Approximate Majority with Catalytic Inputs

Talley Amir
Yale University, New Haven, CT, USA
talley.amir@yale.edu

James Aspnes
Yale University, New Haven, CT, USA
james.aspnes@gmail.com

John Lazarsfeld
Yale University, New Haven, CT, USA
john.lazarsfeld@yale.edu

Abstract

Population protocols [6] are a class of algorithms for modeling distributed computation in networks of finite-state agents communicating through pairwise interactions. Their suitability for analyzing numerous chemical processes has motivated the adaptation of the original population protocol framework to better model these chemical systems. In this paper, we further the study of two such adaptations in the context of solving approximate majority: persistent-state agents (or *catalysts*) and spontaneous state changes (or *leaks*).

Based on models considered in recent protocols for populations with persistent-state agents [3, 5, 14], we assume a population with n catalytic input agents and m worker agents, and the goal of the worker agents is to compute some predicate over the states of the catalytic inputs. We call this model the Catalytic Input (CI) model. For $m = \Theta(n)$, we show that computing the parity of the input population with high probability requires at least $\Omega(n^2)$ total interactions, demonstrating a strong separation between the CI model and the standard population protocol model. On the other hand, we show that the simple third-state dynamics [7, 20] for approximate majority in the standard model can be naturally adapted to the CI model: we present such a constant-state protocol for the CI model that solves approximate majority in $O(n \log n)$ total steps with high probability when the input margin is $\Omega(\sqrt{n \log n})$.

We then show the robustness of third-state dynamics protocols to the transient leaks events introduced by [3, 5]. In both the original and CI models, these protocols successfully compute approximate majority with high probability in the presence of leaks occurring at each step with probability $\beta \leq O\left(\sqrt{n \log n}/n\right)$. The resilience of these dynamics to leaks exhibits similarities to previous work involving Byzantine agents, and we define and prove a notion of equivalence between the two.

24th International Conference on Principles of Distributed Systems (OPODIS 2020).
Editors: Quentin Bramas, Rotem Oshman, and Paolo Romano; Article No. 19; pp. 19:1–19:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1   Introduction

The population protocol model [6] is a theoretical framework for analyzing distributed computation in ad hoc networks of anonymous, mobile agents: at each step, a random pair of agents is chosen to interact, and their local states are updated according to a global transition function. Population protocols can solve numerous problems in distributed computing, including majority (which is also referred to as consensus) [5, 7, 12], source detection [3, 14], and leader election [4, 16, 17].

Population protocols are a special case of chemical reaction networks (CRNs), which are systems of transition rules describing how a set of chemical reactants stochastically transform into a set of products. In particular, population protocols are chemical reaction networks with exactly two reactants which form two products, where each transition rule for a pair of reactants is weighted with probability 1. Given their suitability for modeling chemical processes, population protocols have been used to study computation not only by chemical reaction networks [10], but also DNA strand displacement [11, 21] and biochemical networks [9]. These applications of population protocols in chemistry have inspired various adaptations of the model. In this paper, we focus on two such variations on population protocols in the context of solving **majority**, the problem of determining which of two states is initially more prevalent in a population.

The first modification to the model we consider, which was introduced to the literature in previous works studying source detection and bit-broadcast [3, 14] and later studied in the context of the majority problem [5, 13], is the presence of **persistent-state agents**, or agents whose state never changes. While some works use persistent-state agents to model authoritative sources of information [14] or "stubborn" nodes that are unwilling to change state [13], others describe these entities as an embodiment of chemical catalysts because they induce a state transition in another agent without themselves changing state [3, 5]. Using the latter perspective, we refer to these persistent-state agents as **catalysts**.

In this work, we call the class of population protocols with catalysts the **catalytic input (CI) model**. We formally define the model to consist of n **catalytic input agents**, which in accordance with their name do not ever change state, and m **worker agents** that can change state and wish to compute some function on the states of the catalysts. While the CI model is similar to the standard population protocol model, we show that there exists a strong separation between the two in terms of their computational power.

The next variation on the model we consider is the introduction of transient leak events, studied previously in the contexts of solving source detection and comparison [3, 5]. In brief, a "leak" simulates the low-probability event that a molecule undergoes a reaction that would typically take place in the presence of a catalyst. In population protocols, this is modeled by a spontaneous change of state at a single agent, and note that catalytic agents in the CI model are not susceptible to leaks because they never change state. A leak replaces an interaction between two agents at any given step with some fixed probability, known as the **leak rate** [3]. Although leaks have typically been studied in the presence of catalysts, we consider leaks to more generally model unpredictable or adversarial behavior which may occur in the absence of catalysts as well.

We explore the impact of leaks on third-state dynamics [7, 20] solving majority. Our work demonstrates that third-state dynamics can solve **approximate majority**, or majority with a lower-bounded initial difference between the counts of the two input states, with upper-bounded leak rate both in the standard and CI population models.

1.1 Related Work

The third-state dynamics protocol in the original population model (sometimes called *undecided-state dynamics*) was introduced by Anlguin et al. [7] and independently by Perron et al. [20]. An agent is either in a state X or Y, or in a *blank* state B (sometimes called an *undecided* state). The transition rules are shown in Figure 1, and we refer to this protocol as DBAM[1]. Assuming an initial X majority, a simplified analysis from Condon et al. [12] showed that all n agents in the population transition to the X state within $O(n \log n)$ total interactions with high probability, so long as the input margin $|X| - |Y|$ at the start of the protocol is at least $\Omega(\sqrt{n \log n})$. The DBAM protocol is also robust to a small subset of faulty *Byzantine* agents [7, 12], meaning that all but a $O(\sqrt{n \log n}/n)$ fraction of the population still reaches the X state within $O(n \log n)$ interactions with high probability, despite the presence of these dishonest agents.

The DBAM protocol and similar variants of third-state dynamics have been shown to more generally compute consensus (where all agents converge to either X or Y, but where this need not be the initial majority value), both in the original population protocols model [7, 12] and in other similar distributed models [8, 13]. In particular, the closely related results of d'Amore et al. [13] analyzed an analogous version of the DBAM protocol in the synchronous PULL model. The authors considered systems with *stubborn* agents (as in [24]) which are similar to the persistent-state catalytic agents we consider in the present work. However, the parallel synchronous scheduling model considered in [13] is fundamentally distinct from the sequential pairwise scheduling used in population protocols.

The notion of a persistent *source* state in population protocols originated from [14], where sources are used to solve **detection** (the detection of a source in the population) and **bit broadcast** (the broadcast of a 0 or 1 message from a set of source agents). An accompanying work [3] introduces the concept of leaks, or spontaneous state changes, and investigates the detection problem in their presence. Generally, leaks can be dealt with using error-correcting codes [22]; however, for certain problems there are more efficient specialized solutions. For example, Alistarh et al. [3] demonstrate that detection in the presence of leaks (up to rate $\beta = O(1/n)$) can be solved with high probability using $\log \frac{n}{k} + O(\log \log n)$ states, where $k \leq n$ is the number of sources in the population.

More recently, [5] examines leaks in the context of the **comparison** problem. Comparison is a generalization of the majority problem, where some possibly small subset of the population is in input state X_0 or Y_0 and the task of the population is to determine which of the two states is more prevalent. Alistarh et al. [5] solve comparison in $O(n \log n)$ interactions with high probability using $O(\log n)$ states per agent, assuming $|X_0| \geq C|Y_0|$ for some constant C, and $X_0, Y_0 \geq \Omega(\log n)$. The protocol is self-stabilizing, meaning that it dynamically responds to changes in the counts of input states.

1.2 Our Contribution

In this work, motivated by the recent interest in population models with catalytic agents and with transient leaks, we study the well-known third-state dynamics protocols [7, 20] for solving approximate majority in the presence of each of these variants separately as well

---

[1] DBAM stands for *double-B approximate majority* where *double-B* captures the fact that following an $X + Y$ interaction, both agents transition to the B state. This protocol is the two-way variant of the original protocol from [7], which uses one-way communication and where only one agent updates its state per pairwise interaction.

$$X + B \to X + X$$
$$Y + B \to Y + Y$$
$$X + Y \to B + B$$

$$X + B \to X + X \qquad I_X + B \to I_X + X$$
$$Y + B \to Y + Y \qquad I_Y + B \to I_Y + Y$$
$$X + Y \to B + B$$

■ **Figure 1** Transition rules for the `DBAM` protocol [7] in the original population model.

■ **Figure 2** Transition rules for our `DBAM-C` protocol in the CI model.

as together. To begin, we formalize the CI model consisting of n catalysts and m workers, where $N = n + m$. While conceptually similar to other models considering these types of catalytic agents [3, 5, 13], introducing the distinction between the two (possibly unrelated) population sizes provides a new level of generality for designing and analyzing protocols in this setting, both with and without leaks.

Although the CI and original population models are almost identical, we show a strong separation between the computational power of the two. When $m = \Theta(n)$, we prove a lower bound showing that **parity**, the problem of determining whether the number of catalysts is odd or even, cannot be computed in fewer than $\Omega(n^2)$ interactions with high probability in the CI model. On the other hand, the result of [19] shows that this predicate *is* computable within $O(n \operatorname{polylog} n)$ total steps in the standard model with high probability[2].

While some problems have strictly different lower bounds on running time in these two models, others do not and can in fact be solved using nearly identical techniques. In particular, we show that the approximate majority problem can be solved in the CI model by naturally extending the `DBAM` protocol.

In the approximate majority problem in the CI model, each catalytic input agent holds a persistent value of I_X or I_Y and each worker agent holds either an undecided, or blank value B, or an X or Y value corresponding to a belief in an I_X or I_Y input majority, respectively. The worker agents seek to correctly determine the larger of $|I_X|$ and $|I_Y|$ so long as the input margin $||I_X| - |I_Y||$ is sufficiently large. By adapting the third-state dynamics process [7], we present a constant-state protocol for approximate majority with catalytic inputs called `DBAM-C` (see Figure 2). The protocol converges with high probability in $O(N \log N)$ total steps when the initial input margin is $\Omega(\sqrt{N \log N})$ and $m = \Theta(n)$. We then show that this input margin is optimal in the CI model up to a $O(\sqrt{\log N})$ factor when $m = \Theta(n)$.

Moreover, in the presence of transient leak events, we show that both the third-state dynamics protocol in the original model and our adapted protocol in the CI model exhibit a strong robustness to leaks. When the probability of a leak event is bounded, we show that with high probability both protocols still quickly reach a configuration where nearly all agents share the correct input majority value.

Notice that the approximate majority problem in the CI model is equivalent to the comparison problem considered by [5], so we demonstrate how our protocol compares to the results of this work. We show that our `DBAM-C` protocol converges correctly within the same time complexity of $O(n \log n)$ total steps, while only using *constant* state space (compared to the logarithmic state used by the protocols in their work). Moreover, in populations where $m = \Theta(n)$, our protocol tolerates a less restrictive bound on the input margin compared to [5] ($\Omega(\sqrt{n \log n})$ compared to $\Omega(n)$). In the presence of transient leaks, our protocol also shows

---

[2] We define "high probability" to mean with probability at least $1 - n^{-c}$ where n is the total number of agents and $c \geq 1$.

robustness to a higher leak rate of $\beta \leq O(\sqrt{n \log n}/n)$. However, unlike [5], our protocol is not self-stabilizing and requires that the number of inputs be at least a constant fraction of the total population for our main results. In order to achieve these results, we leverage the random walk analysis techniques and analysis structure introduced by [12].

Finally, we compare the impact of leaks on population protocols with that of faulty Byzantine processes. While the fast robust approximate majority protocol of [7] is proven to be robust to a number of Byzantine agents that is bounded by the input margin [7, 12], we show that `DBAM` is robust to a similarly bounded leak rate and has sampling error matching the result from [12].

The structure of the remainder of the paper is as follows: in Section 2 we introduce notation and definitions central to our results. Section 3 presents our lower bounds over the CI model, which demonstrates the separation between the CI and original population models. In Section 4, we analyze the correctness and efficiency of the `DBAM-C` protocol for approximate majority in the CI model, and in Section 5 we demonstrate the leak-robustness of both the `DBAM-C` and original `DBAM` protocols. Then in Section 6, we compare the notion of transient leaks with the adversarial Byzantine model, demonstrating parallels between previous results examining Byzantine behavior and our work.

Throughout the paper, we provide overviews of the intuition and techniques used to obtain our results and defer most proofs to the full version.

2 Preliminaries

We begin with some definitions. Denote by N the number of agents in the population.

Population Protocols

Population protocols are a class of algorithms which model interactions between mobile agents with limited communication range. Agents only interact with one another if they are within close enough proximity of each other. In order to model this type of system in an asynchronous setting, interactions between pairs of agents are executed in sequence. The interaction pattern of these agents is dictated by a **scheduler**, which may be random or adversarial. In this work we will assume that the scheduler is uniformly random, meaning that an ordered pair of agents is chosen to interact at each time step independently and uniformly at random from all $N(N-1)$ ordered pairs of agents in the system.

As defined by [6] which first introduced the model, a population protocol $\mathcal{P}$ consists of a **state set** $\mathcal{S} = \{s_1, s_2, ..., s_k\}$, a **rule set** $\mathcal{R} : \mathcal{S}^2 \mapsto \mathcal{S}^2$, an **output alphabet** $\mathcal{O}$, and an **output function** $f : \mathcal{S} \mapsto \mathcal{O}$. The output function computes the evaluation of some function on the population locally at each agent. The **configuration** of the population is denoted as a vector $\mathbf{c} = \langle c_1, c_2, ..., c_k \rangle$ such that each $c_i \geq 0$ is equal to the number of agents in the population in state s_i, from which it follows that $\sum c_i = N$. For convenience, we denote by $|s_i|$ the number of agents in the population in state s_i.

At each point in time, the scheduler chooses an ordered pair of agents (a_i, a_j), where a_i is the **initiator** and a_j is the **responder** [6]. The agents interact and update their state according to the corresponding rule in $\mathcal{R}$. In general, a rule in $\mathcal{R}$ is written as $A+B \longrightarrow C+D$ to convey that two agents, an initiator in state A and a responder in state B, interact and update their states to be C and D, respectively. By convention, $N/2$ interactions make one unit of **parallel time** [7]. This convention is equivalent to assuming every agent interacts once per time unit on average.

An **execution** is the sequence of configurations of a run of the protocol, which **converges** when the population arrives at a configuration **d** such that all configurations chronologically after **d** have the same output at each agent as those in **d** [6]. In order to determine the success or failure of an execution of $\mathcal{P}$, we will consider a sample of the population to signify the outcome of the protocol [3]. After the expected time to converge, one agent is selected at random and its state is observed. The output associated with the agent's state is considered the output of the protocol. The probability of sampling an agent whose state does not reflect the desired output of the protocol is called the **sample error rate**. Multiple samples can be aggregated to improve the rate of success.

Catalysts and Leaks

Following [3], in an interaction of the form $A + B \longrightarrow A + D$, we say A *catalyzes* the transformation of the agent in state B to be in state D. If A catalyzes every interaction it participates in, A is referred to as a **catalyst**.

In chemistry, a reaction that occurs in the presence of a catalyst also occurs at a lower rate in the absence of that catalyst. For this reason, recent work in DNA strand displacement, chemical reactions networks, and population protocols [3, 5, 21] have studied the notion of *leakage*: When a catalytic reaction $A + B \longrightarrow A + D$ is possible, then there is some probability that a transition $B \longrightarrow D$ can occur without interacting with A at all. This type of event, called a **leak**, was introduced in [21].

The probability with which the non-catalyzed variation of a reaction takes place is the **leak rate**, which we denote by β. We simulate a leak as follows: At each step in time, with probability $1 - \beta$, the scheduler samples an ordered pair of agents to interact with one another as described in the beginning of the section; the rest of the time (i.e. with probability β) one agent is chosen uniformly at random from all possible agents and the **leak function** $\ell : \mathcal{S} \to \mathcal{S}$ is applied to update this agent's state. Note that we only consider *non-catalytic* agents to be susceptible to these faulty events.

Catalytic Input Model

In this work, we formalize a **catalytic input** (CI) model consisting of n catalytic agents that supply the input and m worker agents that perform the computation and produce output. We define $N = m + n$ to be the total number of agents in the population. At each time step, the scheduler samples any two agents in the population to interact with one another. If two catalysts are chosen to interact, then the interaction is considered to be **null** as no nontrivial state transition occurs. When $n = o(m)$, the probability that two catalysts are chosen to interact is upper bounded by a constant, and so the total running time of the protocol is asymptotically equivalent to the number of non-null interactions needed to reach convergence. In the CI model, we consider convergence to be a term that refers to the states of the worker agents only, as the catalytic agents never change state. Namely, for the approximate majority problem, successful convergence equates to the *worker agents* being in the majority-accepting state. In general, we wish to obtain results that hold with high probability with respect to the *total* number of agents N.

3   Catalytic Input Model Lower Bounds

In this section, we characterize the computational power of the CI population protocol model. Using information-theoretic arguments, we prove two lower bounds over the catalytic model when the number of input agents is a constant fraction of the total population:

▶ **Theorem 1.** *In the catalytic input model with n input agents and $m = \Theta(n)$ worker agents, any protocol that computes the parity of the inputs with probability at least $1 - N^{-\gamma}$ requires at least $\Omega(N^2)$ total steps for any $\gamma \geq 1$.*

▶ **Theorem 2.** *In the catalytic input model with n input agents and $m = \Theta(n)$ worker agents, any protocol that computes the majority of the inputs within $O(N \log N)$ total steps requires an input margin of at least $\Omega(\sqrt{N})$ to be correct with probability at least $1 - N^{-\gamma}$ for any $\gamma \geq 1$.*

The first result can be viewed as a separation between the CI and original population models: since it is shown in [19] that the parity of agents can be computed in the original model within $O(\text{polylog } n)$ parallel time with high probability, our result indicates that not all semi-linear predicates over the input population in the CI model can be computed in sub-linear parallel time with high probability. Additionally, this rules out the possibility of designing fast protocols for exact majority in the CI model when the input size is a constant fraction of the entire population. On the other hand, the second result indicates the existence of a predicate – approximate majority – that does not require a large increase in convergence time to be computed with high probability in this new model.

One key characteristic of a CI population is the inability for worker agents to distinguish which inputs have previously interacted with a worker. Instead, every worker-input interaction acts like a random sample with replacement from the input population. For proving lower bounds in this model, this characteristic of a CI population leads to the following natural argument: consider a population of n catalytic input agents and a worker population consisting of a single **super-agent**. Here, we assume the super-agent has unbounded state and computational power, and it is thus able to simulate the entire worker population of any protocol with more workers. In this simulation, any interaction between a worker and an input agent is equivalent to the super-agent interacting with an input chosen uniformly at random: in other words, as a sample with replacement from the input population. Thus we view the super-agent as running a central randomized algorithm to simulate the random interactions that occur in population protocols. If the super-agent needs S samples to compute some predicate over the inputs with high probability, then so does any multi-worker protocol in the CI model. We denote this information-theoretic model as the **Super CI model**, and restate the above argument more formally in the following lemma.

▶ **Lemma 3.** *Consider a population with n catalytic input agents and a worker population consisting of a single super-agent W. Let P be a predicate over the input population that requires S total interactions between W and the input population in order for W to correctly compute P with probability ϵ. Then for a CI population with n catalytic inputs and m worker agents, computing P correctly with probability ϵ requires at least S total interactions.*

Proof Sketch of Theorem 1

In a CI model population with n input agents and m worker agents where $m = \Theta(n)$, Theorem 1 shows that computing the parity or exact majority of the inputs requires at least $\Omega(n^2) = \Omega(N^2)$ total interactions to be correct with high probability. We prove this by showing that in the Super CI model described in the previous section, a computationally unbounded super-agent W requires at least $\Omega(n^2)$ samples of the input population to correctly compute the input parity with high probability. Applying Lemma 3 then gives Theorem 1.

More formally, for an input population C of n agents, each with input value 0 or 1, the parity of C is said to be 1 if an odd number of agents have input value 1, and 0 otherwise. Now, consider the majority predicate over C, which is simply the majority value of the input

population. Letting X denote the number of 1-inputs, and Y the number of 0-inputs, we refer to the *input margin* of the population C as the quantity $|X - Y|$. Suppose that n is odd and the input margin of C is 1. Then X and Y are either $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$ or vice versa. These two cases can be distinguished either by computing the majority predicate or the parity predicate, making both of these problems equivalent to distinguishing the two cases under this constraint on the input. We will now argue that distinguishing these cases in the Super CI model requires $\Omega(n^2)$ samples.

Recall that in the Super CI model, a predicate over the input population C is computed by a single super agent worker W with unbounded computational power. Thus, the output of W can be viewed as a mapping between a string of input values obtained from interactions with between W and the input population and the output set $\{0, 1\}$. We refer to interactions between W and the input population as **samples** of the input, and for a fixed number of samples S, we refer to W's output as its **strategy**.

First, we show that for some fixed distribution over the input values of C, the strategy that maximizes W's probability of correctly outputting the majority value of C is simply to output the majority value of its samples. Let $I \in \{0, 1\}^S$ be the **sample string** representing the S independent samples with replacement taken by W, and let $\mathbb{S}$ denote the set of all 2^S possible sample strings. We model the population of input agents as being generated by an adversary. Specifically, let M denote the majority value (0 or 1) of the input population, where we treat M as a a random variable whose distribution is unknown. In any realization of M, we assume a fixed fraction $p > 1/2$ of the inputs hold the majority value. Given an input population, the objective of the worker agent is to correctly determine the value of M through its input sample string I. By Yao's principle [23], the error of any randomized algorithm (i.e., the randomized simulation run by the super-agent) on the worst case value of M is no smaller than the error of the best deterministic algorithm on some fixed distribution over M. So our strategy is to pick a distribution over M, and to use the the error of the best deterministic strategy with respect to this distribution as a lower bound on the worst-case error of any randomized algorithm used by the super-agent.

Thus, assuming M is chosen according to some fixed distribution, we model the worker's strategy as a fixed map $f : \{0, 1\}^S \rightarrow \{0, 1\}$. Letting $\mathcal{F}_S$ denote the set of all such maps, W then faces the following optimization problem: $\max_{f \in \mathcal{F}_S} \Pr[f(I) = M]$. For a given $f \in \mathcal{F}_S$, let $p_f = \Pr[f(I) = M]$, and let $\Phi \in \mathcal{F}_S$ denote the map that outputs the majority value of the input sample string I. In the following lemma, we show that when the distribution over M is uniform, setting $f := \Phi$ maximizes p_f. In other words, to maximize the probability of correctly guessing the input population majority value, the worker's optimal strategy is to simply guess the majority value of its S independent samples. The proof of the lemma simply uses the definitions of conditional probability and the Law of Total Probability to obtain the result.

▶ **Lemma 4.** *Let $I = \{0, 1\}^S$ be a sample string of size S drawn from an input population with majority value M and majority ratio p, and assume $\Pr[M = 1] = \Pr[M = 0] = 1/2$. Then $\Pr[\Phi(I) = M] \geq \Pr[f(I) = M]$ for all maps $f \in \mathcal{F}_S$, where Φ is the map that outputs the majority value of the sample string I.*

We have established by Lemma 4 that to correctly output the input population majority, the super worker agent's error-minimizing strategy is to output the majority of its S samples when the distribution over M is uniform. Now the following lemma shows that when the input margin of the population is 1, this strategy requires at least $\Omega(n^2)$ samples in order to output the input majority with probability at least $1 - n^{-c}$ for some constant $c \geq 1$. The proof uses a tail bound on the Binomial distribution to show the desired trade off between the error of probability and the requisite number of samples needed to achieve this error.

▶ **Lemma 5.** *Let C be a Super CI population of n agents with majority value M and input margin 1, and consider an input sample string $I = \{0, 1\}^S$ obtained by a super worker agent W. Then for any $c \geq 1$, letting $\Phi(I)$ denote the sample majority of I, $\Pr[\Phi(I) \neq M] \leq n^{-c}$ only holds when $S \geq \Omega(n^2)$.*

The proof of Theorem 1 follows from Lemmas 3, 4, and 5 by invoking Yao's principle.

Overview of Theorem 2

As mentioned, Theorem 1 implies a strong separation between the CI model and original population model, as [19] and [1,2] have shown that both parity and majority are computable with high probability within $O(n \operatorname{polylog} n)$ total steps in the original model, respectively. Thus, the persistent-state nature of input agents in the CI model may seem to pose greater challenges than in the original model for computing predicates quickly with high probability. However, using the same sampling-based lower bound techniques developed in the preceding section, Theorem 2 shows that when $m = \Theta(n)$, and when restricted only to $S = O(n \log n)$ total steps, any protocol computing majority in the CI model requires an input margin of at least $\Omega(\sqrt{n}) = \Omega(\sqrt{N})$ to be correct with high probability in N.

Moreover, in Section 4 we present a protocol for approximate majority in the CI model that converges correctly with high probability within $O(N \log N)$ total steps, so long as the initial input margin is $\Omega(\sqrt{N \log N})$. Thus, the existence of such a protocol indicates that the $\Omega(\sqrt{N})$ lower bound on the input margin is nearly tight (up to $\sqrt{\log N}$ factors) for protocols limited to $O(N \log N)$ total steps when $m = \Theta(n)$.

4 Approximate Majority with Catalytic Inputs

We now present and analyze the `DBAM-C` protocol for computing approximate majority in the CI model. The protocol is a natural adaptation of a third-state dynamics from the original model, where we now account for the behavior of n catalytic input agents and m worker agents. Using the CI model notation introduced in Section 2, we consider a population with $N = n + m$ total agents. Each input agent begins (and remains) in state I_X or I_Y, and we assume each worker agent begins in a *blank* state B, but may transition to states X or Y according to the transition rules found in Figure 1. Letting i_X and i_Y (and similarly x, y and b) be random variables denoting the number of agents in states I_X and I_Y (and respectively X, Y, and B), we denote the *input margin* of the population by $\epsilon = |i_X - i_Y|$. Throughout the section, we assume without loss of generality that $i_X \geq i_Y$.

Intuitively, an undecided (blank) worker agent adopts the state of a decided agent (either an input or worker), but decided workers only revert back to a blank state upon interactions with other workers of the opposite opinion. Thus the protocol shares the opinion-spreading behavior of the original `DBAM` protocol, but note that the inability for decided worker agents to revert back to the blank state upon subsequent interactions with an input allows the protocol to converge to a configuration where all workers share the same X or Y opinion.

The main result of the section characterizes the convergence behavior of the `DBAM-C` protocol when the input margin ϵ is sufficiently large. Recall that we say the protocol *correctly computes* the *majority* of the inputs if we reach a configuration where $x = m$. The following theorem shows that, subject to mild constraints on the population sizes, when the input margin is $\Omega(\sqrt{N \log N})$, the protocol correctly computes the majority value of the inputs in roughly logarithmic parallel time with high probability.

▶ **Theorem 6.** *There exists some constant $\alpha \geq 1$ such that, for a population of n inputs, m workers, and initial input margin $\epsilon \geq \alpha\sqrt{N \log N}$, the `DBAM-C` protocol correctly computes the majority value of the inputs within $O\left(\frac{N^4}{m^3} \log N\right)$ total interactions with probability at least $1 - N^{-c}$ for any $c \geq 1$ when $m \geq n/10$ and N is sufficiently large.*

Because the CI model allows for distinct (and possibly unrelated) input and worker population sizes, we aim to characterize all error and success probabilities with respect to the *total* population size N. The analysis in the proof of Theorem 6 characterizes the convergence behavior of the protocol in terms of both population sizes m and n, and thus the convergence time of $O((N^4/m^3) \log N)$ is not always equivalent to $O(N \log N)$. On the other hand, in the case when $m = \Theta(n)$ – which is an assumption used to provide lower bounds over the CI model from Section 3 – we have as a corollary (further below) that the protocol correctly computes the majority of the inputs within $O(N \log N)$ total steps with probability at least $1 - N^{-\alpha}$.

Analysis Overview

The proof of the main result leverages and applies the random walk tools from [12] (in their analysis of the original `DBAM` protocol) to the `DBAM-C` protocol. Given the uniformly-random behavior of the interaction scheduler, the random variables x, y and b (which represent the count of X, Y, and B worker agents in the population) each behave according to some one-dimensional random walk, where the biases in the walks change dynamically as the values of these random variables fluctuate. Based on the coupling principle that an upper bound on the number of steps for a random walk with success probability p to reach a certain position is an *upper bound* on the step requirement for a second random walk with probability $\hat{p} \geq p$ to reach the same position, we make use of several *progress measures* that give the behavior of the protocol a natural structure. As used in the analysis of Condon et al. [12], we define $\hat{x} = x + b/2$, $\hat{y} = y + b/2$, and $P = \epsilon + \hat{x} - \hat{y}$. It can be easily seen that $\hat{x} + \hat{y} = m$ will hold throughout the protocol. On the other hand, the progress measure P captures the collective gap between the majority and non-majority opinions in the population. Observe that the protocol has correctly computed the input majority value when $P = \epsilon + m$ and $\hat{y} = 0$.

Our analysis uses a structure of **phases** and **stages** to prove the correctness and efficiency of the protocol. Every correctly-completed stage of Phase 1 results in the progress measure P doubling, and the phase completes correctly once P is at least ϵ plus some large constant fraction of m. Then, every correctly-completed stage of Phase 2 results in the progress measure $\hat{y}$ decreasing by a factor of two, and the phase completes correctly once $\hat{y}$ drops to $O(\log m)$. Finally, Phase 3 of the protocol ends correctly once $\hat{y}$ drops to 0. The details of this structure are stated formally in the full version of the paper.

Note that among the protocol's non-null transitions (see Figure 2), only the interactions $I_X + B$, $I_Y + B$, $X + B$, and $Y + B$ change the value of either progress measure. For this reason, we refer to the set of non-null transitions (which includes $X + Y$ interactions) as **productive** steps, and the subset of interactions that change our progress measures as the set of **blank-consuming** productive steps The analysis strategy for every phase and stage is to employ a combination of standard Chernoff bounds and martingale techniques (in general, see [18] and [15]) to obtain with-high-probability estimates of (1) the number of productive steps needed to complete each phase/stage correctly, and (2) the number of total steps needed to obtain the productive step requirements. Given an input margin that is sufficiently large, and also assuming a population where the number of worker agents is at least a small constant fraction of the input size, we can then sum over the error probabilities of each phase/stage and apply a union bound to yield the final result of Theorem 6.

While the `DBAM-C` protocol is conceptually similar to the original `DBAM` protocol, the presence of persistent-state catalysts whose opinions never change requires a careful analysis of the convergence behavior. Moreover, simulation results presented in Section 5 show interesting differences in the evolution of the protocol for varying population sizes. As a simple corollary of Theorem 6, we also state the following result, which simplifies the convergence guarantees of the `DBAM-C` protocol in the case when $m = \Theta(n)$.

▶ **Theorem 7.** *There exists some constant $\alpha \geq 1$ such that, for a population of n inputs, $m = cn$ workers where $c \geq 1$, and an initial input margin $\epsilon \geq \alpha\sqrt{N \log N}$, the `DBAM-C` protocol correctly computes the majority of the inputs within $O(N \log N)$ total interactions with probability at least $1 - N^{-a}$ for any $a \geq 1$ when N is sufficiently large.*

We note that the result of Corollary 7 implies that the CI model input margin lower bound from Theorem 2 is tight up to a multiplicative $O(\sqrt{\log N})$ factor.

5 Approximate Majority with Transient Leaks

We now consider the behavior of the `DBAM` and `DBAM-C` protocols in the presence of transient leak faults. Even in the presence of these adversarial events (which occur up to some bounded rate β), both the `DBAM` and `DBAM-C` protocols will, with high probability, reach configurations where nearly all agents share the input majority opinion. In the presence of leaks, we consider the approximate majority predicate to be computed correctly upon reaching these low sample-error configurations.

Recall that a transient leak is an event where an agent spuriously changes its state according to some leak function ℓ. For example, we denote by $U \to V$ the event that an agent in state U transitions to state V due to a leak event, where the timing of such events are dictated by the random scheduler and occur with probability β at each subsequent interaction step. In both the `DBAM` and `DBAM-C` protocols, the only state changes that could possibly take place due to leaks are $X \to B$, $Y \to B$, $B \to X$, and $B \to Y$ because these describe all possible state changes that could take place in the presence of an interacting partner. However, our analysis considers an *adversarial* leak event $X \to Y$, which maximally decreases our progress measures and can be considered the "worst" possible leak. Though this leak event is not chemically sound (because no normal interaction can cause an X agent to transition to the Y state), our results demonstrate that both `DBAM` and `DBAM-C` protocols are robust to this strong adversarial leak event. Thus in a more realistic chemically sound setting, our results will also hold, as the set of transitions working against our progress measures are weaker.

Leak Robustness of the `DBAM` Protocol

We start by showing the leak-robustness of the `DBAM` protocol for approximate majority in the original population protocol model. Recall that in the standard model, *all* agents are susceptible to leaks. Our main result shows that when the leak rate β is sufficiently small, the protocol still reaches a configuration with bounded *sample error* (the proportion of agents in the non-initial-majority state) within $O(n \log n)$ total interactions with high probability. Unlike the scenario without leak events, note that the protocol will never be able to fully converge to a configuration where all agents remain in the majority opinion. However, reaching a configuration where despite leaks, *nearly* all agents hold the input majority value state matches similar results of [3, 5, 7, 12]. Formally, we have the following Theorem, which characterizes the eventual sample error of the protocol with respect to the magnitude of the leak rate β.

▶ **Theorem 8.** *There exists some constant $\alpha \geq 1$ such that, for a population with initial input margin $\epsilon \geq \alpha\sqrt{n \log n}$ and adversarial leak rate $\beta \leq (\alpha\sqrt{n \log n})/12672n$, an execution of the DBAM protocol will reach a configuration with*
1. *sample error $O(\log n/n)$ when $\beta \leq O(\log n/n)$*
2. *sample error $O(\beta)$ when $\omega(\log n/n) \leq \beta \leq (\alpha\sqrt{n \log n})/12672n$*
within $O(n \log n)$ total interactions with probability at least $1 - n^{-c}$ for any $c \geq 1$ when n is sufficiently large.

To prove Theorem 8, we again make modified use of the random walk tools from [12]. Using the progress measures $\hat{y} = y + b/2$ and $P = \hat{x} - \hat{y}$, observe that an $X \to Y$ leak event incurs twice as much negative progress to both measures as opposed to $X + B$ events. Compared to the analysis from the non-leak setting, the analysis *with* adversarial leaks must account for the stagnation (or potentially the reversal) of the protocol's progress toward reaching a high-sample-error configuration. Note that since the sample error of a configuration is defined to be $(y + b)/n$ (since we assume an initial x majority wlog), we will use the value $\hat{y}/n$ to approximate a configuration's sample error.

We use a similar structure of phases and stages as in the previous section, and we list these details formally in the full version of this work. In this leak-prone setting, we refer to *productive interactions* as any of the non-null transitions found in Figure 1 in addition to a leak event. The set of three non-null and non-leak transitions are referred to as *non-leak productive steps*. For each phase and stage, we obtain high-probability estimates on the number of productive and total steps needed to complete the phase/stage correctly in two steps: first, we bound the number of leak events that can occur during a fixed interval of productive events, and we then show that a smaller sub-sequence of non-leak productive steps is sufficient to ensure that enough progress is made to offset the negative progress of the leaks. We again rely on a combination of Chernoff concentration bounds and martingale inequalities in order to show this progress at every phase and stage.

Theorem 8 also separates the behavior of the protocol into two classes: when $\beta \leq O(\log n/n)$ (*small* leak rate), and when $\omega(\log n/n) \leq \beta \leq O(\sqrt{n \log n}/n)$ (*large* leak rate). When the leak rate is large, the probability of a leak event conditioned on a productive step becomes roughly equal to the conditional probability of a non-leak productive step when $\hat{y} = O(\beta n)$. Thus, we cannot expect the protocol to make further "progress" toward a lower sample-error configuration with high probability beyond $\hat{y} = O(\beta n)$. The same holds for small leak rate when $\beta = O(\log n/n)$, and for even smaller values of β, our analysis tools only allow for the high-probability guarantee that $\hat{y}$ eventually drops to $O(\log n)$. the protocol reaches a configuration with $\hat{y} = O(\log n)$. In the full version of the paper, we give additional arguments showing that the protocol remains in a configuration with sample error $O(\log n/n)$ for small leak rate, and with sample error $O(\sqrt{n \log n}/n)$ for large leak rate, for at least a polynomial number of interactions with high probability.

Leak Robustness of the DBAM-C Protocol

The analysis of the previous subsection is adapted to show that the DBAM-C protocol also exhibits a similar form of leak-robustness in the CI model, and in the following theorem we prove the case where $m = \Theta(n)$. Recall that in the CI model, only the non-catalytic worker agents are susceptible to leak events.

▶ **Theorem 9.** *There exist constants $\alpha, d \geq 1$ such that, for a population with $m = cn$ for $c \geq 1$ and input margin $\epsilon \geq \alpha\sqrt{N \log N}$, the DBAM-C protocol will reach a configuration with*
1. *sample error $O(\log N/N)$ when $\beta \leq O(\log N/N)$*

2. *sample error $O(\beta)$ when $\omega(\log N/N) \leq \beta \leq (\alpha\sqrt{N \log N})/dN$*
within $O(N \log N)$ total interactions with probability at least $1 - N^{-a}$ for $a \geq 1$ when N is
sufficiently large.

The proof of the theorem uses the same progress measures and phase and stage structure introduced in the non-leak setting in Section 4, and the final sample-error guarantee of the protocol is again defined with respect to the magnitude of the leak rate. The behavior of the protocol between the two classes of leak rate is similar as in the `DBAM` analysis, and the formal details can be found in the full version of the paper. Note that as the upper bound on the leak rate β is a decreasing function in N, the sample error guarantees of both protocols increase with population size. This relationship is shown across various simulations of the `DBAM-C` protocol in Figure 3. Moreover, Figure 4a depicts aggregate sample data over many executions of the `DBAM-C` protocol for varying values of N, and Figure 4b illustrates the logarithmic parallel time needed to reach convergence in the non-leak setting.
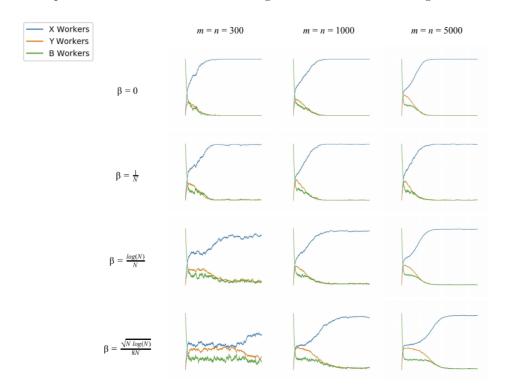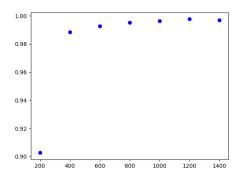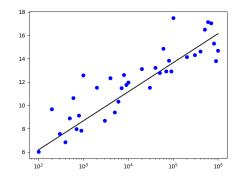

Figure 3 Simulations of the `DBAM-C` protocol, where each subplot shows the proportion of X, Y and B worker agents evolve over the course of an execution. All simulations are for $m = n$, input margin $\epsilon = \sqrt{N \log N}$ (with an I_X majority), and varying values of leak rate β over $4N \log N$ total interactions. Note that these plots are of single executions and thus provide a qualitative illustration of behavior, rather than statistically significant data. However, we can see that for larger values of $m = n$ and smaller values of β, the number of X worker agents reaches a larger count more quickly.

6 Leaks Versus Byzantine agents

The original third-state dynamics approximate majority protocol [7] is robust to a bounded number of Byzantine agents, and as shown in the previous sections, both the `DBAM` protocol and the `DBAM-C` protocol in the CI model are robust to a bounded leak rate. In this section,

(a) Sample success rate (y-axis) averaged over 3000 executions of `DBAM-C` for $n = 600$, $\Delta_0 = \sqrt{N \log N}$, and $\beta = 1/N$, and varying values of m (x-axis). Samples were drawn uniformly from the worker population after $4N \log N$ total interactions.

(b) Parallel time (y-axis) for `DBAM-C` without leaks to reach consensus for varying population sizes (x-axis) where $m = 2n$. Data points represent a single execution. The solid black line is $\frac{3}{4} \log_2(N)$, showing that convergence takes $O(N \log N)$ interactions.

Figure 4 Success rate and running time of `DBAM-C` over various executions of the protocol.

we consider the connection between these two types of faulty behavior. While leaks can occur at any agent with fixed probability throughout an execution, Byzantine agents are a fixed subset of the population, and while a leak event does not change the subsequent behavior of an agent, Byzantine agents may continue to misbehave forever. However, there are parallels between these two models of adversarial behavior. A leak at one agent can cause additional agents to deviate from a convergent configuration; similarly, interactions among non-Byzantine agents, some of which have deviated from a convergent configuration by interacting with a Byzantine agent, can cause additional non-Byzantine agents to diverge.

We prove that for the `DBAM` and `DBAM-C` protocols, introducing a leak rate of β has the same asymptotic effect as introducing $O(\beta N)$ Byzantine agents to the population, which demonstrates an equivalence between these two notions of adversarial behavior among the class of third-state dynamics protocols. Although the results of the previous section assumed leaks that do not follow the laws of chemistry, the following result considers **weak leaks**, which cause the selected agent to decrease its confidence in the majority value by one degree (i.e. a leak causes an agent in state X to transition to B and an agent in state B to transition to Y, matching the $X + Y$ and $Y + B$ transitions). For our purposes, we define two adversarial models $\mathcal{M}_1$ and $\mathcal{M}_2$ to be **equivalent** for some protocol $\mathcal{P}$ if $\mathcal{P}$ converges to the same asymptotic sample error rate in the same asymptotic running time in both models. We then have the following equivalence result:

▶ **Theorem 10.** *A population of N agents running* `DBAM` *(or* `DBAM-C`*) with weak leak rate $O(\beta)$ is equivalent to a population of $N + B$ agents, where $B = O(N\beta)$ agents are Byzantine, running* `DBAM` *(or* `DBAM-C`*) without leaks, where in either setting the protocol converges in $O(N \log N)$ interactions with error probability $O(\beta)$.*

7 Conclusion and Open Problems

We have shown that third-state dynamics can be used to solve approximate majority with high probability in $O(n \log n)$ steps up to leak rate $\beta = O(\sqrt{n \log n}/n)$, both in the standard population protocol model as well as the CI model when $m = \Theta(n)$. While we showed a separation between the CI and original population models, it remains an open question what

other problems (similar to approximate majority) can be computed quickly in the CI model. Additionally, identifying which families of protocols are naturally robust to leak events in the original population model (similar to third-state dynamics) also remains an open question.

─── **References** ───

1   Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L. Rivest. Time-space trade-offs in population protocols. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2560–2579. SIAM, 2017. `doi:10.1137/1.9781611974782.169`.

2   Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2221–2239. SIAM, 2018. `doi:10.1137/1.9781611975031.144`.

3   Dan Alistarh, Bartłomiej Dudek, Adrian Kosowski, David Soloveichik, and Przemysław Uznański. Robust detection in leak-prone population protocols. In *International Conference on DNA-Based Computers*, pages 155–171. Springer, 2017.

4   Dan Alistarh and Rati Gelashvili. Polylogarithmic-time leader election in population protocols. In *Proceedings, Part II, of the 42Nd International Colloquium on Automata, Languages, and Programming - Volume 9135*, ICALP 2015, pages 479–491, Berlin, Heidelberg, 2015. Springer-Verlag. `doi:10.1007/978-3-662-47666-6_38`.

5   Dan Alistarh, Martin Töpfer, and Przemysław Uznański. Robust comparison in population protocols, 2020. `arXiv:2003.06485`.

6   Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, pages 235–253, March 2006.

7   Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, 2008.

8   Luca Becchetti, Andrea Clementi, and Emanuele Natale. Consensus dynamics: An overview. *ACM SIGACT News*, 51(1):58–104, 2020.

9   Luca Cardelli and Attila Csikász-Nagy. The cell cycle switch computes approximate majority. *Scientific reports*, 2:656, September 2012. `doi:10.1038/srep00656`.

10 Ho-Lin Chen, David Doty, and David Soloveichik. Deterministic function computation with chemical reaction networks. *Nat. Comput.*, 13(4):517–534, 2014. `doi:10.1007/s11047-013-9393-6`.

11 Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik, and Georg Seelig. Programmable chemical controllers made from dna. *Nature nanotechnology*, 8, September 2013. `doi:10.1038/nnano.2013.189`.

12 Anne Condon, Monir Hajiaghayi, David Kirkpatrick, and Ján Maňuch. Approximate majority analyses using tri-molecular chemical reaction networks. *Natural Computing*, pages 1–22, 2019.

13 Francesco d'Amore, Andrea E. F. Clementi, and Emanuele Natale. Phase transition of a non-linear opinion dynamics with noisy interactions - (extended abstract). In Andrea Werneck Richa and Christian Scheideler, editors, *Structural Information and Communication Complexity - 27th International Colloquium, SIROCCO 2020, Paderborn, Germany, June 29 - July 1, 2020, Proceedings*, volume 12156 of *Lecture Notes in Computer Science*, pages 255–272. Springer, 2020. `doi:10.1007/978-3-030-54921-3_15`.

14 Bartlomiej Dudek and Adrian Kosowski. Universal protocols for information dissemination using emergent signals. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 87–99. ACM, 2018. `doi:10.1145/3188745.3188818`.

15   William Feller. *An introduction to probability theory and its applications. Vol. I.* Third edition. John Wiley & Sons Inc., New York, 1968.

16   Leszek Gasieniec and Grzegorz Stachowiak. Fast space optimal leader election in population protocols. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2653–2667. SIAM, 2018. `doi:10.1137/1.9781611975031.169`.

17   Leszek Gąsieniec, Grzegorz Stachowiak, and Przemyslaw Uznanski. Almost logarithmic-time space optimal leader election in population protocols. In *The 31st ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '19, pages 93–102, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3323165.3323178`.

18   Geoffrey R Grimmett and David R Stirzaker. *Probability and Random Processes*. Oxford University Press, 2001.

19   Adrian Kosowski and Przemysław Uznański. Population protocols are fast. *arXiv preprint arXiv:1802.06872*, 2018.

20   Etienne Perron, Dinkar Vasudevan, and Milan Vojnovic. Using three states for binary consensus on complete graphs. In *IEEE INFOCOM 2009*, pages 2527–2535. IEEE, 2009.

21   Chris Thachuk, Erik Winfree, and David Soloveichik. Leakless DNA strand displacement systems. In Andrew Phillips and Peng Yin, editors, *DNA Computing and Molecular Programming - 21st International Conference, DNA 21, Boston and Cambridge, MA, USA, August 17-21, 2015. Proceedings*, volume 9211 of *Lecture Notes in Computer Science*, pages 133–153. Springer, 2015. `doi:10.1007/978-3-319-21999-8_9`.

22   Boya Wang, Chris Thachuk, Andrew D. Ellington, Erik Winfree, and David Soloveichik. Effective design principles for leakless strand displacement systems. *Proceedings of the National Academy of Sciences*, 115(52):E12182–E12191, 2018. `doi:10.1073/pnas.1806859115`.

23   A. C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 222–227, 1977.

24   Ercan Yildiz, Asuman Ozdaglar, Daron Acemoglu, Amin Saberi, and Anna Scaglione. Binary opinion dynamics with stubborn agents. *ACM Transactions on Economics and Computation (TEAC)*, 1(4):1–30, 2013.