color and, hence, every active node must have the same chance to be elected.

Non-cooperative setting. Besides permanent node faults, we consider nodes that act as *selfish (rational) agents* according to the standard definition in Game Theory. For this reason, we denote the problem as the *rational fair consensus*. Formally, each node (from now on, *agent*) $u \in [n]$ has a utility function $\text{util}_u(s)$ defined on every final state $s \in S$ of the protocol, where $S = \Sigma \cup \{\bot\}$ (the protocol can either converge to a color or, if agreement is not reached, fail). We focus on the natural scenario where the utility function of agent u is maximal when the winning color is c_u, it is much less when the protocol converges to another color, and, it is minimal (in fact, it is very bad) when the protocol fails. We assume the following (normalized) payoff scheme: For each agent u there is exactly one value $c_u \in \Sigma$ such that $\text{util}_u(c_u) = 1$; moreover, $\text{util}_u(s = \bot) = -\chi$, for an arbitrary fixed value $\chi \geq 0$, while $\text{util}_u(c') = 0$ otherwise.

The strategy of an agent u is that of choosing an *adaptive* local algorithm σ_u from a set of feasible rules satisfying the system constraints. The adaptive algorithm defines the actions of an agent at every round: These actions may depend on the set of messages received so far during the process. Each agent chooses such an algorithm in order to maximize his expected utility, where the expectation is defined over the random choices performed by the agents during the process. A protocol thus results in the vector of the n local (randomized) algorithms chosen by every agent (also called *strategy profile* in Game Theory).

Given a protocol $\mathcal{P}$ and an initial color configuration $\vec{c}$, we call $Q(\mathcal{P}, \vec{c})$ the set of all the possible executions of $\mathcal{P}$ starting from $\vec{c}$. Moreover, let $q(\mathcal{P}, \vec{c})$ be the random variable over $Q(\mathcal{P}, \vec{c})$ representing a random execution of $\mathcal{P}$. We define $f : Q(\mathcal{P}, \vec{c}) \to S$ as the function that returns the outcome of any execution. For brevity's sake, for any agent u, we define $r_u(\mathcal{P}, \vec{c}) = \text{util}_u(f(q(\mathcal{P}, \vec{c})))$.

We adopt the following notion of equilibrium, called *whp t-strong equilibrium* that it is a probabilistic relaxation of t-strong equilibrium (a similar relaxation is considered in [11] for deterministic *truthfulness*). Such an equilibrium is a protocol (strategy profile) such that, for any deviation of any fixed coalition of size at most t, there is an agent in the coalition that will not improve his expected utility, w.h.p. This is formalized by conditioning the expected utility of the agents to a large subset of "good" executions of the protocol. Formally, let us consider a protocol $\mathcal{P}$, a coalition $C \subseteq [n]$ and a (restricted) protocol $\mathcal{P}'$ for C. By $(\mathcal{P}_{-C}, \mathcal{P}'_C)$ we denote the protocol where the agents in $\mathcal{A} \setminus C$ follow $\mathcal{P}$ while the active agents in C follow $\mathcal{P}'$. Given a color configuration $\vec{c}$, protocols $\mathcal{P}$ and $\mathcal{P}'$, we let $\Omega = \Omega(\mathcal{P}, \vec{c})$ and $\Omega' = \Omega((\mathcal{P}_{-C}, \mathcal{P}'_C), \vec{c})$ be the probability spaces yielded by running $\mathcal{P}$ and $(\mathcal{P}_{-C}, \mathcal{P}'_C)$ from $\vec{c}$, respectively.

Definition 1. *We say a protocol $\mathcal{P}$ is a* whp-t-strong equilibrium *if, for any initial color configuration $\vec{c}$, for every coalition C of at most t agents, and for every restricted protocol $\mathcal{P}'$ for C, the following properties hold:*

- *There is a subset $\mathcal{G} \subseteq \Omega$ of executions such that $\Pr_\Omega(\mathcal{G}) \geq 1 - \frac{1}{n^{\Theta(1)}}$;*

- *There is a subset $\mathcal{G}' \subseteq \Omega'$ of executions such that $\Pr_{\Omega'}(\mathcal{G}') \geq 1 - \frac{1}{n^{\Theta(1)}}$;*

- *There is an agent $w \in C$ such that*

$$
\begin{aligned}
&\mathbf{E}_\Omega[r_w(\mathcal{P}, \vec{c}) | q((\mathcal{P}, \vec{c}) \in \mathcal{G}] \geq \\
&\mathbf{E}_{\Omega'}[r_w((\mathcal{P}_{-C}, \mathcal{P}'_C), \vec{c}) | q((\mathcal{P}_{-C}, \mathcal{P}'_C), \vec{c}) \in \mathcal{G}'].
\end{aligned}
\tag{1}
$$

3 An Efficient Protocol for Rational Fair Consensus

Informal description of the protocol. In order to reach *fair consensus*, our protocol adopts a simple and natural idea (see, for example [2]): Choose u.a.r. an active agent of the network and then lead the system to stabilize on the color supported by this agent. It is easy to show that if all the agents follow the protocol then a fair consensus is achieved. However, the presence of a coalition of rational agents requires further protocol actions in order to prevent convergence towards unfair consensus: This is obtained using some verification procedures that work in logarithimic time and use messages of size $O(\log^2 n)$.

The protocol is parametrized in the maximum number αn of faulty agents (where $0 \leq \alpha < 1$ is the so-called fault-tolerance parameter of the protocol) and it assumes every agent knows the size n of the system. Its local rules are organized in the following consecutive phases. A detailed description of the protocol is given in Algorithm 1.

---

[4]if agent v does not reply (or replies in a unexpected way), then he is marked as faulty ($\forall j \in [q], h_{v,j} = 0$).

[5]For each z_v appearing in $W_{\min} \cap L_u$ check if the vote is the same.

Algorithm 1 Protocol $\mathcal{P}$

The local rules for Protocol $\mathcal{P}$

Local Data: Each agent $u \in [n]$ knows label u, his supported color $c_u \in \Sigma$, the agent number n, and the fault-tolerance parameter γ;

Initialize(): u computes the parameters $m = n^3$ and the number of rounds $q = \gamma \log n$;

Voting-Intention(): Choose a list of votes H_u
$H_u := \{(h_{u,1}, z_{u,1}) \ldots (h_{u,q}, z_{u,q})\}$ where
$\forall i$ in $[q]$ $h_{u,i}$ is chosen u.a.r. in $[m]$ and $z_{u,i}$ is chosen u.a.r. in $[n]$

Commitment(): Compute a list L_u of collected vote intentions
$L_u := \emptyset$

for q rounds **do**
   Pull from an agent v chosen u.a.r. his list H_v[4]
   $\forall j \in [q]$ update $L_u := L_u \cup \{(v, h_{v,j}, z_{v,j})\}$;
   Receiving a pull requests: send your own H_u list
end for

Voting(H_u): Push your votes according to H_u and collect the received votes in W_u
$W_u := \emptyset$

for $i = 1, \ldots, q$ rounds **do**
   Push $h_{u,i}$ to agent $z_{u,i}$
   Receiving votes: let $\{h_1, \ldots, h_\ell\}$ be the votes received (in round i) from agents $\{z_1, \ldots, z_\ell\}$, respectively and update $W_u := W_u \cup \{(h_1, z_1) \ldots (h_\ell, z_\ell)\}$
end for

Compute the value $k_u := \sum_{h \in W_u} h \mod m$

Find-Min(CE$_u = (k_u, W_u, c_u, u)$):
$\text{CE}_u^{min} := \text{CE}_u$

for q rounds **do**
   Pull from an agent v u.a.r. in $[n]$ his Certificate CE_v^{min}
   if $k_v^{min} < k_u^{min}$ **then**
     $\text{CE}_u^{min} := \text{CE}_v^{min}$
   end if
   Receiving a pull request: send CE_u^{min}
end for

- **Coherence (CE_u^{min}):**

for q rounds **do**
   Push to an agent v u.a.r. in $[n]$ the Certificate CE_u^{min}
   Receiving a set of *Certificates* CE:
   if $\exists \text{CE}_v^{min} \in \text{CE} : \text{CE}_u^{min} \neq \text{CE}_v^{min}$ **then**
     Make the protocol fail;
   end if
end for
$\text{CE}^{min} := \text{CE}_u^{min}$

Verification(L_u):
$\text{CE}^{min} := (k_{min}, W_{min}, c_{min}, z_{min})$:
if $k_{min} = (\sum_{h \in W_{min}} h \mod m)$ and W_{min} is consistent[5] with the list of votes in L_u **then**
   Support the color c_{min}
else if **then**
   Make the protocol fail;
end if

- In the VOTING-INTENTION phase each agent u randomly chooses a "small" (i.e. a logarithmic) number of agents and, for each of them, he decides one random *vote* (chosen u.a.r. in the range $[n^3]$): The resulting list is called the *vote intention* H_u of agent u.
- In the COMMITMENT phase, each agent u asks (using pull operations) a small number of agents to send him their *vote intentions*: All such data will be stored in a set we call L_u. If an agent v does not answer to one of u's requests, then v is marked as *faulty* by u and, from now on, u will consider all the votes of H_v equal to zero.
- In the VOTING phase, each agent u votes (via the push mechanism) according to H_u and, thus, in turn, u also gets the set W_u of the received votes from the other agents. Now each agent u can compute the value k_u equal to the sum of all the received votes modulo $m = n^3$ and creates his *Certificate* CE_u. The certificate contains the value k_u, the received votes W_u, his color c_u and his label u. The choice of this value for m ensures that all k_u's are different, w.h.p. and, so, the minimum is unique (this fact will be exploited in the next phase).
- All the agents start the FIND-MIN phase that makes every active agent converge on the "minimal" certificate $\mathrm{CE}_z = (k_z, W_z, c_z, z)$ such that $k_z = \min_{v \in \mathcal{A}} k_v$. The agents perform this task using pull operations as in the standard $\mathcal{GOSSIP}$ broadcast protocol [19], taking $O(\log n)$ rounds. More precisely, at every round, every agent u stores the current "minimal" certificate, i.e., that with the minimum value of k_z he has seen so far and u asks (via a pull operation) to a random neighbor v his current minimal certificate. We call $\mathrm{CE}_u^{\min}$ the certificate owned by u at the end of the FIND-MIN phase.
- The COHERENCE phase is performed in order to ensure that all the agents posses the same certificate, namely the one resulting from the FIND-MIN phase. In particular, agent u sends his $\mathrm{CE}_u^{\min}$ to a logarithmic number of randomly chosen agents and he makes the protocol fail[6] if he receives a different certificate $\mathrm{CE}_v^{\min}$ from an agent v.
- At the end of the VERIFICATION phase, every agent u agrees on the color c_z if the votes in W_z are compatible with the votes in L_u. The votes are not compatible if there is a vote in W_z, say a vote given to z by w, which is different from the vote to z by w stored in L_u (hence, u pulled w in the COMMITMENT phase).

In Subsection 3.1, we show that the proposed protocol w.h.p. achieves fair consensus in presence of at most αn faulty agents, while, in Subsection 3.2, we prove that our protocol is a whp t-strong equilibrium for any $t = o(n/\log n)$.

3.1   Analysis of the protocol in the cooperative setting

In this section we analyse Protocol $\mathcal{P}$ when all the active agents follow $\mathcal{P}$. We first give the concept of a *good* execution of $\mathcal{P}$. In a good execution, every active agent receives $\Theta(\log n)$ votes, all the k_u values are all distinct (so, $k_{\min}$ is unique), and after the FIND-MIN phase, every active agent agrees on the same Certificate of minimal value. Formally, we introduce the following definition:

Definition 2. *Let $q(\mathcal{P}, \vec{c}) \in Q(\mathcal{P}, \vec{c})$ be a random execution of the protocol. We say that $q(\mathcal{P}, \vec{c})$ is* good *(and define $\mathcal{G} \subset \Omega$ as the set of all good executions) if all the following events hold:*

---

[6]For instance, the agent can enter in an invalid state by supporting a color not in Σ.

1. *Every agent in $\mathcal{A}$ receives $\Theta(\log n)$ votes.*

2. *The k_u values are all distinct (so, $k_{\min}$ is unique).*

3. *Let CE^{min} be the certificate of the agent getting the minimal value $k_{\min}$. Then, after the FIND-MIN phase, for every active agent u, we have $CE_u^{min} = CE^{min}$.*

Lemma 3 below shows that, if number of non-faulty agents is $\Theta(n)$, a random execution of $\mathcal{P}$ is good w.h.p.

Lemma 3. *Let α be an absolute constant such that $0 \leq \alpha < 1$. If the number of faulty agents is at most αn, then the random execution of $\mathcal{P}$ (with a suitable choice of parameter $\gamma = \gamma(\alpha)$) is good, w.h.p., i.e. $\Pr_\Omega(\mathcal{G}) \geq 1 - \frac{1}{n^{\Theta(1)}}$.*

Sketch of Proof. We assume that there are at most αn faulty nodes and that all the active agents follow the algorithm for $\gamma \log n$ rounds (for a suitable constant $\gamma(\alpha)$). As for Point 1, for every agent v, consider the random variable X_v that counts the number of votes agent v will get after the VOTING phase. In this phase, at each of the $\gamma \log n$ rounds, every active node chooses independently and u.a.r. one agent to vote. So, X_v can be written as the sum of $\Theta(n \log n)$ mutually independent Bernoulli random variables. Using Chernoff's bound (see Lemma 8) on every random variable X_v and the Union Bound, we have that (for a suitable choice of parameter $\gamma = \gamma(\alpha)$) two positive constants β_1, β_2 exist such that

$$\beta_1 \log n \leq X_u \leq \beta_2 \log n \,, \ \forall u \in \mathcal{A}, \text{ w.h.p.}$$

As for Point 2, since the k_u values are independently chosen u.a.r. in $[m] = [n^3]$, using standard argument, there will be no collisions and, thus, the minimum of these values is unique, w.h.p.

As for Point 3, observe that the FIND-MIN phase is equivalent to a standard single-source broadcast operation of the message CE^{min} on the complete subgraph induced by the subset $\mathcal{A}$ of active agents. The convergence time of this basic task on the complete graph for the $\mathcal{GOSSIP}$ model - when agents use the pull mechanism - is known to be $\Theta(\log n)$ (w.h.p) [19]. The only difference here is the presence of faulty agents. However, by a suitable choice of the constant $\gamma = \gamma(\alpha)$, we can easily adapt the analysis in [19] for the complete subgraph induced by any subset $\mathcal{A}$ of active agents provided that $|\mathcal{A}| \geq (1 - \alpha)n$ (essentially, the presence of αn faulty agents is balanced by a slightly longer broadcast phase). □

The three properties guaranteed by a good execution are the key ingredients in the proof of the next theorem stating that Protocol $\mathcal{P}$ achieves a fair-consensus.

Theorem 4. *Let α be an absolute constant such that $0 \leq \alpha < 1$. If the number of faulty agents is at most αn, Protocol $\mathcal{P}$ (with a suitable choice of parameter $\gamma = \gamma(\alpha)$) computes a fair consensus within $O(\log n)$ rounds and using messages of size $O(\log^2 n)$, w.h.p.*

Sketch of Proof. Conditioning to $q(\mathcal{P}, \vec{c}) \in \mathcal{G}$ (an event that holds w.h.p. because of Lemma 3), we can assume that the protocol does not fail. Indeed, from Definition 2 (property 1), for every active agent u the value k_u is defined, from

Definition 2 (property 2) the value k_{min} is unique and, from Definition 2 (property 3), after the FIND-MIN phase all active agents converge to a unique CE^{min}. So, there are no multiple minimal certificates that can make fail the COHERENCE phase and in the VERIFICATION phase the Certificate is valid (each agent votes as declared in the COMMITMENT phase). By simple probabilistic arguments, the computation of $k_u = \sum_{h \in W_u} h \mod m$ performed by every agent u implies that every agent has the same chance to get the (unique) minimal value. So, the protocol computes a fair leader election and the network converges to the leader color in the VERIFICATION phase. The protocol terminates in $O(\log n)$ rounds (by construction) and the largest message is the Certificate of the most voted agent that have size $O(\log^2 n)$. Indeed, thanks to Definition 2.1, it gets $O(\log n)$ votes, each of them having size $O(\log m) = O(\log n)$. $\square$

3.2 Analysis of the protocol in the presence of rational agents

In this section we analyse Protocol $\mathcal{P}$ in the presence of rational agents and show that $\mathcal{P}$ is a w.h.p. t-strong equilibrium, for any $t = o\left(\frac{n}{\log n}\right)$. We recall that $\mathcal{A}$ is the set of active agents, C the set of agents that deviate to a new set of local algorithms $\mathcal{P}'_C$ while $\mathcal{A} \setminus C$ is the subset of active agents that follow Algorithm 31. W.l.o.g. we assume $C \subseteq \mathcal{A}$. Moreover, we say that an agent u is in the *vote intention* of an agent $v \in \mathcal{A} \setminus C$ if it holds $(*, u) \in H_v$ (thus in the VOTING phase u will receive a vote of v).

Following the same approach of Section 3.1, we first revise the notion of good execution in order to deal with the selfish behaviour of rational agents.

Definition 5. *Let C be the coalition, $\mathcal{P}'_C$ be the new set of local algorithms for C, and $q((\mathcal{P}_{-C}, \mathcal{P}'_C), \vec{c})$ be the random variable over $Q((\mathcal{P}_{-C}, \mathcal{P}'_C), \vec{c})$. We say that $q((\mathcal{P}_{-C}, \mathcal{P}'_C), \vec{c})$ is good (and define $\mathcal{G}' \subset \Omega'$ as the subset of all good executions) if the following events hold:*

1. *In the COMMITMENT phase each agent $u \in \mathcal{A}$ receives at least one pull request by an agent $v \in \mathcal{A} \setminus C$ asking for (a copy of) H_u.*

2. *At the end of the COHERENCE phase either Protocol $(\mathcal{P}_{-C}, \mathcal{P}'_C)$ fails or every agent $v \in \mathcal{A} \setminus C$ gets the same certificate CE^{min}.*

3. *At the end of the COMMITMENT phase, let $M \subset \mathcal{A}$ be the set of agents that have received at least a pull request by an agent in C. Then for every agent $u \in \mathcal{A}$, there exists an agent $v \in \mathcal{A} \setminus (C \cup M)$ such that u is in the vote intention of v (i.e. u receives the vote from v in the VOTING phase).*

Under some reasonable assumptions of the number of faulty agents and the size of the coalition, the next lemma shows that the random execution of protocol is good w.h.p., even when a coalition deviates from $\mathcal{P}$.

Lemma 6. *Let α be an absolute constant such that $0 \le \alpha < 1$. For any set of faulty agents of size at most αn and for any coalition C of size $o\left(\frac{n}{\log n}\right)$, the random execution $q((\mathcal{P}_{-C}, \mathcal{P}'_C), \vec{c})$ (with a suitable choice of the parameter $\gamma = \gamma(\alpha)$) is good, w.h.p., i.e. $\Pr_{\Omega'}(\mathcal{G}') \ge 1 - \frac{1}{n^{\Theta(1)}}$.*

Proof. The theorem hypothesis imply that $|\mathcal{A} \setminus C| = \alpha' n$ for some constant $0 < \alpha' < 1 - \alpha$ and we recall that each active agent in $\mathcal{A} \setminus C$ runs each phase of Algorithm 31 for $\gamma \log n$ rounds.

1) For any agent $u \in \mathcal{A}$ and for any agent $v \in \mathcal{A} \setminus C$, define the binary random variable $X_{u,v} = 1$ iff in the COMMITMENT phase agent $u \in \mathcal{A}$ receives a pull request by v asking for (a copy of) H_u. Since v follows the protocol, it easily holds that

$$\Pr(X_{u,v} = 0) = \left(1 - \frac{1}{n}\right)^{\gamma \log n}$$

Since, all agents $v \in \mathcal{A} \setminus C$ follow the protocol, thus making mutually independent u.a.r. pull requests, we get

$$\Pr(\forall v \in \mathcal{A} \setminus C : X_{u,v} = 0) = \left(1 - \frac{1}{n}\right)^{\alpha' n \cdot \gamma \log n}$$

$$\leq e^{-\alpha' \gamma \log n} = \frac{1}{n^{\alpha' \gamma}}$$

Finally, choosing a sufficiently large γ and applying the Union Bound, we get

$$\Pr(\exists u, \forall v \in \mathcal{A} \setminus C : X_{u,v} = 0) = \frac{1}{n^{\Theta(1)}}.$$

2) Assume that at the beginning of the COHERENCE phase there are at least two distinct Certificates, and let CE$'$ be one of them. We thus consider the subset X of $\mathcal{A} \setminus C$ formed by all the agents having CE$'$. Without loss of generality assume $|\mathcal{A} \setminus (C \cup X)| \geq |X|$ thus $|\mathcal{A} \setminus (C \cup X)| \geq \frac{\alpha'}{2} n$. Then, using similar arguments to those in the proof of Point 1, we can fix $\gamma(\alpha)$ such that (after $\gamma \log n$ rounds) there is (at least) one agent in X that, following the protocol, will send his Certificate to an agent in $\mathcal{A} \setminus (C \cup X)$ w.h.p. Then the protocol fails.

3) Since $|C| = o(\frac{n}{\log n})$ and the length of the COMMITMENT phase is $O(\log n)$ rounds, the overall number of pull requests during this phase made by C is $o(n)$. Thus we can assume that $|\mathcal{A} \setminus (C \cup M)| \geq \lambda n$ for a fixed constant λ, with $0 \leq \lambda < \alpha'$. Moreover, the overall number of votes sent by $\mathcal{A} \setminus (C \cup M)$ (towards an agent chosen independently u.a.r. in $[n]$) in the VOTING phase is greater than $\lambda \gamma n \log n$. Hence, since all the agents in $\mathcal{A} \setminus (C \cup M)$ follow the protocol, using similar arguments to those in the proof of Point 1, we can fix $\gamma(\alpha)$ in order to ensure that every agent in $\mathcal{A}$ receives at least one vote from an agent in $\mathcal{A} \setminus (C \cup M)$, w.h.p. □

Lemma 6 ensures that, at the end of a good execution, the following facts hold w.h.p.: 1) The vote intention of any agent u in the coalition C can be verified by at least one agent v which does not belong to C; 2) If the protocol does not fail, then all the agents which does not belong to C agree on the same certificate CE^{min} and they check the same set of votes $W_{\min}$; 3) All the agents receive at least one vote from an agent which does not belong to C. This guarantees that for every agent u the value k_u cannot be controlled by the coalition C, and thus, according to the protocol rules, k_u is chosen u.a.r. in the range $[1, m]$; The three facts above will be used to prove that the protocol $\mathcal{P}$ is a whp t-strong equilibrium.

Theorem 7. *Let α be an absolute constant such that $0 \leq \alpha < 1$. For any set of faulty agents of size at most αn, protocol $\mathcal{P}$ (with a suitable choice of the parameter $\gamma = \gamma(\alpha)$) is a whp t-strong equilibrium for any $t = o\left(\frac{n}{\log n}\right)$.*

Proof. Let us consider an arbitrary coalition C of at most t agents and fix the set of local algorithms $\mathcal{P}'_C$ for them. We need the following preliminary definitions:

- $h^*_{v,u}$ is the vote declared by agent v for the agent u in the first declaration of v to some agent $z \in \mathcal{A} \setminus C$ during the COMMITMENT phase (we recall that if v has not correctly replied to z or $(*, u) \notin H_v$ then $h^*_{v,u} = 0$). We also define $k^*_u = \sum_{v \in \mathcal{A}} h^*_{v,u}$. Notice that k^*_u may be different from k_u, the latter being the value that u should declare (in the Certificate CE_u) during the FIND-MIN phase and also observe that k_u is a value agent u can lie on. The difference between k^*_u and k_u leads us to introduce the following two concepts of winner.

- We call *Winner* the agent whose label is contained in the (unique - whenever the execution is good) certificate $\mathrm{CE}^{\min}$ after the COHERENCE phase. Notice that the certificate $\mathrm{CE}^{\min}$ contains the value k_{min} which is the minimum value among the declared values k_u.

- Let $a = \arg\min_{v \in \mathcal{A} \setminus C} k_v$, and let $b = \arg\min_{v \in C} k^*_v$. We say that the *Legitimate Winner* is a if $k_a < k^*_b$, and it is b otherwise. Notice that the definition of *Legitimate Winner* does not depend on the values k_v declared by agents in C and thus it may be case that *Winner* and *Legitimate Winner* are not the same. In particular we are interested in the following distinct two events.

- For any $u \in \mathcal{A}$, let E_u be the event "the *Legitimate Winner* is u" and define $E_C = \cup_{u \in C} E_u$. Furthermore E'_C is the event "the *Winner* is an agent in C".

We now prove that, in a non-failing good execution of $(\mathcal{P}_{-C}, \mathcal{P}'_C)$, if the *Legitimate Winner* is not in C, then the *Winner* is not in C as well.

Claim 1. *Let us consider any good execution which does not fail. Conditioning to the event $\bar{E}_C$, the Winner turns out to be the Legitimate Winner (hence $\bar{E}_C$ implies $\bar{E}'_C$).*

Proof. (of Claim 1) The proof argument is by contradiction. Assume that the *Legitimate Winner* v belongs to $\mathcal{A} \setminus C$ and the protocol does not converge to c_v. Then this happens only if agent v accepts a certificate $\mathrm{CE}_u = (k_u, W_u, c_u, u)$ different from his own Certificate $\mathrm{CE}_v = (k_v, W_v, c_v, v)$ and such that $k_u < k_v$. Notice that, by definition of *Legitimate Winner*, u must belong to C and $k_u \neq k^*_u$. Hence, thanks to Definition 5 (property 2), at the end of the COHERENCE phase every agent $v \in \mathcal{A} \setminus C$ gets the same certificate CE_u (hence, the same set W_u) and thanks to Definition 5 (property 1), some agent $z \in \mathcal{A} \setminus C$ exists whose local data (i.e. L_z) is not consistent w.r.t. W_u thus making the protocol fail. A contradiction. $\square$

Claim 2. *Let us consider any good execution, every agent in $\mathcal{A}$ has the same chance to be the Legitimate Winner (i.e. $\forall u \in \mathcal{A}$, $\mathbf{Pr}(E_u) = \frac{1}{|\mathcal{A}|}$).*

Proof. (of Claim 2) We will argue that (i) for any $b \in C$ we have that k^*_b is u.a.r. in $[m]$ and (ii) for any $a \in \mathcal{A} \setminus C$ we have that k_a is u.a.r. in $[m]$. To prove

12

(i), notice that k_b^* is defined in the COMMITMENT phase: Thanks to Definition 5 (property 3) at the end of this phase, for every $b \in C$ there is still at least one agent z in $\mathcal{A} \setminus C$ that voted b and was not pulled by any agent of C. Since $z \in \mathcal{A} \setminus C$, $h_{z,b}^*$ coincides to the vote of z actually given (in the VOTING phase) to b, which is distributed u.a.r. in $[m]$. For the principle of deferred decision this implies that k_b^* is u.a.r. in $[m]$ as well. To prove (ii), notice that k_a is determined in the VOTING phase: Thanks to Definition 5 (property 3) at the end of the VOTING phase, for every $a \in \mathcal{A} \setminus C$ there is still at least one agent z in $\mathcal{A} \setminus C$ that voted for a and was not pulled by any agent of C. Since $z \in \mathcal{A} \setminus C$, $h_{z,a}^*$ coincides to the vote of z actually given (in the VOTING phase) to a which is distributed u.a.r. in $[m]$. For the principle of deferred decision this implies that k_a is u.a.r. in $[m]$ as well. Observe that, since both z and a are in $\mathcal{A} \setminus C$, $h_{z,a}^*$ cannot be discovered by any agent in C during the VOTING phase.
Claim 2 follows from (i), (ii), and from the fact that, for simple symmetry argument, any agent has the same chance to get the minimal value. $\qquad\square$

Given any subset $X \subseteq \mathcal{A}$ and any color $c \in \Sigma$, we define $N(X, c)$ as the number of agents in X supporting c. Then Claims 1 and 2, easily imply the following properties of a good execution.

Claim 3. *Let us consider any good execution which does not fail. Conditioning to the event $\bar{E}_C$, the protocol converges to a color $c \in \Sigma$ with probability $\frac{N(\mathcal{A} \setminus C, c)}{|\mathcal{A} \setminus C|}$.*

Claim 4. *Let us consider any good execution which does not fail. Then it holds that $\Pr(E_C') \leq \frac{|C|}{|\mathcal{A}|}$.*

Thanks to Lemma 6 we can now consider only good executions (i.e. $q((\mathcal{P}_{-C}, \mathcal{P}_C'), \vec{c}) \in \mathcal{G}'$) where Claim 3 and Claim 4 do hold. We now use such claims to show by contradiction that the protocol is a whp t-strong equilibrium.
For any $c \in \Sigma$ define $\Pr(c)$ as the probability the protocol converges to color c. Then for every agent $u \in C$, we can evaluate his expected utility[7]:

$$
\begin{aligned}
&\mathbf{E}_{\Omega'}[r_u((\mathcal{P}_{-C}, \mathcal{P}_C'), \vec{c})] = \\
&\Pr(f(q((\mathcal{P}_{-C}, \mathcal{P}_C'), \vec{c}) = c_u) - \chi \Pr(f(q((\mathcal{P}_{-C}, \mathcal{P}_C'), \vec{c}) = \perp) \leq \\
&\Pr(f(q((\mathcal{P}_{-C}, \mathcal{P}_C'), \vec{c}) = c_u) = \\
&\Pr(E_C') \Pr(c_u | E_C') + \Pr(\bar{E}_C') \Pr(c_u | \bar{E}_C')
\end{aligned}
$$

Hence,

$$
\begin{aligned}
&\mathbf{E}_{\Omega'}[r_u((\mathcal{P}_{-C}, \mathcal{P}_C'), \vec{c})] \leq \\
&\Pr(E_C') \Pr(c_u | E_C') + \Pr(\bar{E}_C') \Pr(c_u | \bar{E}_C')
\end{aligned} \tag{2}
$$

Thanks to Claim 3 and 4, we can fix some $\delta \in [0, 1]$ such that the above formula can be rewritten as:

$$
\left(\frac{|C|}{|\mathcal{A}|} - \delta \right) \Pr(c_u | E_C') + \left(\frac{|\mathcal{A} \setminus C|}{|\mathcal{A}|} + \delta \right) \frac{N(\mathcal{A} \setminus C, c_u)}{|\mathcal{A} \setminus C|}
$$

---

[7]Recall that all the events in Ω' and Ω are conditioned to $q((\mathcal{P}_{-C}, \mathcal{P}_C'), \vec{c}) \in \mathcal{G}'$ and $q(\mathcal{P}, \vec{c}) \in \mathcal{G}$, respectively. For the sake of clarity, with a little abuse of notation, we will not explicitly write it.

Note that, thanks to Theorem 4, if all the agents follow Protocol $\mathcal{P}$ the expected utility of every agent u is $N(\mathcal{A}, c_u)/|\mathcal{A}|$. According to Definition 1, in order to have a profitable deviation for C, for every agent $u \in C$, we should have:

$$\mathbf{E}_{\Omega'}[r_u((\mathcal{P}_{-C}, \mathcal{P}'_C), \vec{c})] > \mathbf{E}_{\Omega}[r_u(\mathcal{P}, \vec{c})]$$

Thanks to Inequality 2, the above condition implies that

$$\left(\frac{|C|}{|\mathcal{A}|} - \delta\right) \Pr(c_u|E'_C) + \left(\frac{|\mathcal{A} \setminus C|}{|\mathcal{A}|} + \delta\right) \frac{N(\mathcal{A} \setminus C, c_u)}{|\mathcal{A} \setminus C|}$$
$$> \frac{N(\mathcal{A}, c_u)}{|\mathcal{A}|}$$

Hence,

$$\Pr(c_u|E'_C) >$$
$$\frac{1}{|C|/|\mathcal{A}| - \delta} \cdot \left(\frac{N(\mathcal{A}, c_u)}{|\mathcal{A}|} - \left(\frac{|\mathcal{A} \setminus C|}{|\mathcal{A}|} + \delta\right) \cdot \frac{N(\mathcal{A} \setminus C, c_u)}{|\mathcal{A} \setminus C|}\right)$$

It thus follows that

$$\Pr(c_u|E'_C) > \frac{N(C, c_u)/|\mathcal{A}| - \delta N(\mathcal{A} \setminus C, c_u)/|\mathcal{A} \setminus C|}{|C|/|\mathcal{A}| - \delta}, \tag{3}$$

where in the last inequality we used the fact that, by definition,

$$N(C, c_u) = N(\mathcal{A}, c_u) - N(\mathcal{A} \setminus C, c_u)$$

Let $\Sigma(C)$ be set of all the colors that are supported by at least one agent in C. So, for any $c \in \Sigma(C)$, Inequality 3 should hold. Then, saturating the above inequalities over all colors in $\Sigma(C)$, we get:

$$\sum_{c \in \Sigma(C)} \Pr(c|E'_C) >$$
$$\sum_{c \in \Sigma(C)} \frac{N(C, c)/|\mathcal{A}| - \delta N(\mathcal{A} \setminus C, c)/|\mathcal{A} \setminus C|}{|C|/|\mathcal{A}| - \delta} \tag{4}$$

Since

$$\sum_{c \in \Sigma(C)} N(C, c) = |C|,$$

then the r.h.s. of Inequality 4 can be rewritten as

$$\sum_{c \in \Sigma(C)} \frac{N(C, c)/|\mathcal{A}| - \delta N(\mathcal{A} \setminus C, c)/|\mathcal{A} \setminus C|}{|C|/|\mathcal{A}| - \delta} =$$
$$\frac{|C|/|\mathcal{A}| - \delta \sum_{c \in \Sigma(C)} N(\mathcal{A} \setminus C, c)/|\mathcal{A} \setminus C|}{|C|/|\mathcal{A}| - \delta} \tag{5}$$

Since, by definition, it holds that

$$\sum_{c \in \Sigma(C)} N(\mathcal{A} \setminus C, c) \le |\mathcal{A} \setminus C|$$

14

then we get

$$\frac{|C|/|\mathcal{A}| - \delta \sum_{c \in \Sigma(C)} N(\mathcal{A} \setminus C, c)/|\mathcal{A} \setminus C|}{|C|/|\mathcal{A}| - \delta} \geq 1 \qquad (6)$$

From Inequalities 4-6, we should have $\sum_{c \in \Sigma(C)} \Pr(c|E'_C) > 1$: This is clearly false. Thus, there must be at least one agent in C that will not increase his expected utility, concluding the proof. $\qquad\square$

3.3 Useful probability bounds

Lemma 8 (Chernoff bounds). *Let $X = \sum_{i=1}^{n} X_i$ where X_i's are independent Bernoulli random variables and let $\mu = \mathbf{E}[X]$. Then,*

1. *For any $0 < \delta \leqslant 4$, $\Pr(X > (1 + \delta)\mu) < e^{-\frac{\delta^2 \mu}{4}}$;*

2. *For any $\delta \geqslant 4$, $\Pr(X > (1 + \delta)\mu) < e^{-\delta\mu}$;*

3. *For any $\lambda > 0$, $\Pr(X \geqslant \mu + \lambda) \leqslant e^{-2\lambda^2/n}$.*

4 Conclusions

Efficient algorithmic methods for consensus tasks in fully-decentralized systems where agents may reveal a selfish behaviour is a central issue that arises in several scientific fields such as social networks, peer-to-peer networks, biological systems, e-commerce, and crypto-currency. Hence, the definition of reasonable distributed models and specific problems capturing some of the major technical questions is a line of research that is currently attracting increasing interest from the distributed computing community. One of the technical goals in this context is that of reducing *local memory and communication cost* of the proposed consensus protocols. Considering the specific network scenarios where this kind of rational consensus may play an important role, we believe this is an important question which is still far to be well-understood. Our contribution provides a first step for this general aim since it shows that, on complete networks, fair rational consensus can be obtained in logarithmic time in a communication model, the $\mathcal{GOSSIP}$ one, that severely restricts both local memory and message communication.

In our opinion, two specific open problems "suggested" by our work look rather interesting. The first one is to provide $\mathcal{GOSSIP}$ algorithms for rational fair consensus in other relevant classes of graphs, while the second one is the study of this problem in the asynchronous (i.e. sequential) $\mathcal{GOSSIP}$ model where, at every round, only one (possibly random) agent is awake.

References

[1] I. Abraham, D. Dolev, R. Gonen, and J.Y. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In Eric Ruppert and Dahlia Malkhi, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 53–62. ACM, 2006.

[2] I. Abraham, D. Dolev, and J.Y. Halpern. Distributed protocols for leader election: A game-theoretic perspective. In Yehuda Afek, editor, *Distributed Computing - 27th International Symposium, DISC 2013, Jerusalem, Israel, October 14-18, 2013. Proceedings*, volume 8205 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2013.

[3] Y. Afek, Y. Ginzberg, S.L. Feibish, and M. Sulamy. Distributed computing building blocks for rational agents. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 406–415. ACM, 2014.

[4] Nir Andelman, Michal Feldman, and Yishay Mansour. Strong price of anarchy. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 189–198, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[5] Dana Angluin, James Aspnes, and David Eisenstat. A Simple Population Protocol for Fast Robust Approximate Majority. *Distributed Computing*, 21(2):87–102, 2008. (Preliminary version in DISC'07).

[6] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and R. Silvestri. Plurality Consensus in the Gossip Model. In *Proc. of the 26th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'15)*, pages 371–390. SIAM, 2015.

[7] X. Bei, W. Chen, and J. Zhang. Distributed consensus resilient to both crash failures and strategic. In *http://arxiv.org/abs/1203.4324; version 3*, 2012.

[8] Keren Censor-Hillel, Bernhard Haeupler, Jonathan Kelner, and Petar Maymounkov. Global computation in a poorly connected world: Fast rumor spreading with no dependence on conductance. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 961–970, New York, NY, USA, 2012. ACM.

[9] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proc. of the 6th Ann. ACM Symposium on Principles of Distributed Computing (PODC'12)*, pages 1–12. ACM, 1987.

[10] B. Doerr, L. A. Goldberg, L. Minder, T. Sauerwald, and C. Scheideler. Stabilizing consensus with the power of two choices. In *Proc. of the 23rd Ann. ACM Symp. on Parallelism in Algorithms and Architectures (SPAA'11)*, pages 149–158. ACM, 2011.

[11] Andrew V. Goldberg and Jason D. Hartline. Collusion-resistant mechanisms for single-parameter agents. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 620–629, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[12] A. Groce, J. Katz, Thiruvengadam, and V. Zikas. Byzantine agreement with a rational adversary. In *Proc. 39th ICALP, LNCS*, pages 561–572, 2012.

[13] Joseph Halpern and Vanessa Teague. Rational secret sharing and multi-party computation. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 623–632. ACM, 2004.

[14] J.Y. Halpern and X. Vilaca. Rational consensus: Extended abstract. In *Proc. ACM PODC'16*, pages 561–572, 2016.

[15] Yehuda Hassin and David Peleg. Distributed probabilistic polling and applications to proportionate agreement. *Inf. Comput.*, 171(2):248–268, January 2002.

[16] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *Proc. of the 41th Ann. IEEE Symp. on Foundations of Computer Science (FOCS'00)*, pages 565–574. IEEE, 2000.

[17] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-Based Computation of Aggregate Information. In *Proc. of 43rd Ann. IEEE Symp. on Foundations of Computer Science (FOCS'03)*, pages 482–491. IEEE, 2003.

[18] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, 2015.

[19] Devavrat Shah. Gossip algorithms. *Found. Trends Netw.*, 3(1):1–125, January 2009.