Robust Detection in Leak-Prone Population Protocols

Dan Alistarh[1,5], Bartłomiej Dudek[2], Adrian Kosowski[3], David Soloveichik[4],
and Przemysław Uznański[1(✉)]

[1] ETH Zürich, Zürich, Switzerland
{dan.alistarh,przemyslaw.uznanski}@inf.ethz.ch
[2] University of Wrocław, Wrocław, Poland
bartlomiej.dudek@cs.uni.wroc.pl
[3] Inria Paris and IRIF, Université Paris Diderot, Paris, France
adrian.kosowski@inria.fr
[4] University of Texas, Austin, TX, USA
david.soloveichik@utexas.edu
[5] IST Austria, Klosterneuburg, Austria

Abstract. In contrast to electronic computation, chemical computation is noisy and susceptible to a variety of sources of error, which has prevented the construction of robust complex systems. To be effective, chemical algorithms must be designed with an appropriate error model in mind. Here we consider the model of chemical reaction networks that preserve molecular count (population protocols), and ask whether computation can be made robust to a natural model of unintended "leak" reactions. Our definition of leak is motivated by both the particular spurious behavior seen when implementing chemical reaction networks with DNA strand displacement cascades, as well as the unavoidable side reactions in any implementation due to the basic laws of chemistry. We develop a new "Robust Detection" algorithm for the problem of fast (logarithmic time) single molecule detection, and prove that it is robust to this general model of leaks. Besides potential applications in single molecule detection, the error-correction ideas developed here might enable a new class of robust-by-design chemical algorithms. Our analysis is based on a non-standard hybrid argument, combining ideas from discrete analysis of population protocols with classic Markov chain techniques.

1 Introduction

A major challenge in designing autonomous molecular systems is to achieve a sufficient degree of error tolerance despite the error-prone nature of the chemical substrate. While considerable effort has focused on making the chemistry itself

more robust, here we look at the possibility of developing chemical algorithms that are inherently resilient to the types of error encountered. Before designing robust chemical algorithms, we must decide on a good error model that is relevant to the systems we care about. In this paper we focus on a very simple and general error model that is motivated both by basic laws of chemistry as well as by implementation artifacts in strand displacement constructions for chemical reaction networks. We begin by listing the types of errors we aim to capture.

Leaks due to Law of Catalysis. A fundamental law of chemical kinetics is that for every catalyzed reaction, there is an uncatalyzed one that occurs at a (often much) slower rate. By a *catalytic* reaction, we mean a reaction that involves some species X but does not change its amount; this species is called a *catalyst* of that reaction. For example, the reaction $X + Y \rightarrow X + Z$ is catalytic, and species X is the catalyst since its count remains unchanged by the firing of this reaction. By the law of catalysis, reaction $X + Y \rightarrow X + Z$ must be accompanied by a (slower) *leak* reaction $Y \rightarrow Z$. (A more general formulation of the law of catalysis is that if any sequence of reactions does not change the net count of X, then there is a pathway that has the same effect on all the other species, but can occur in the absence of X (possibly much slower). Thus for example, if $X + Y \rightarrow W$ and $W \rightarrow X + Z$ are two reactions, then there must also be a leak reaction $Y \rightarrow Z$. Formally defining catalytic cycles and catalysts is non-trivial and is beyond the scope of this paper [1].)

Leaks due to Law of Reversibility. Another fundamental law of chemical kinetics is that any reaction occurs also in the reverse direction at some (possibly much slower) rate. In other words, reaction $X + Y \rightarrow Z + W$ must be accompanied by $Z + W \rightarrow X + Y$. (The degree of reaction reversibility is related to the free-energy use, such that irreversible reactions would require "infinite" free energy.)

Leaks due to Spurious Activation in Strand-Displacement Cascades. Arbitrary chemical reaction networks can in principle be implemented with DNA strand displacement cascades [2,3]. Implementations based on strand displacement also suffer from the problem of leaks [4]. The implementation of a reaction like $X + Y \rightarrow Z + W$ consists of "fuel" complexes present in excess, that hold Z and W sequestered. A cascaded reaction of the fuel complex with X and Y results in the release of active Z and W. Leaks in this case consist of Z and W becoming spuriously activated, even in the absence of active X or Y.

Importantly, for a catalytic reaction such as $X + Y \rightarrow X + Z$, it is possible to design a strand displacement implementation that does not leak the catalyst X. This implementation would release the same exact molecule of X as was consumed to initiate the process, as in the catalytic system described in [5]. Since fuels do not hold X sequestered, X cannot be produced in the leak process (although Z can).

Modeling Reactions and Leaks. Note that in all cases above, we can guarantee that a species does not leak if it is exclusively a catalyst in every reaction it occurs in. This allows us some handle on the leak. In particular, we will ensure that the species we are trying to detect (called D below) will be a catalyst in

every reaction that involves it. Otherwise, there might be a leak pathway to generating D itself—which is fundamentally irrecoverable.

We express the implementations below in the *population protocol* formalism [6]. That is, we consider a system with n molecules (aka nodes), which interact uniformly at random in a series of discrete steps. A population protocol is given as a set of reactions (aka transition rules) of the form

$$A + B \rightarrow C + D.$$
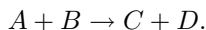
Note that unlike general reaction networks, population protocols conserve total molecular count since molecules never combine or split. For this reason, compared to general chemical reaction networks, this model is easier to analyze.

Given the set of reactions defining a protocol, we partition the species into *catalytic* states, which never change count as a consequence of any reaction, and *non-catalytic*, otherwise. Crucially, we model *leaks* as spurious reactions which can consume and create arbitrary non-catalytic species. More formally, a leak is a reaction of the type

$$S \rightarrow S',$$

where S and S' denote arbitrary non-catalytic species. In the following, we do not make any assumptions on the way in which these leak transitions are chosen (i.e., they could in theory be chosen *adversarially*), but we assume an upper bound on the *rate* at which leaks may occur in the system, controlled by a parameter β.

Leak-Robust Detection. A computationally simple task which already illustrates the difficulty of information processing in such an error-prone system is *single molecule detection*. Consider a solution of n molecules, in which a single molecule D may or may not be present. Intuitively, the goal is to generate large-scale (in the order of n) change in the system, depending on whether or not D is present or absent. Our time complexity measure is *parallel time*, defined as the number of pairwise interactions, divided by n. This measure of time naturally captures the parallelism of the system, where each molecule can participate in a constant number of interactions per unit time. Subject to leaks, our goal is to design the chemical interaction rules (formalized as a population protocol) to satisfy the following behavior. If D is present then it is detected fast, in logarithmic parallel time, and that the output is probabilistically "stable" in the sense that sampled at a random future time the system is in the "detected configuration" with high probability. By contrast, if D is absent, then the system sampled at a random future time should be in the "undetected configuration" with high probability. This basic task has several variations, for instance signal amplification or approximate counting of D.

We first develop some intuition about this problem, by considering some strawman approaches.

A first trivial attempt would be to have neutral molecules become "detectors" (state T) as soon as they encounter D, that is,

$$D + N \rightarrow D + T.$$
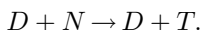
This approach suffers from two fatal flaws. First, it is *slow*, in that detection takes *linear* parallel time. Second, it has no way from recovering from leaks of the type $N \to T$.

A second attempt could try to implement an epidemic-style detection of D, that is:

$$D + N \to D + T$$

$$T + N \to T + T.$$
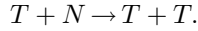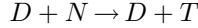
This approach is *fast*, i.e. converges in *logarithmic* parallel time in case D is present. However, if D is not present, the algorithm converges to a *false positive* state: a leak of the type $N \to T$ brings the system to an all-T state, despite the absence of D. One could try to add a "neutralization" pathway by having T turn back to N after a constant number of interactions, but a careful analysis shows that this approach also fails to recover from leaks of the type $N \to T$.

Thus, it is not clear whether leak-resistant detection is possible in population protocols (or more generally chemical reaction networks). There has been considerable work in the algorithmic community on diffusion based models, e.g. [7]. However, such results do not seem to apply to this setting, since leak models have not been considered previously, and none of the known techniques are robust to leaks. In particular, it appears that techniques for *deterministic* computation in population protocols do not carry over in the presence of leaks. More generally, this seems to create an unfortunate gap between the algorithmic community, which designs and analyzes population protocols in leak-free models, and more practically-minded research, which needs to address such implementation issues.

Contribution. In this paper, we take a step towards bridging this gap. We provide a general algorithmic model of leaks, and apply it to the detection problem. Specifically, our immediate goal is to elucidate the question of whether efficient, leak-robust detection is possible.

We prove that the answer is yes. We present a new algorithm, called *Robust-Detect*, which guarantees the following. Assume that the rate at which leaks occur is upper bounded by $\beta/n \ll 1/n$, and that we return the output mapping (detect/non-detect) of a randomly chosen molecule after $O(\log n)$ parallel time. Then the probability of a *false negative* is at most $1/e + o(1)$, and the probability of a *false positive* is at most β. (Note that as the total molecular count n increases, the chance that a particular interaction involves D decreases linearly with n. Thus the leak rate must also decrease linearly with n, or else the leaks will dominate. Alternatively, we can view some fixed leak rate as establishing an upper bound on the molecular count n, see below.)

Algorithm Description. We now sketch the intuition behind the algorithm and its properties, leaving the formal treatment to Sects. 4 and 5. Fix a parameter $s \geq 1$, to be defined later. We define a set of "detecting" species $X_1, \ldots, X_s$, arranged in consecutive levels. Whenever a molecule meets D, it moves to the highest "alert" level, X_1. Since leaks might produce this species as well, we decay it gracefully across s levels. More precisely, whenever a molecule at level X_i meets another molecule at level X_j, both molecules move to state $X_{\min(i,j)+1}$.

A molecule which would move beyond level X_s following a reaction becomes neutral, i.e. moves to species N. Nodes in state X_i with $i < s$ turn N into X_{i+1}, whereas molecules in state X_s also become neutral when interacting with N.

Analysis. Intuitively, the algorithm's dynamics for the case where a single molecule is in state D are as follows. The counts of molecules in state X_i tend to increase *exponentially* with the alert level i, up to levels $\approx \log n$, when the count becomes a constant fraction of n. However, once level $\log n$ is reached, these counts decrease *doubly exponentially*. Thus, it suffices to set $s = \log n$ to obtain that a fraction of at least $(1 - 1/e)$ molecules are in one of the alert states X_i in case D is present. It is not hard to prove that leaks cannot meaningfully affect the convergence behavior in this case.

The other interesting case is when D is not present, but leaks may occur, leading to possible false positives. Intuitively, we can model this case as one where states X_1 at the highest alert level simply are created at a lower rate $\beta/n \ll 1/n$. A careful analysis of this setting yields that the probability of a *false positive* (D detected, but not present) in this case is at most β, corresponding to the leak rate parameter.

Our analysis technique works by characterizing the stationary behavior of the Markov chain corresponding to the algorithm, and the convergence properties (mixing time) of this chain. For technical reasons, the analysis uses a non-standard hybrid argument, combining ideas from discrete analysis of population protocols with classic Markov chain techniques. The argument proves that the algorithm always stabilizes to the correct output in logarithmic parallel time.

The analysis further highlights a few interesting properties of the algorithm. First, if the detectable species D is present in a higher count $k > 1$, then the algorithm effectively skips the first $\log k$ levels, and thus requires $\log(n/k) + O(\log \log n)$ states. Second, it is not necessary to know the exact value of $\log n$, as the counts of species past this threshold decrease doubly exponentially.

Alternative Formulations. An alternative view of this protocol is as solving the following related *amplification* problem: we are given a signal of strength (rate) ϕ, and the algorithm's behavior should reflect whether this strength is below or above some threshold. The detection problem requires us to differentiate thresholds set at β/n and $1/n$, for constant $\beta \ll 1$, but our analysis applies to more general rates.

Above, we have assumed that the leak rate decreases linearly with n, to separate from the case where a single instance of D is present. However, it is also reasonable to consider that the leak rate is *fixed*, say, upper bounded by a constant λ. In this case, the analysis works as long as the number of molecules n satisfies $\lambda \ll 1/n$.

Self-stabilization. Our algorithm is self-stabilizing in the sense that if the count of D changes due to some external reason, the output quickly adapts (within logarithmic parallel time). This is particularly interesting if the algorithm is used in the context of a "control module" for a cell detecting D and the amount of

D changes over time. Note that strawman solutions considered above cannot be "untriggered" once D has been detected, and thus cannot adapt to a changing input.

2 Related Work

There is much work on attempting to decrease error in the underlying chemical substrate. A famous example includes kinetic proofreading [8]. In the context of DNA strand displacement systems in particular, leak reduction has been a prevailing topic [9]. Despite the importance of handling leaks, there are few examples of non-trivial *algorithms*, where leaks are handled through computation embedded in chemistry. One algorithm that appears to be able to handle errors is *approximate majority* [10], originally analyzed in a model where a fraction of the nodes are Byzantine, in that they can change their reported state in an adversarial way. Potentially due to its robustness properties, the approximate majority algorithm appears to be widely used in biological regulatory networks [11], and it was also one of the first chemical reaction network algorithms implemented with strand displacement cascades [4].

Our algorithm can be viewed as a timed, self-stabilizing version of rumor spreading. For analysis of simple rumor-spreading, see [12]. Other work include fault-tolerant rumor spreading [13], push-pull models [7] and self-stabilizing broadcasting [14]. A rumor-spreading formulation of the molecule detection problem is also considered in recent work [15], which relies on a different source amplification mechanism based on oscillator dynamics. This protocol [15] is self-stabilizing in a weaker (probabilistic) sense compared to the algorithms from this paper and does not provide leak robustness guarantees.

3 Preliminaries

3.1 Population Protocols with Leaks

Population Protocols. We start from a standard population protocol model, where n molecules (nodes) interact uniformly at random in a series of discrete steps. In our formulation, in each step, a coin is flipped to decide whether the current interaction is a regular reaction or a leak reaction. In the former case, two molecules are picked uniformly at random, and interact according to the rules of the protocol. In the latter case, a leak reaction occurs (see below).

A population protocol is given as a set of reactions (transition rules) of the form
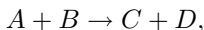
$$A + B \rightarrow C + D,$$

(where some of A, B, C, D might be the same). We (arbitrarily) match the first reactant (A) with the first product (C), and the second reactant (B) with the second product (D), and think of A as changing state to C, and B as changing state to D. If the two molecules picked to interact do not have a corresponding

interaction rule, then they don't change state and we call this a null interaction. Population protocols are a special case of the stochastic chemical reaction networks kinetic model (e.g., [16](A.4)).

Catalytic and Non-Catalytic Species. Given a set of reactions, we define the set of *catalytic* species as the set of states which never change as a consequence of any reaction. That is, for every reaction, the species is present in the same count both in the input and the output of the reaction. For example, in the reactions

$$A + C \rightarrow B + C$$
$$A + B \rightarrow A + D$$
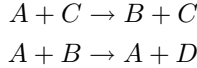
we call C *catalytic*. Note that A acts as a catalyst in the second reaction, but its count is changed by the first reaction, thus it is not overall catalytic. All species whose count is modified by some reaction are called non-catalytic. Note that it is possible that a species is never created, but disappears as a consequence of an interaction. For example, in the reaction

$$L + L \rightarrow A + B,$$
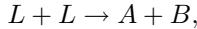
L is such as species. We define such species as *non-catalytic*, since their creation is possible by the law of reversibility, and thus they can leak.

An Algorithmic Model of Leaks. A leak is a reaction of the type

$$S \rightarrow S'$$

where S and S' are arbitrary non-catalytic species produced by the algorithm. Note that the input and output species of a leak may be the same (although in that case the reaction is trivial). In the following, we make no assumptions on the way in which the input and output of a leak reaction are chosen—we assume that they are chosen adversarially. Instead, we assume an absolute bound on the probability of a leak.

We assume that each reaction is either a *leak reaction* or a *normal reaction*, which follows the algorithm. We formalize this as follows.

Definition 1. *Given an algorithm, defined by a set of reactions, the set of catalysts is the set of species whose count does not change as a consequence of any reaction. A* leak *is a spurious reaction, which changes an arbitrary non-catalytic species to an arbitrary non-catalytic species. The* leak rate β/n *is the probability that any given interaction is a leak reaction.*
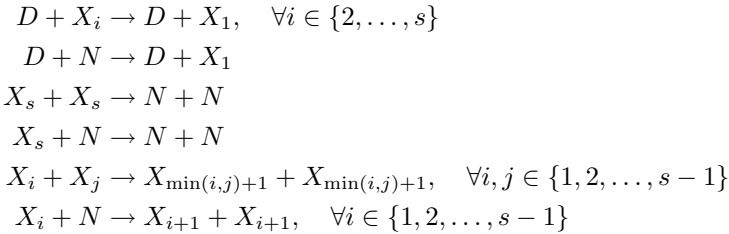
3.2   The Detection Problem

In the following, we consider the following *detection* task: we are given a distinct species D, whose presence or absence must be detected by the algorithm, in the presence of leaks. More precisely, if the species D is present, then the algorithm

should stabilize to a state in which molecules map to output value "detect". Otherwise, if D is not present, then the algorithm should stabilize to a state in which molecules map to output value "non-detect". To observe the algorithm's output, we sample a molecule at random, and return its output mapping. (Alternatively, to boost accuracy, we can take a number of samples, and return the majority output mapping.) We require that species D are catalytic.

4   The Robust-Detect Algorithm

Description. As given in the problem statement, we assume that there exists a distinguished species D, which is to be detected, and which never changes state. Our algorithm implements a chain of detection species $X_1, \ldots, X_s$, for some parameter s, each of which maps to output "detect", but with decreasing "confidence". Further, we have a neutral species N, which maps to output "non-detect". We assume that the parameter $s = \lceil \log n \rceil$, and that initially all molecules are in state N. We specify the transitions below, and provide the intuition behind them.

Algorithm 1

$$D + X_i \to D + X_1, \quad \forall i \in \{2, \ldots, s\}$$
$$D + N \to D + X_1$$
$$X_s + X_s \to N + N$$
$$X_s + N \to N + N$$
$$X_i + X_j \to X_{\min(i,j)+1} + X_{\min(i,j)+1}, \quad \forall i, j \in \{1, 2, \ldots, s-1\}$$
$$X_i + N \to X_{i+1} + X_{i+1}, \quad \forall i \in \{1, 2, \ldots, s-1\}$$

The intuition behind the algorithm is as follows. The "detecting" species $X_1, \ldots, X_s$ are arranged in consecutive levels. Whenever a molecule meets D, it moves to the highest "alert" level, X_1. Since leaks might produce this species as well, we decay it gracefully across s levels. After going through these levels, a molecule moves to neutral state N, in case it is not brought back either by meeting D, or some molecule at a lower alert level. For this, whenever two of these species X_i and X_j meet, they both move to level $\min(i, j)+1$. This reaction has the double purpose of both decaying the alert level of the molecule at the lower level, and of bringing back the molecule with the higher alert level. Further, whenever a molecule at level X_i meets a neutral molecule N, it advances its level by 1. At the same time, neutral molecules are turned into detector molecules whenever meeting some molecule at an alert level smaller than s.

Intuitive Dynamics. Roughly, the chain of alert levels have the property that, for the first $\sim \log n$ levels, the count roughly *doubles* with level index. At the same time, past this point, counts exhibit a steep (doubly exponential) drop, so

that a small constant fraction of molecules are always neutral. The presence of D acts like a trigger, which maintains the chain in "active" state. The analysis in the next section makes this intuition precise. These dynamics are illustrated in Fig. 1.
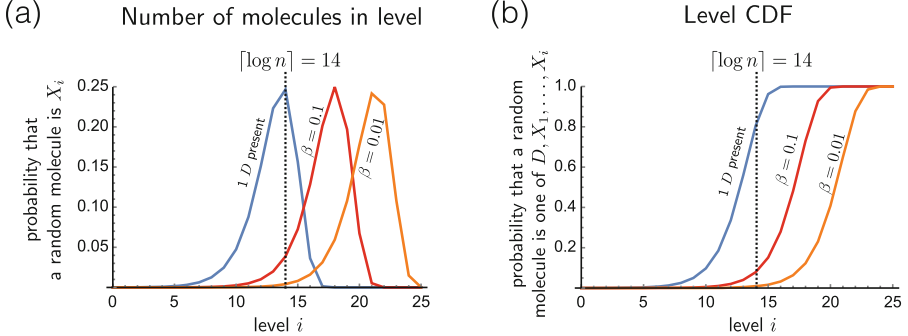

Fig. 1. Steady state probabilities of the Robust-Detect algorithm for $n = 10^4$ molecules. Three conditions are evaluated: (blue) 1 molecule of D is present and no leak (leak parameter $\beta = 0$); (orange, red) no D is present but with worst-case leak for false-positives (leak reactions $X_i \to X_1$ and $N \to X_1$) (orange: leak parameter $\beta = 0.01$, red: leak parameter $\beta = 0.1$). (a) The probabilities of each level i. (b) The cumulative probabilities of levels $\leq i$, capturing the probability that a random molecule is in a "detect" state. Note that it is enough to set the number of levels $s = 14 = \lceil \log n \rceil$ to have both false positive and false negative error probabilities small, although for smaller leak rates ($\beta = 0.01$) increasing s beyond $\log n$ can help better distinguish true and false positives. Numerical probabilities are computed using Eqs. (1) and (2).

5 Analysis

Overview. We divide the analysis of the detection algorithm into two parts. First, we derive stationary probabilities of the underlying Markov chain of transitions of particles, by solving recursively the equations following from the underlying dynamics. Later, we derive optimal bounds on the mixing time of this Markov chain—that is we show that probability distribution of states at every time $t \geq cn \log n$ (for some constant c) is almost the same as the stationary distribution.

Simplified Algorithm. For the purpose of analysis, let us consider a following rephrasing of the detection algorithm: molecule states are $D, X_1, X_2, X_3, \ldots,$ and interactions are as follows:

Algorithm 2

$$D + X_i \to D + X_1,$$
$$X_i + X_j \to X_{\min(i,j)+1} + X_{\min(i,j)+1}.$$
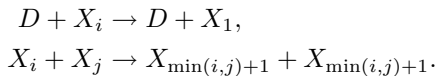
This algorithm uses infinite number of states, thus it is useful only for purposes of theoretical analysis. However, it captures the behavior of the original algorithm in the following way: if in Algorithm 2 all states $X_{s+1}, X_{s+2}, \ldots$ are collapsed to N, the transitions are equivalent to Algorithm 1. However, formulation of Algorithm 2 is oblivious to parameter s, thus captures simultaneously the dynamics of all possible instances of Algorithm 1.

5.1   Stationary Analysis

Let us consider an initial state when $k \geq 0$ instances of state D are present, with special attention given to $k = 0$ and $k = 1$. Those molecules do not change their state.

We can imagine tracking a particular molecule through its state transitions, such that its state can be expressed as a Markov chain. In the following, we will focus on analyzing the stationary distribution of this Markov chain.

For any $i \in \{1, 2, \ldots, s\}$, we let $p_i^\star$ be the stationary probability that a molecule chosen uniformly at random is in state X_i. We let $p_0^\star = \frac{k}{n}$ be the (stationary) probability that the molecule is in the state D. Let $p_{\leq i}^\star = p_0^\star + \ldots + p_i^\star$ be the probability that a molecule is in any of the states $D, X_1, \ldots, X_i$.

Let us now analyze these stationary probabilities.

Stable State with No Leaks. We first analyze the simplified case where no leaks occur. A molecule u is in one of states $D, X_1, \ldots, X_i$ at time t, in two cases:

- It was in state $D, X_1, \ldots, X_i$ at time $t - 1$, and did not get selected for a reaction, which occurs with probability $1 - 2/n$.
- It got selected for a reaction with element u', and either u or u' was in one of states $D, X_1, \ldots, X_{i-1}$.

Hence, by stationarity, we get that

$$p_{\leq i}^\star = p_{\leq i}^\star \left(1 - \frac{2}{n}\right) + \frac{2}{n} \cdot (1 - (1 - p_{\leq(i-1)}^\star)^2).$$

From this we get that

$$1 - p_{\leq i}^\star = (1 - p_{\leq(i-1)}^\star)^2,$$

which solves to

$$p_{\leq i}^\star = 1 - \left(1 - \frac{k}{n}\right)^{2^i}. \tag{1}$$

This gives us following estimates: if $k \geq 1$, then $p_i^\star \approx 2^{i-1}\frac{k}{n}$ for $i \leq \log(n/k)$. Additionally, for $i = \log(n/k) + 1 + j$, $p_i^\star \approx e^{-2^j}$. Thus, for $i \geq \log(n/k) + \Theta(\log \log n)$ in all practicalities $p_i^\star \approx 0$.

To analyze the probability of detection when $k = 1$, we sum probabilities for all i from 0 to $s = \lceil \log n \rceil$

$$\Pr[\text{detect}] = \sum_{i=0}^{s} p_i^{\star} = p_{\leq s}^{\star} \geq 1 - \left(1 - \frac{1}{n}\right)^n \geq 1 - \frac{1}{e}.$$

Probability of False Positives with Leaks. A useful side effect of the previous analysis is that we also get probability bounds for detection in the case where D is *not* present, i.e. false positives. We model this case as follows. Assume that there exists an upper bound λ on the probability that a certain reaction is a leak. Examining the structure of the algorithm, we note that the worst-case adversarial application of leaks would be if this probability is entirely concentrated into leaks which produce species X_1.

To preserve molecular count, we assume the following simplified leak model, which is equivalent to the general one, but easier to deal with in the confines of our algorithm.

Each reaction is a *leak* with probability $\lambda = \beta/n$, where $\beta \ll 1$ is a small constant. If a reaction is a leak, it selects a molecule at random, and transforms it into an arbitrary state. In this case, we will assume adversarially that all leaked molecules are transformed into state X_1. Notice that the assumption that $\beta \ll 1$ is required to separate this setting from the case where D is present in the system, where the probability of producing state X_1 is $2/n$.

We continue with calculations of $p_0^{\star}, p_1^{\star}, \ldots$ for the above formulation. Note that the recurrence relation for $p_{\leq i}^{\star}, i \geq 1$ is changed as follows:

- If at that round there was no leak, the transition probabilities are as previously. This happens with probability $1 - \frac{\beta}{n}$.
- If there was a leak, then the molecule either is selected as a leaked molecule (this happens with probability $\frac{1}{n} \cdot \frac{\beta}{n}$) or it was not selected as a leaked molecule, but it was already in the proper state (probability $\frac{\beta}{n} \cdot \frac{n-1}{n} p_{\leq i}^{\star}$).

The recursive formulation gives

$$p_{\leq i}^{\star} = \left(p_{\leq i}^{\star}\left(1 - \frac{2}{n}\right) + \frac{2}{n}\left(1 - (1 - p_{\leq(i-1)}^{\star})^2\right)\right)\left(1 - \frac{\beta}{n}\right) + \left(\frac{1}{n} + \frac{n-1}{n}p_{\leq i}^{\star}\right)\frac{\beta}{n}.$$

Which is equivalent to

$$1 - p_{\leq i}^{\star} = \frac{\left(1 - \frac{\beta}{n}\right)}{\left(1 - \frac{\beta}{2n}\right)}(1 - p_{\leq(i-1)}^{\star})^2$$

leading to (using estimate $(1 - \frac{\beta}{n})/(1 - \frac{\beta}{2n}) \approx (1 - \frac{\beta}{2n})$)

$$p_{\leq i}^{\star} \approx 1 - \left(1 - \frac{\beta}{2n}\right)^{1+2+\ldots+2^{i-1}} = 1 - \left(1 - \frac{\beta}{2n}\right)^{2^i - 1}. \tag{2}$$

This gives us following estimates: $p_i^\star \approx 2^{i-2}\frac{\beta}{n}$ for $i \leq \log(2n/\beta)$. Additionally, for $i = \log(2n/\beta) + 1 + j$, $p_i^\star \approx e^{-2^j}$. Thus, for $i \geq \log(2n/\beta) + \Theta(\log\log n)$ in all practicalities $p_i^\star \approx 0$.

This immediately implies that

$$\Pr[\text{detect}] = \sum_{i=0}^{s} p_i^\star = p_{\leq s}^\star \leq 1 - \left(1 - \frac{\beta}{2n}\right)^{2n} = 1 - \frac{1}{e^\beta} \approx \beta,$$

which means that the probability that a randomly chosen molecule is in detect state when chosen uniformly at random is at most β.

Probability of False Negatives with Leaks. Under the same leak model, it is easy to notice that the "best" adversarial strategy for our algorithm in case D is present is to concentrate all leaks to create the neutral species N (or X_∞ in case of Algorithm 2). It is easy to see that this just decreases the total probability of detect states by the leak probability $\lambda = \beta/n$. More formally, we compute once again stationary probabilities. The recurrent relation is

$$p_{\leq i}^\star = \left(p_{\leq i}^\star\left(1 - \frac{2}{n}\right) + \frac{2}{n} \cdot (1 - (1 - p_{\leq (i-1)}^\star)^2)\right)\left(1 - \frac{\beta}{n}\right) + \frac{\beta}{n} \cdot \frac{n-1}{n}p_{\leq i}^\star.$$

Using estimate $(1 - \frac{\beta}{n})/(1 - \frac{\beta}{2n}) \approx (1 - \frac{\beta}{2n})$ we reach

$$p_{\leq i}^\star = \left(1 - \frac{\beta}{2n}\right)(1 - (1 - p_{\leq (i-1)}^\star)^2).$$

Thus we have for the first $\log(n/k)$ levels the dampening factor of $(1 - \beta/(2n))$ per level (compared to the leakless case). It can be easily shown by induction that

$$\left(1 - \frac{\beta}{2n}\right)^i \left(1 - \frac{k}{n}\right)^{2^i} \leq p_{\leq i}^\star \leq \left(1 - \frac{k}{n}\right)^{2^i}.$$

The estimates for $p_i^\star$ follow from the leakless case, after taking into the account the composed dampening factor:

$$\Pr[\text{detect}] = \sum_{i=0}^{s} p_i^\star = p_{\leq s}^\star \geq \left(1 - \frac{1}{e}\right) \cdot \left(1 - \frac{\beta}{2n}\right)^{\log n} = 1 - \frac{1}{e} - \mathcal{O}\left(\frac{\log n}{n}\beta\right).$$

Finally, we summarize the results in this section as follows:

Theorem 1. *Assuming leak rate β/n for $\beta \ll 1$, Robust-Detect guarantees the following.*

- *The probability of a false positive is at most β.*
- *The probability of a false negative is at most $1/e + \mathcal{O}(\beta \cdot (\log n)/n)$.*

Notice that these probabilities can be boosted by standard sampling techniques.

5.2   Convergence Analysis

We now proceed with an analysis of the convergence speed of the previously described protocols. To avoid separate analysis for each of the aforementioned cases (no leaks, false positives, false negatives) and to be independent from all possible initializations of the algorithm, we first start with showing that, under no leaks and with no D present, all states $X_1, \ldots, X_c$ are quickly killed.

In this section, it is more convenient to use t to refer to the total number of interactions, rather than parallel time. To convert to parallel time, one needs to divide by n, the number of molecules.

Lemma 1. *Assume arbitrary (adversarial) initial state in $t = 0$ and evolution with no leaks ($\beta = 0$) and no D is present. For any $c(n) \geq 1$, there is $t = \mathcal{O}(n \cdot (c(n) + \log n))$ such that with probability $1 - 1/n^{\Theta(1)}$ (with high probability) there is no molecule in any of the states $X_1, X_2, \ldots, X_{c(n)}$ after t interactions.*

Proof. We assign a potential to each molecule, based on the state it is currently in: $\Phi(X_i) = 3^{-i}$. We also define a global potential Φ_t as sum of all molecular potentials after t interactions. Observe, that when two molecules interact, following rule $X_i + X_j \rightarrow X_{\min(i,j)+1} + X_{\min(i,j)+1}$, then there is:

$$\Phi(X_{\min(i,j)+1}) + \Phi(X_{\min(i,j)+1}) \leq 2/3 \cdot (\Phi(X_i) + \Phi(X_j)),$$

which can be interpreted that each interacting molecule loses at least $1/3$ of its potential. Since each molecule participates in an interaction with probability $\frac{2}{n}$ in each round, the following bound holds:

$$\mathbf{E}[\Phi_{t+1} - \Phi_t | \Phi_t] \geq \cdot \sum_v \Pr(v \text{ interacts in round } t) \cdot \frac{1}{3}\Phi_t(v) = \frac{2}{3n}\Phi_t,$$

$$\mathbf{E}[\Phi_{t+1} | \Phi_t] \leq \left(1 - \frac{2}{3n}\right)\Phi_t.$$

Substituting $\Phi_0 \leq n$ and fixing $t \geq \frac{3}{2}n \ln(n \cdot 3^{c(n)} \cdot n^{\Theta(1)}) = \mathcal{O}(n(c(n) + \log n + \Theta(\log n)))$ we have

$$\mathbf{E}[\Phi_t] \leq \left(1 - \frac{2}{3n}\right)^t \cdot n \leq e^{-\ln(n \cdot 3^{c(n)} \cdot n^{\Theta(1)})} \cdot n = 3^{-c(n)} \cdot \frac{1}{n^{\Theta(1)}}.$$

By Markov's inequality, this means that there is no molecule in any of the states $X_1, X_2, \ldots, X_c$ with probability at least $1 - n^{-\Theta(1)}$, that is with high probability.   □

We mention one additional useful property of Algorithm 2, that its actions on population are decomposable with respect to levels. That is, define $\text{LEVEL}_t(u) = i$ if molecule u at time t is in state X_i, and $\text{LEVEL}_t(u) = 0$ if it is in state D.

Observation 1. *Let* $\{u_1, u_2, \ldots, u_n\}, \{v_1, v_2, \ldots, v_n\}, \{w_1, w_2, \ldots, w_n\}$ *be 3 disjoint populations each on* n *molecules, following evolution defined by Algorithm 2. Moreover, let their evolutions be coupled: at each time* t, *in each population the corresponding molecules interact (i.e., the interaction is* $u_i + u_j$, $v_i + v_j$, $w_i + w_j$ *in the three populations for some* i, j).*

If $\forall_i \text{LEVEL}_0(u_i) = \min(\text{LEVEL}_0(v_i), \text{LEVEL}_0(w_i))$, *then at any time* $t > 0$: $\forall_i \text{LEVEL}_t(u_i) = \min(\text{LEVEL}_t(v_i), \text{LEVEL}_t(w_i))$.

This observation can be naturally generalized to more than 3 populations. As shown below, the observation implies that to analyze detection under noisy start, we can decouple starting noise from detected particle and analyze evolution under those two separately. Denote by $p_i(t)$ and $p_{\leq i}(t)$ the probability for a randomly picked molecule after t interactions to be in the state X_i or $D, X_1, \ldots, X_i$ respectively.

Theorem 2. *Fix arbitrary leak model (i.e. no leaks, false-positives, false-negatives) and arbitrary concentration of* D. *For any* $c \geq 1$, *and* $t = \Omega(n \cdot (c + \log n))$, *there is*

$$\left| p^\star_{\leq c} - p_{\leq c}(t) \right| \leq 1/n^{\Theta(1)},$$

where $p^\star$ *is the stationary probability distribution of the identical process.*

Proof. First, for simplicity we collapse all states $X_{c+1}, X_{c+2}, \ldots$ into N, since it has no effect on $p_{\leq c}$ distributions. Consider a population of size n, under no leaks, no D, evolution. By Lemma 1, in $\tau = \mathcal{O}(n \cdot (c + \log n))$ steps it reaches all-N state, regardless of initial configuration, with high probability. Thus evolution of any population $\{u_i\}$, under no leaks, with D present, is a coupling (as in Observation 1) of following evolutions:

– initial configuration of population $\{u_i\}$, with each D replaced with N;
– for every timestep t_i such that D interacted with X_i or N creating X_1, we couple a population with corresponding molecule set to X_1 and every other molecule set to N, shifted in time so its evolution starts at time t_i.

Observe, that evolution of population of both types will reach all-N state in τ steps, with high probability. Thus, conditioned on this high probability, the configuration at any $t \geq \tau$ is the result of coupling of all-N (result of evolution of first type population) with possibly several configurations of the second type, where at each timestep $t' \in [t - \tau, t]$ such population was created independently with some probability only depending on n and k. However, the coupling we just described is invariant from the choice of t, as long as $t \geq \tau$. Thus, for any $t_1, t_2 \geq \tau$, there is $|p_{\leq c}(t_1) - p_{\leq c}(t_2)| \leq 1/n^{\Theta(1)}$. Since $p^\star_{\leq c} = \lim_{t \to \infty} \frac{1}{t} \sum_{i=1}^{t} p_{\leq c}(i)$, the claimed bound follows.

To take into account errors, we say that whenever there is a leak changing state of molecule v to some state S at time t, we change state of v at that time in all existing populations to N, and create new population where v has state S, and all other molecules are in N state. The same reasoning as in the error-less case follows, since switching molecules to N state it only speeds up convergence

of populations to all-N state and since populations created due to leaks are created at each step with the same probability depending only on n and error model.   □

6   Simulation Results

We simulated the Robust-Detect algorithm (Algorithm 1) using a modified version of the CRNSimulatorSSA Mathematica package [17]. Figure 2 shows the shape of typical trajectories when there is one molecule in state D ($k = 1$), compared with no molecules in state D ($k = 0$) but with the worst-case leak for false-positives. Note that D is quickly detected if present, and if absent the system exhibits random perturbations that are quickly extinguished and are clearly distinguishable from the true positive case.
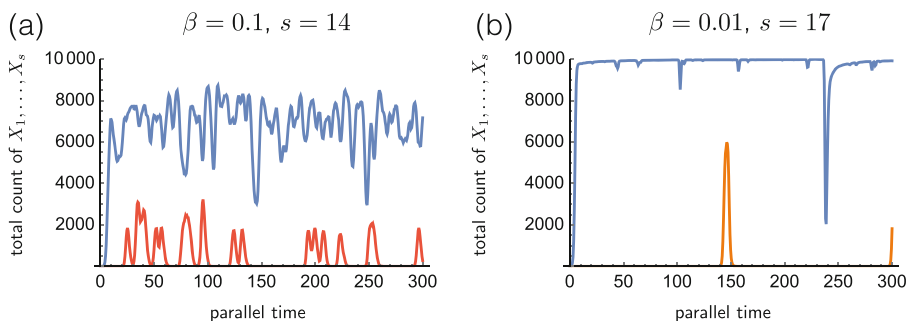

Fig. 2. Typical time-evolution of the Robust-Detect algorithm (Algorithm 1). Three colors correspond to the three conditions described in Fig. 1: (blue) 1 molecule of D is present and no leak (leak parameter $\beta = 0$); (orange, red) no D is present but with worst-case false-positive leak $X_i \to X_1$ and $N \to X_1$ (orange: leak parameter $\beta = 0.01$, red: leak parameter $\beta = 0.1$). All X_i states map to output value "detect", and thus we plot the sum of all their counts. (a) $s = 14$ layers, $\beta = 0.1$ (red). (b) $s = 17$ layers, $\beta = 0.01$ (orange). See Fig. 1 for the corresponding steady state probabilities. Note that with smaller leak ($\beta = 0.01$), it is possible to better distinguish true positives and false positives by increasing the number of layers (from 14 to 17). In all cases there are $n = 10^4$ molecules, and the initial configuration is all molecules in neutral state N. Parallel time (number of interactions divided by n) corresponds to the natural model of time where each molecule can interact with a constant number of other molecules per unit time. (Color figure online)

7   Conclusions

We have considered the problem of modeling and withstanding leaks in chemical reaction networks, expressed as population protocols. We have presented an arguably simple algorithm which is probabilistically correct under assumptions on the leak rate, and converges quickly to the correct answer.

Beyond the specific example of robust detection, we hope that our results motivate more systematic modeling of leaks, and further work on algorithmic techniques to withstand them. As such errors appear to spring from the basic laws of chemistry, their explicit treatment appears to be necessary. The authors found it surprising that many of the algorithmic techniques developed in the context of deterministically correct population protocols might not carry over to implementations, due to their inherent non-robustness to leaks.

In future work, we plan to perform an exhaustive examination of which of the current algorithmic techniques could be rendered leak-robust, and whether known algorithms can be modified to withstand leaks via new techniques. Another interesting avenue for future work is *lower bounds* on the set of computability or complexity of fundamental predicates in the leak model. Finally, we would like to examine whether our robust detection algorithm can be implemented in strand displacement systems.

References

1. Gopalkrishnan, M.: Catalysis in reaction networks. Bull. Math. Biol. **73**(12), 2962–2982 (2011)
2. Soloveichik, D., Seelig, G., Winfree, E.: DNA as a universal substrate for chemical kinetics. Proc. Natl. Acad. Sci. **107**(12), 5393–5398 (2010)
3. Cardelli, L.: Two-domain DNA strand displacement. Math. Struct. Comput. Sci. **23**(02), 247–271 (2013)
4. Chen, Y.-J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., Seelig, G.: Programmable chemical controllers made from DNA. Nat. Nanotechnol. **8**(10), 755–762 (2013)
5. Zhang, D.Y., Turberfield, A.J., Yurke, B., Winfree, E.: Engineering entropy-driven reactions and networks catalyzed by DNA. Science **318**(5853), 1121–1125 (2007)
6. Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M., Peralta, R.: Computation in networks of passively mobile finite-state sensors. Distrib. Comput. **18**, 235–253 (2006). Preliminary version appeared in PODC 2004
7. Karp, R.M., Schindelhauer, C., Shenker, S., Vöcking, B.: Randomized rumor spreading. In: 41st Annual Symposium on Foundations of Computer Science, FOCS 2000, pp. 565–574, IEEE Computer Society (2000)
8. Hopfield, J.J.: Kinetic proofreading: a new mechanism for reducing errors in biosynthetic processes requiring high specificity. Proc. Natl. Acad. Sci. **71**(10), 4135–4139 (1974)
9. Thachuk, C., Winfree, E., Soloveichik, D.: Leakless DNA strand displacement systems. In: Phillips, A., Yin, P. (eds.) DNA 2015. LNCS, vol. 9211, pp. 133–153. Springer, Cham (2015)
10. Angluin, D., Aspnes, J., Eisenstat, D.: A simple population protocol for fast robust approximate majority. Distrib. Comput. **21**(2), 87–102 (2008)
11. Cardelli, L.: Morphisms of reaction networks that couple structure to function. BMC Syst. Biol. **8**(1), 84 (2014)
12. Pittel, B.: On spreading a rumor. SIAM J. Appl. Math. **47**(1), 213–223 (1987)

13. Doerr, B., Doerr, C., Moran, S., Moran, S.: Simple and optimal randomized fault-tolerant rumor spreading. Distrib. Comput. **29**(2), 89–104 (2016)
14. Boczkowski, L., Korman, A., Natale, E.: Minimizing message size in stochastic communication patterns: fast self-stabilizing protocols with 3 bits. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, pp. 2540–2559
15. Dudek, B., Kosowski, A.: Spreading a confirmed rumor: A case for oscillatory dynamics, CoRR, vol. abs/1705.09798 (2017)
16. Soloveichik, D.: Robust stochastic chemical reaction networks and bounded tau-leaping. J. Comput. Biol. **16**(3), 501–522 (2009)
17. http://users.ece.utexas.edu/~soloveichik/crnsimulator.html