Synthesizing and tuning chemical reaction networks with specified behaviours

Neil Dalchau, Niall Murphy, Rasmus Petersen, Boyan Yordanov

Microsoft Research, Cambridge, CB1 2FB, UK,
{ndalchau,a-nimurp,a-rapete,yordanov}@microsoft.com

Abstract. We consider how to generate chemical reaction networks (CRNs) from functional specifications. We propose a two-stage approach that combines synthesis by satisfiability modulo theories and Markov chain Monte Carlo based optimisation. First, we identify candidate CRNs that have the *possibility* to produce correct computations for a given finite set of inputs. We then optimise the reaction rates of each CRN using a combination of stochastic search techniques applied to the chemical master equation, simultaneously improving the *probability* of correct behaviour and ruling out spurious solutions. In addition, we use techniques from continuous time Markov chain theory to study the expected termination time for each CRN. We illustrate our approach by identifying CRNs for majority decision-making and division computation, which includes the identification of both known and unknown networks.

Keywords: Chemical Reaction Networks, Program Synthesis, Parameter optimisation, Chemical Master Equation, Satisfiability Modulo Theories, Markov chain Monte Carlo

1   Introduction

A central goal of molecular programming is to be able to implement arbitrary dynamical behaviours. Chemical reaction networks (CRNs) are a popular formalism for describing biochemical systems, such as protein interaction networks, gene regulatory networks, synthetic logic circuits and molecular programs built from DNA. Extensive theoretical understanding exists about the behaviour of a multitude of CRNs, and the behaviour of some networks has been exhaustively explored [1]. Besides describing chemical systems, CRNs provide a common language for expressing problems studied in computer science theory (e.g. Petri nets, network protocols) as well as control theory and engineering. Methods exist to convert CRNs into equivalent physical implementations, based on DNA strand displacement [2,3] the DNA toolbox system [4] and genelets [5]. Therefore, we sought to develop a methodology for proposing candidate CRNs that exhibit a pre-specified behaviour.

The computational power of CRNs has been extensively studied [6]. It is known that error-free (stably computing [7]) CRNs compute exactly the class of semi-linear functions [8,9]. However, if the stability restriction is relaxed and we

allow the CRN to sometimes compute the wrong answer then it is possible to implement a register machine, that is, CRNs with error can compute functions beyond the semi-linear class (indeed they are equivalent in power to Turing machines) [10,6].

Although there are procedures to generate CRNs for semi-linear functions [8,10], primitive recursive functions [6], or even from arbitrary Turing machines [6], the proposal of practical (i.e. experimentally implementable) CRNs that compute a given function has thus far mostly been a manual effort. In this work, we attempt to automate the proposal of CRNs, by formally specifying a behaviour and automatically identifying CRNs that satisfy the desired behaviour with high probability. First, we formalise the problem of identifying CRNs that have the capacity to produce correct, finite computations for a given finite set of inputs. This corresponds to a synthesis problem, as opposed to verification, where the goal is to determine the correctness of a given CRN [11]. We express CRN synthesis as a satisfiability modulo theories (SMT) problem, which can be addressed using solvers such as Z3 [12]. This allows us to generate a number of canditate CRNs or to prove that no such CRN of a given size (in terms of numbers of reactions, species and computation lengths) exists. However, while the existence of correct computations is guaranteed for each generated CRN, the probability of these computations might be low.

To determine whether correct computations can occur with high probability, we next optimise the reaction rates of each generated CRN. To solve the optimisation problem, we combine stochastic search strategies based on Markov chain Monte Carlo (MCMC) with numerical integration of the chemical master equation (CME). This part of the problem was recently addressed in [13,14], though applied only to a single input.

In this paper, we specifically focus on uniform CRNs, those that have a fixed number of species and reactions for all input sizes. We also restrict our attention to bimolecular CRNs, where there are precisely 2 reactants and 2 products in every reaction. Bimolecular CRNs are equivalent to Population Protocols (PPs) [7] and also guarantee that mass is conserved in the system. We applied our two-step approach first to majority decision-making, in which the network seeks to identify which of two inputs is in an initial majority. Majority networks are well-studied in the literature, and there are many known CRNs that give approximate solutions [15,16,17]. We then applied our approach to division, a non-linear function which has been relatively less studied. We show a range of CRNs for majority and division identified automatically using our method, some of which have been identified and characterised previously, though some of which are entirely novel. This illustrates the potential for automatically determining CRNs with a specified behaviour.

2 Preliminaries

A chemical reaction network (CRN) is a tuple $\mathcal{C} = (\Lambda, \mathcal{R})$, where $\Lambda = \{s_0, \ldots, s_n\}$ and $\mathcal{R} = \{r_0, \ldots, r_m\}$ denote the finite sets of species and reactions, respectively.

A reaction is a tuple $r = (\mathbf{r}^r, \mathbf{p}^r, k^r)$ where $\mathbf{r}^r$ and $\mathbf{p}^r$ are the reactant and product *stoichiometry* vectors ($\mathbf{r}^r_s \in \mathbb{N}_0$ and $\mathbf{p}^r_s \in \mathbb{N}_0$ denote the stoichiometry of each species $s \in \Lambda$), $k^r \in \mathbb{R}_{\geq 0}$ denotes the rate of r and $\mathbf{k}$ denotes the vector of all reaction rates. Given a reaction $r = (\mathbf{r}^r, \mathbf{p}^r, k^r)$, the set of reactants of r is $\{s \in \Lambda \mid \mathbf{r}^r_s > 0\}$ and the set of products of r is $\{s \in \Lambda \mid \mathbf{p}^r_s > 0\}$. In this paper, we focus on the class of *bimolecular* CRNs, where $\sum_{s \in \Lambda} \mathbf{r}^r_s = 2$ and $\sum_{s \in \Lambda} \mathbf{p}^r_s = 2$, for all reactions $r \in \mathcal{R}$.

The dynamical behaviour of bimolecular CRNs can be understood as follows. The set of all possible system states is $X = \mathbb{N}_0^{|\Lambda|}$, where a state $x \in \mathbb{N}_0^{|\Lambda|}$ represents the number of molecules of each species. We denote the number of molecules of species $s \in \Lambda$ at state x by x_s. Given a reaction $r \in \mathcal{R}$ where $\mathbf{r}^r_s = 2$ for some $s \in \Lambda$, the *propensity*[1] of r at x is $k^r_x = k^r \cdot \frac{x_s \cdot (x_s - 1)}{2}$. If, on the other hand, $\mathbf{r}^r_s = \mathbf{r}^r_{s'} = 1$ for some species s, s', the propensity of r is $k^r_x = k^r \cdot x_s \cdot x_{s'}$. The time at which reaction r would fire, once the system enters state $x \in X$, is stochastic and follows an exponential distribution with a rate determined by the reaction's propensity k^r_x. Assuming that reaction r is the first one to fire, the state of the system is updated as $x'_s = x_s - \mathbf{r}^r_s + \mathbf{p}^r_s$ for all $s \in \Lambda$, where x and x' are the current and next states.

An abstraction of CRNs that preserves reachability but does not consider reaction rates or time is given by the *transition system* $\mathcal{T}^{\mathcal{C}} = (X, T)$, where the transition relation T is defined as

$$\forall x, x' \in X \, . \, T(x, x') \leftrightarrow \bigvee_{r \in \mathcal{R}} \bigwedge_{s \in \Lambda} \left(x_s \geq \mathbf{r}^r_s \wedge x'_s = x_s - \mathbf{r}^r_s + \mathbf{p}^r_s \right). \tag{1}$$

In other words, the choice between reactions from $\mathcal{R}$ is non-deterministic but enough molecules of each reactant must be present in state x for the reaction to fire. The transition between states x and x' happens when any reaction $r \in \mathcal{R}$ fires and the number of molecules is updated accordingly. A path $x_0, x_1, \ldots$ of $\mathcal{T}$ satisfies $T(x_i, x_{i+1})$ for $i = 0, 1, \ldots$ and, given an initial state x_0 we call state x_f reachable from x_0 if there exists a path $x_0, \ldots, x_f$.

Given a CRN $\mathcal{C}$, let $X_0 \subseteq X$ denote a finite set of initial states and $X_r \subseteq X$ denote the set of states reachable from X_0. Assuming that X_r is finite, $\mathcal{C}$ can be represented as a *continuous time Markov chain* (CTMC) that preserves information about the transition probabilities and rates that determine the stochastic behaviour of the system and the expected execution times. We define a CTMC to be a tuple $\mathcal{M} = (X_r, \pi_0, Q)$, where X_r is a finite set of states, $\pi_0 : X_r \to \mathbb{R}$ is the initial distribution of molecule copy numbers of all species, and $Q : X_r \times X_r \to \mathbb{R}$ is a matrix of transition propensities. While the set of initial states is not represented explicitly, it is captured through the initial distribution, i.e. $X_0 = \{x \in X_r \mid \pi_0(x) > 0\}$. A CTMC $\mathcal{M}^{\mathcal{C}}$ is constructed from a CRN $\mathcal{C}$ by first determining the set of reachable states, and then evaluating the propensities of each reaction. The $(i, j)^{\text{th}}$ entry of Q, q_{ij}, represents a transition from state x_i to

---

[1] We assume that the reaction volume is 1 to allow for later volume scaling e.g. k^r_x / v is the propensity for a reaction volume equal to v

state x_j. Accordingly, q_{ii} is the remaining probability mass, equal to $-\sum_{i \neq j} q_{ij}$. The transient probability vector π_t evolves according to $\frac{d\pi_t}{dt} = \pi_t Q$, which is known as the chemical master equation (CME).

Following [13,14], a parametric CTMC (pCTMC) is a CTMC where the reaction rates are parameterised by $\mathbf{k}$, as above. Denote by $\mathcal{P}$ the parameter space, $\mathcal{P} : \mathbb{R}^P_{\geq 0}$, such that $\mathbf{k}$ is instantiated by a parameter point $p \in \mathcal{P}$. Accordingly, given a pCTMC $\mathcal{M}$ and parameter space $\mathcal{P}$, an instantiated pCTMC $\mathcal{M}_p = (X, \pi_0, Q_p)$ is an evaluation at point $p \in \mathcal{P}$.

3   Problem formulation

The main problem we consider in this paper, which we formalise in this section, is the identification of CRNs that satisfy given properties. Specifically, we are interested in finite reachability properties, which capture a range of interesting CRN behaviours.

Let $\mathcal{C} = (\Lambda, \mathcal{R})$ be a given CRN and $\mathcal{T}^{\mathcal{C}} = (X, T)$ and $\mathcal{M}^{\mathcal{C}} = (X_r, \pi_0, Q)$ denote its transition system abstraction and CTMC representation, as discussed in Section 2. Let $\phi : X \to \mathbb{B}$ denote a *state predicate*, constructed using

$$\phi ::= E_b$$
$$E_b ::= true \mid false \mid E_c \mid \neg E_b \mid E_b \rhd E_b \text{ where } \rhd \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$$
$$E_c ::= E_a \rhd E_a \text{ where } \rhd \in \{<, \leq, =, >, \geq\}$$
$$E_a ::= s \in \Lambda \mid c \in \mathbb{Z} \mid E_a \rhd E_a \text{ where } \rhd \in \{+, -, *\}.$$

For example, if $\phi := s > 5$, then $\phi(x)$ denotes that $x_s > 5$.

In this paper, we consider *path predicates* $\Phi = (\phi_0, \phi_F)$, which are expressed using two state predicates that must be satisfied at the initial (ϕ_0) and at some final (ϕ_F) state of a path. Let K denote the number of steps we consider.

Definition 1. *Given a finite path $\rho : x_0 \dots x_K$ of $\mathcal{T}^{\mathcal{C}}$ we say that ρ satisfies path predicate $\Phi = (\phi_0, \phi_F)$, denoted as $\rho \vDash \Phi$, if and only if $\phi_0(x_0) \wedge \phi_F(x_K)$ evaluates to true and no reactions are enabled in x_K (i.e. x_K is a terminal state)[2].*

We define the probability of Φ, denoted P_Φ, using $\mathcal{M}^{\mathcal{C}}$ as follows. Let $X_0 = \{x \in X \mid \phi_0(x)\}$ denote the set of states that satisfy the initial state predicate. We initialise $\mathcal{M}^{\mathcal{C}}$ with a uniform sample from the states that satisfy ϕ_0, which defines π_0 as

$$\pi_0(x) = \begin{cases} \frac{1}{|X_0|} & \text{if } x \in X_0 \\ 0 & \text{otherwise} \end{cases}$$

---

[2] We consider terminating computations by enforcing that no reactions are enabled at the state that satisfies ϕ_F. Alternative strategies possible within our approach could consider reaching a fix-point (i.e. the firing of any enabled reaction does not cause a transition to a different state), or reaching a cycle along which ϕ_F is satisfied, to guarantee that the correct output is eventually computed and remains unchanged by any subsequent reactions.

Similarly, $X_F = \{x \in X \mid \phi_F(x)\}$ denotes the set of states satisfying the final state predicate.

Definition 2. *The probability of Φ is defined as*

$$P_\Phi = \sum_{x \in X_F} \pi_t(x),$$

where t denotes the maximal time we consider and π_t is the probability vector at time t computed using the CME introduced in Section 2. In other words, we define P_Φ as the average probability of the states satisfying ϕ_F at time t.

Note that it is possible to optimise for both speed and accuracy by, for example, defining P_Φ to be the integration of the probability mass of all states satisfying ϕ_F from time 0 to time t.

Problem 1. Given a finite set of path predicates $\{\Phi_0, \dots, \Phi_n\}$, find a bimolecular CRN $\mathcal{C}$ such that

1. for each Φ_i, there exists a path ρ_i of $\mathcal{T}^\mathcal{C}$, such that $\rho_i \vDash \Phi_i$ and
2. the average probability $\frac{\sum_{i=0}^{n} P_{\Phi_i}}{n+1}$ defined using $\mathcal{M}^\mathcal{C}$ is maximised.

4 Synthesis and tuning of CRNs

We solve Problem 1 by addressing each of the two subproblems separately. First, we generate a number of CRNs that satisfy the specifications from Problem 1.1 using a satisfiability modulo theories (SMT)-based approach (Section 4.1). The CRNs identified at that point are capable of producing a path that satisfy each path predicate, which addresses Problem 1.1 but they might also include incorrect paths and the probability of correct computations might be low. Therefore, we tune the reaction rates of these CRNs in order to maximise the average probability (discussed in Section 4.2), which addresses Problem 1.2.

4.1 SMT-based synthesis

Here, we present our approach to finding a bimolecular CRN $\mathcal{C}$ that satisfies a specification expressed as path predicates $\{\Phi_0, \dots, \Phi_n\}$ (Problem 1.1). We address this problem by encoding $\mathcal{T}^\mathcal{C}$ symbolically for any possible bimolecular CRN $\mathcal{C} = (\Lambda, \mathcal{R})$ where $|\mathcal{R}| = M$ and $|\Lambda| = N$ (i.e. the number of species and reactions is given), together with the specification $\{\Phi_0, \dots, \Phi_n\}$ for some finite number of steps K, as a satisfiability modulo theories (SMT) problem. We then use the SMT solver Z3 [12] to enumerate bimolecular CRNs that satisfy the specification or prove that no such CRNs exists for the given N, M, and K. Finally, we apply an incremental procedure to search for CRNs of increasing complexity (larger N and M) or to provide more complete results by increasing K.

   Using Z3's theory of linear integer arithmetic, we represent the stoichiometry of $\mathcal{C}$ as two symbolic matrices $\mathbf{r} \in \mathbb{N}_0^{M \times N}$ and $\mathbf{p} \in \mathbb{N}_0^{M \times N}$ (using integer

constraints to prohibit negative integers). Given a reaction $r \in \mathcal{R}$ and species $s \in \Lambda$, $\mathbf{r}_s^r$ ($\mathbf{p}_s^r$) defined in Section 2 is now encoded as a symbolic integer. We ensure that only bimolecular CRNs are considered by asserting the constraints $\bigwedge_{i=0}^{M-1} \sum_{j=0}^{N-1} \mathbf{r}_{i,j} = 2$ and $\bigwedge_{i=0}^{M-1} \sum_{j=0}^{N-1} \mathbf{p}_{i,j} = 2$. In addition, we introduce the following constraints.

- We label a subset of the species $\Lambda_I \subseteq \Lambda$ as inputs and assert that $\bigwedge_{s \in \Lambda_I} \bigvee_{r \in \mathcal{R}} \mathbf{r}_s^r > 0$ to ensure all inputs are consumed by at least one reaction.
- We label a subset of the species $\Lambda_O \subseteq \Lambda$ as outputs and assert that $\bigwedge_{s \in \Lambda_O} \bigvee_{r \in \mathcal{R}} \mathbf{p}_s^r > 0$ to ensure all outputs are produced by at least one reaction.
- We assert that $\bigwedge_{r,r' \in \mathcal{R}, r \neq r'} \bigvee_{s \in \Lambda} \mathbf{p}_s^r \neq \mathbf{p}_s^{r'} \vee \mathbf{r}_s^r \neq \mathbf{r}_s^{r'}$ to ensure that two reactions never have the same reactants and products and, therefore, all M reactions are utilised.
- Finally, we assert that $\bigwedge_{r \in \mathcal{R}} \bigvee_{s \in \Lambda} \mathbf{p}_s^r \neq \mathbf{r}_s^r$ to ensure that the firing of each reaction updates the state of the system.

Following an approach inspired by bounded model checking (BMC) [18], we represent the finite path $\rho_i = x_0^i, \ldots x_K^i$ for each Φ_i by defining each state as a symbolic vector $x_j^i \in \mathbb{N}_0^N$ and "unrolling" the transition relation of $\mathcal{T}_C$ (i.e. asserting the constraint $T(x_j^i, x_{j+1}^i)$ for each $i = 0 \ldots n$ and $j = 0 \ldots K - 1$). For each path predicate $\Phi_i = (\phi_0, \phi_F)$ and path ρ_i we then assert the constraint $\phi_0(x_0^i) \wedge \phi_F(x_K^i) \wedge \textit{Terminal}(x_K^i)$ according to Def. 1, where $\textit{Terminal}(x) \triangleq \bigwedge_{r \in \mathcal{R}} \bigvee_{s \in \Lambda} x_s < \mathbf{r}_s^r$, i.e. no reactions are possible due to insufficient molecules of at least one reactant.

The parameter K specifies the maximal trajectory length that is considered. The BMC approach is conservative, since computations that require more than K steps (reaction firings) to reach a state satisfying ϕ_F will not be identified. Increasing K leads to a more complete search, and indeed the approach becomes complete for a sufficiently large K determined by the diameter of a system, but also increases the computational burden. To alleviate this, we follow an approach from [11] and consider *stutter* transitions (corresponding to multiple firings of the same reaction in a single step) by using the following modified transition relation definition T_{st} (as opposed to T from Eqn. 1)

$$\forall x, x' \in X . T_{st}(x, x') \leftrightarrow (\textit{Terminal}(x) \wedge x = x') \vee$$

$$\exists n \in \mathbb{N} . \bigvee_{r \in \mathcal{R}} \bigwedge_{s \in \Lambda} \left(x_s \geq \mathbf{r}_s^r \wedge x_s \geq n \cdot (\mathbf{r}_s^r - \mathbf{p}_s^r) \wedge x_s' = x_s + n \cdot (\mathbf{p}_s^r - \mathbf{r}_s^r) \right).$$

For any enabled reaction r ($x_s \geq \mathbf{r}_s^r$), T_{st} allows r to fire up to n times in the stutter transition. n is limited by the consumption and production of the species needed for the reaction to fire ($x_s \geq n \cdot (\mathbf{r}_s^r - \mathbf{p}_s^r)$). In many cases, stutter transitions dramatically decreases the required trajectory lengths (K), since multiple copies of the same species can react simultaneously. However, this is not restrictive, since for $n = 1$ the original definition of T is recovered. In addition to such stutter transitions, T_{st} allows self loops at terminal states, and therefore computations that require less than K steps to reach a state satisfying ϕ_F can also be identified.

The encoding strategy described so far allows us to represent CRN synthesis as an SMT-problem and apply an SMT solver such as Z3 [12] to produce a CRN that satisfies the specification or prove that no such CRN exists for the choice of M, N and K. More specifically, a solution CRN $\mathcal{C}$ is represented through the valuation of $\mathbf{r}$ and $\mathbf{p}$, which are extracted from the *model* returned by Z3.

In general, we are interested in enumerating many (or all possible) CRNs for the given class (defined by M, N and K), which ensures that no valid solutions are omitted at that stage. To do so, we apply an incremental SMT-based procedure, where at each step we assert an uniqueness constraint guaranteeing that no previously discovered CRNs are generated. Given a concrete, previously generated CRN $\mathcal{C}' = (\Lambda, \mathcal{R}')$ and the new symbolic CRN $\mathcal{C} = (\Lambda, \mathcal{R})$ we are searching for (both of which are defined using the same species Λ), we define the constraint *DifferentFrom*$(\mathcal{C}') \triangleq \neg \bigwedge_{r \in \mathcal{R}} \bigvee_{r' \in \mathcal{R}'} r = r'$, where $r = r'$ if and only if $\mathbf{r}_s^r = \mathbf{r}_s^{r'} \wedge \mathbf{p}_s^r = \mathbf{p}_s^{r'}$ for all $s \in \Lambda$. The new CRN $\mathcal{C}$ cannot simply be a permutation of the same reactions[3]. We start by generating a solution $\mathcal{C}'$ (if one exists), asserting the constraint *DifferentFrom*$(\mathcal{C}')$, and repeating this procedure until the constraints become unsatisfiable, which corresponds to a proof that not additional CRNs exists for the given N, M, and K.

4.2 Tuning CRNs with parameter optimisation

Here, we present our approach to optimising the reaction rates for CRNs satisfying $\{\Phi_0, \ldots, \Phi_n\}$. This becomes a parameter synthesis problem over a pCTMC set, analogous to parameter synthesis for a single pCTMC, as studied in [13,14]. In contrast to this work, we aggregate over the multiple input combinations, as specified in Problem 1.2.

To obtain solutions for the probability at a specified time π_t, we used numerical integration of the CME. Specifically, we used the Visual GEC software (`http://research.microsoft.com/gec`) to encode the CRNs and then integrate the CME for each combination of inputs.

To solve the maximisation problem, we used a Markov chain Monte Carlo (MCMC) method, as implemented in the Filzbach software (`http://research.microsoft.com/filzbach`). Filzbach uses a variation of the Metropolis-Hastings (MH) algorithm to perform Bayesian parameter inference. The MH algorithm is used to approximate the posterior probability of a parameter set from a hypothesised model taking on certain values, constrained by a likelihood function. The probability of each parameter value is then approximated by constructing a Markov chain of sampled parameter sets, such that a proposed parameter set is accepted with some probability, based on the ratio of the likelihood function evaluated at current and proposal parameter sets. For more information on MCMC methods, see [19]. MCMC methods, such as simulated annealing, have also been shown to efficiently find solutions to combinatorial optimisation problems [20], taking a stochastic search approach similar to the MH algorithm. Stochastic

---

[3] At present, our uniqueness constraint does not consider other CRN isomorphisms but certain species symmetries are broken by the specification Φ_i

search can provide benefits over gradient-based optimisers by maintaining a nonzero probability of making up-hill moves, protecting against getting stuck in poor local optima. To use Filzbach for providing solutions to optimisation CRN parameters, it is sufficient to encode the argument of Problem 1.2 as a likelihood function. Subsequently, we generate MCMC chains with suitably many burn-in iterations and samples to obtain an approximate optimising parameter set **k**.

4.3 Calculating expected time

To evaluate the temporal performance of a CRN algorithm $\mathcal{C}$, we make use of Markov chain theory to obtain the expected time until a terminal state is reached. This is an exact measure of the expected running time for a given pCTMC with inputs $i \in \mathcal{I}$, as opposed to using the mean of many stochastic simulations [10].

Let $A \subseteq X_r$ be the absorbing states of a pCTMC $\mathcal{M}_p^{\mathcal{C}} = (X, \pi_0, \boldsymbol{Q}_p)$ and let τ^A be a vector of expected hitting times, corresponding to the expected time of transitioning from a state $x \in X_r$ to A. Then τ^A can be evaluated as the solution to the equations (page 113 of [21])

$$\tau_x^A = 0 \text{ for } x \in A$$
$$-\sum_{x' \in X_r} q_{x,x'} \tau_{x'}^A = 1 \text{ for } x \notin A.$$

Numerical solutions can be obtained by forming a matrix W where the rows and columns of $\boldsymbol{Q}_p$ corresponding to the terminal states (A) have been removed. Then, τ^A is the solution to $W\tau^A = \boldsymbol{1}$, where $\boldsymbol{1}$ is the vector of 1's. Numerical solutions can be obtained using Gaussian elimination.

Note that the time complexity analysis of CRNs typically assumes a volume n equal to the maximum number of molecules in the system at any time [8] (equivalent to parallel time in PPs [10]). This volume can be included by dividing each propensity by n before calculating expected time (see Section 2). In the case of bimolecular CRNs this is equivalent to multiplying τ^A by n.

5 Case studies

5.1 Approximate majority

Approximate Majority is one of the most analysed functions in distributed computing. It is the approximate version of the majority problem, which cannot be exactly computed by bimolecular CRNs (or population protocols) with less than 4 species [22]. For CRNs with 2 and 3 species there are known optimal (in terms of reaction firings) approximate algorithms [15,16].

We specify the majority problem using the path predicate (see Section 2): $\Phi_{AM}(a,b) := (\phi_0(a,b), \phi_F(a,b))$, where

$$\phi_0(a,b) := \begin{cases} A = a \wedge B = b & \text{if } N = 2, \\ A = a \wedge B = b \wedge X = 0 & \text{if } N = 3 \end{cases}$$

$$\phi_F(a,b) := \begin{cases} A^m_{a,b} & \text{if } a > b \\ B^m_{a,b} & \text{if } a < b \\ A^m_{a,b} \vee B^m_{a,b} & \text{otherwise} \end{cases}$$

where $A^m_{a,b} := A = a + b \wedge B = 0$ and

$$B^m_{a,b} := A = 0 \wedge B = a + b$$

We used inputs $a, b \in [1 \ldots 5]^2 \cup [6 \ldots 10]^2$ for both optimisation and synthesis. We applied the SMT approach to identify all CRNs with 2 to 4 reactions and 2 or 3 species that satisfy Φ_{AM} for $K \leq 5$ stutter steps (for N species and M reactions, there are $\binom{N^2(N^2-1)}{M}$ total possible CRNs). We used a short optimisation (20 burn-in, 20 samples) and sorted these solutions by the value of $P_{\Phi_{AM}}$ for each. We then applied a longer optimisation (700 burn-in, 700 samples) to the top 10 CRNs (Fig. 1).
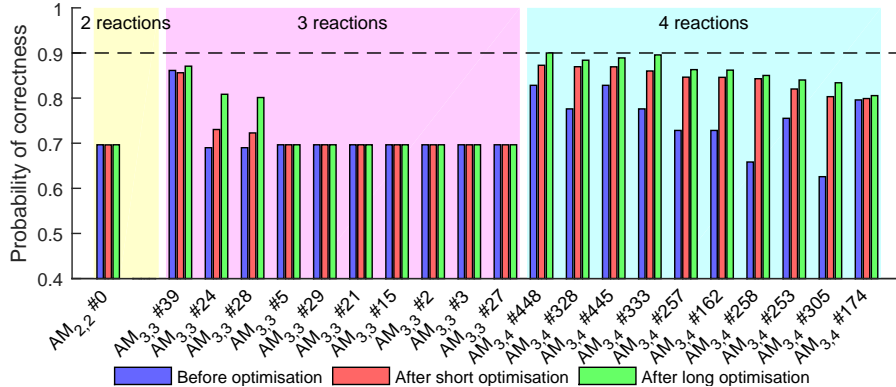

Fig. 1. Performance of approximate majority circuits. The SMT-based method was applied to the approximate majority specification for CRNs with 2, 3 and 4 reactions. For each category, the top 10 CRNs satisfying Φ_{AM} are ordered by their average probability after a short optimisation (20 burn-in, 20 samples; red bars). A longer optimisation (700 burn-in, 700 samples; green bars) was also performed. We also show the average probabilities before optimisation (all rates equal to 1.0; blue bars). The dashed line is the average probability of CRN $AM_{3,4}$ #448 after the longer optimisation, 0.8999, the maximum average probability in this trail.

Using our approach, we found 1 CRN with 2 reactions and 2 species, the known *direct competition* (DC) network [23] (Fig. 2a). Out of 59,640 possible CRNs
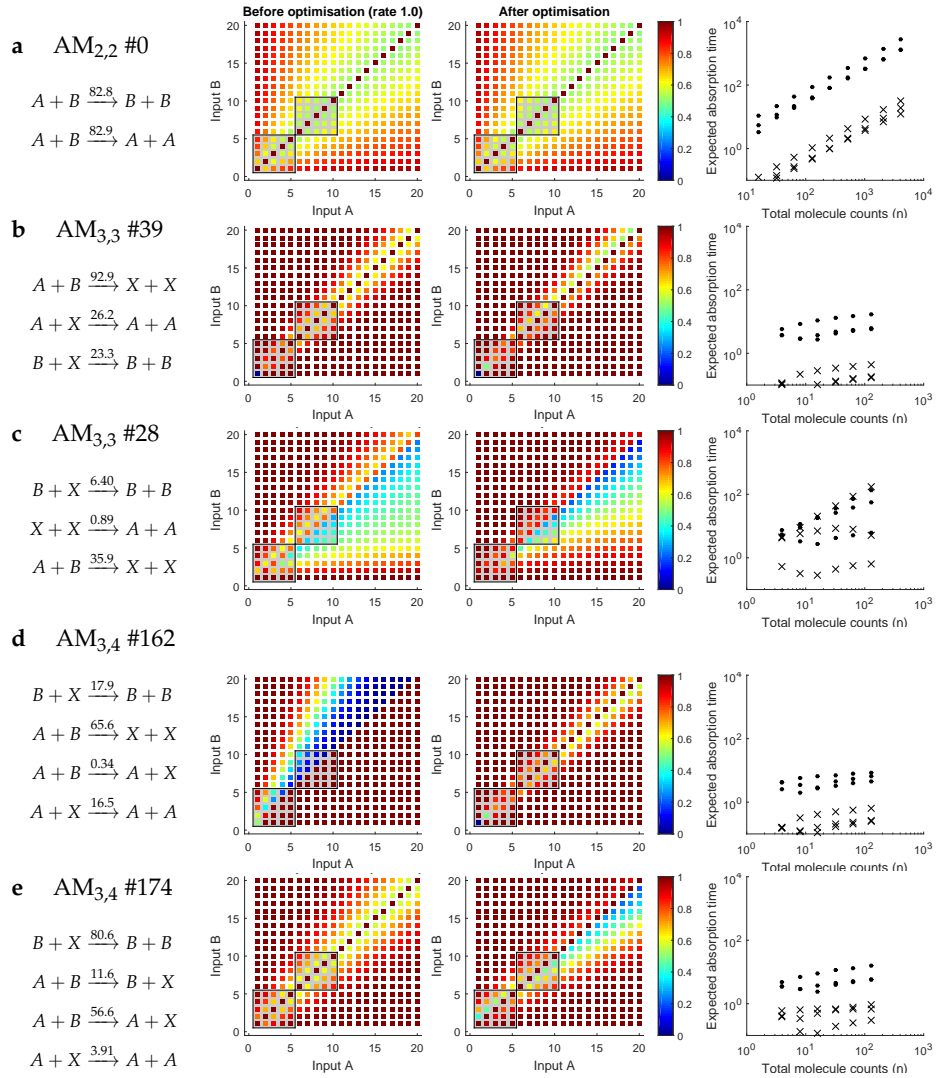
a $AM_{2,2}$ #0

$$A + B \xrightarrow{82.8} B + B$$
$$A + B \xrightarrow{82.9} A + A$$

b $AM_{3,3}$ #39

$$A + B \xrightarrow{92.9} X + X$$
$$A + X \xrightarrow{26.2} A + A$$
$$B + X \xrightarrow{23.3} B + B$$

c $AM_{3,3}$ #28

$$B + X \xrightarrow{6.40} B + B$$
$$X + X \xrightarrow{0.89} A + A$$
$$A + B \xrightarrow{35.9} X + X$$

d $AM_{3,4}$ #162

$$B + X \xrightarrow{17.9} B + B$$
$$A + B \xrightarrow{65.6} X + X$$
$$A + B \xrightarrow{0.34} A + X$$
$$A + X \xrightarrow{16.5} A + A$$

e $AM_{3,4}$ #174

$$B + X \xrightarrow{80.6} B + B$$
$$A + B \xrightarrow{11.6} B + X$$
$$A + B \xrightarrow{56.6} A + X$$
$$A + X \xrightarrow{3.91} A + A$$

Fig. 2. Response of Approximate Majority algorithms to varied inputs. For each input combination, specified as initial copies of species *A* and species *B*, the probability that both have the correct molecule count after 100 time units is reported. Results are shown for a variety of networks that performed well following optimisation (see Fig. 1). The performance of each CRN is compared both before optimisation (all rates equal to 1.0; left panels) and after long optimisation (central panels). The grey boxes show the input ranges used for both generation and optimisation. The expected time until the CTMC reaches a terminal state is calculated for varying total molecule counts (*n*) (right panels). These times consider rates scaled as if occurring in a volume *n* (see Section 4.3). The completion times for three alternative initial configurations (initial copies of *A* were 10%, 60% and 90% of *n* respectively) were calculated, illustrating minor differences in circuit completion times ($\times$ marks systems using optimised rates and $\bullet$ marks systems using 1.0 for all rates).

with 3 species and 3 reactions, the SMT solver found 39 CRNs where Φ_{AM} was satisfied, 2 of which with probability over 0.696 after the short optimisation (see Fig. 1). These two networks ($AM_{3,3}$ #24 and $AM_{3,3}$ #28) are the dual of each other and behave asymmetrically but perform well owing to a compensatory asymmetric parameterisation (Fig. 2c). One might expect that we should discover the known approximate majority circuit [15,17], (see Fig. 2b). However, this CRN does not satisfy the specification Φ_{AM} since, for input $(A = 1, B = 1, X = 0)$ the network terminates in the state $(A = 0, B = 0, X = 2)$ and thus fails to make a decision. If we remove this single problematic input from the specification Φ_{AM}, then this CRN is indeed discovered. We include it for comparison as $AM_{3,3}$ #39. Note that it scores a 0 on inputs $A = 1, B = 1$.

By increasing the number of reactions to 4, the SMT solver found 515 satisfying networks out of the 1,028,790 possible ones. The top 5 networks, $AM_{3,4}$ #448, #328, #445, #333, and #257 have the same rules as the 3 reaction network $AM_{3,3}$ #39 but each has a different 4th reaction. The network $AM_{3,4}$ #162 had a lower performance than $AM_{3,3}$ #39 before optimisation and was almost as good following optimisation. This network was also asymmetric, with a corresponding asymmetric parameterisation after optimisation (Fig. 2d). The known 4 reaction network $AM_{3,4}$ #174 [17] (Fig. 2e) is also identified in 10th position.

Finally, we analysed the expected time until termination for each circuit, using the procedure in Section 4.3 (right-hand panels of Fig. 2). Note that Definition 2 does not reward circuits that reach a high probability before the final time $t_f = 100$. However, in nearly all cases, the estimated hitting time of each system was improved by optimisation.

Computation times The computation times of our procedure depend on the size of the circuit (M and N), length of considered computations (K) and exact specification Φ (including the number of given path predicates). We illustrate the computation times required for the SMT-based synthesis part of our approach with the majority decision-making CRNs (Fig. 3).

To determine how the CME calculation used in our method scales with molecular copy numbers, we first ran calculations of the CME for the established 3-reaction approximate majority CRN (system $AM_{3,3}$ #39). The calculation was initialised with $0.6n$ copies of A and $0.4n$ copies of B, and all rates were set to 1. As increasing the copy number decreases the simulation time interval over which there are transient dynamics, we integrated the CME over the time interval $\left[0, \frac{100}{n}\right]$, where n is the total copy number. We calculated transient probabilities at 500 output points, with $n \in [10, 1000]$. This led to state-spaces of varying size, up to 10^6, with all calculations completing within 7200 seconds (2 h; Fig. 4). Smaller examples took only a few seconds.

We can approximate the total run-time for parameter tuning as a function of the number of iterations of the MCMC algorithm and the number of input combinations assessed. For example, doing 200 iterations over 10 input combinations which all have below 30 total molecules ($\lesssim$1 s each) suggests a tuning procedure of no more than 2,000 seconds.
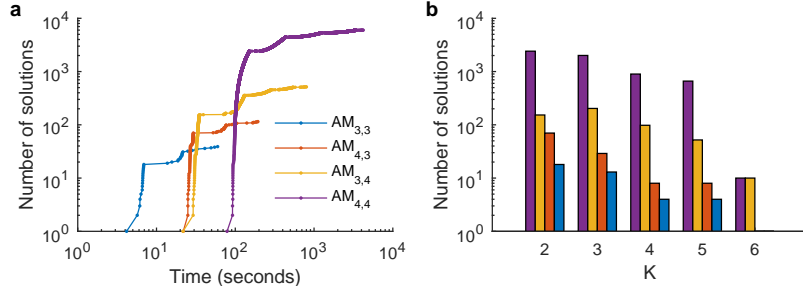
11

Fig. 3. Computation times for the SMT-based synthesis of majority decision-making CRNs. Panel (a) shows the time required to generate a number of solutions (candidate CRNs) for Φ_{AM} for N species and M reactions (denoted $AM_{N,M}$) for $N, M \in \{3,4\}^2$. The computation was halted after 2 hours. Panel (b) shows the number of solutions found as K (the length of considered computations with stutter transitions) increases.
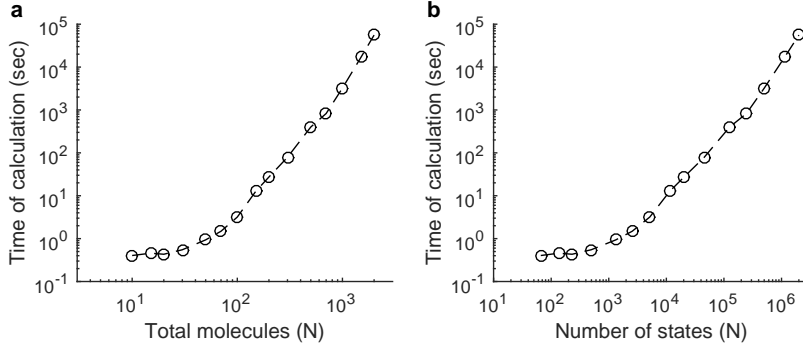

Fig. 4. Transient probability calculation times for CRN $AM_{3,3}$ #39. Times indicated include the enumeration of the state-space, construction of a sparse matrix, then numerical integration in the interval $[0, \frac{100}{n}]$, where n is the total molecule count. A single calculation was conducted for each value of n.

5.2 Division

Division is a non-semi-linear function and therefore it cannot be stably computed by CRNs [8]. However, CRNs have been proposed that might implement the calculation of a ratio [24], which allows plants to ration starch reserves during seasonally changing nights.

We specify the division problem using the path predicate (see Section 2):

$$\Phi_{\text{Div}}(a,b) := (\phi_0(a,b), \phi_F(a,b), \text{ where}$$

$$\phi_0(a,b) := \begin{cases} A = a \wedge B = b \wedge X = 0 & \text{if } N = 3 \\ A = a \wedge B = b \wedge X = 0 \wedge Y = 0 & \text{if } N = 4 \end{cases}$$

$$\phi_F(a,b) := X = \left\lfloor \frac{a}{b} \right\rfloor$$

12

We chose the input ranges $a, b \in [1, \ldots, 10]^2$ for synthesis and optimisation to give diverse selection of responses and to reinforce that $\lfloor \frac{a}{b} \rfloor = 0$ when $a < b$. We applied the SMT approach to CRNs that satisfied Φ_{Div} with $K < 20$ (without stutter transitions). For 3 species and 3 reactions, 22 CRNs were discovered. For 4 species and 3 reactions, 34 CRNs were discovered. For 4 species and 4 reactions the first 105 CRNs were discovered. Of these, only one CRN $DIV_{4,3}$ #29 exceeded an average probability of 0.5, though in most cases, optimisation improved performance substantially (Fig. 5). For many of the generated circuits, high performance was observed only for $b > a$, which should always evaluate to 0, with poor performance for the nonzero output cases of $a > b$ (Fig. 6a,b). Note that $Div_{4,3}$ #29 is so far the top scoring divider CRN in this class. Clearly, none of these circuits can be considered as *good* algorithms for computing division, though our procedure was able to detect some very simple yet mediocre circuits in an automated way. It is possible that better circuits will be found by considering CRNs with more reactions, species, and longer computation paths.
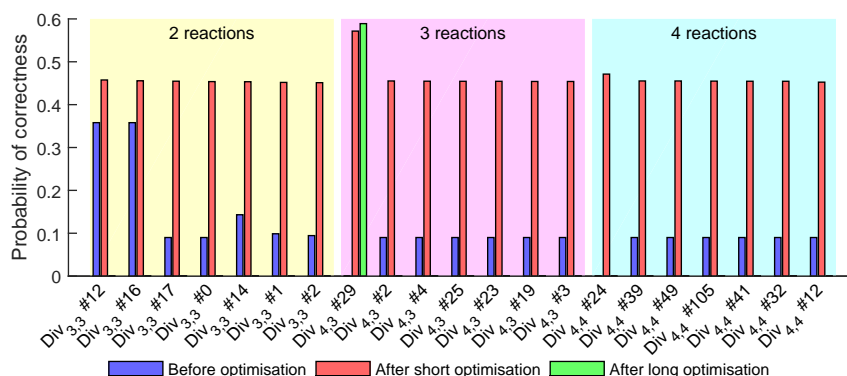

Fig. 5. Performance of division circuits. The SMT-based method was applied to the division specification for CRNs with N species and M reactions for $N, M \in \{(3,3), (3,4), (4,4)\}$. This figure shows the optimisation results for the top 7 CRNs in each category. The results are ranked and sorted by their average probability of being correct in the grey shaded zone after being optimised for 50 MCMC sample and burn-in steps (red bars). If a CRN scored an average probability of over 0.5 then it was optimised for a further 200 MCMC burn-in and sample steps. The average probability is shown for satisfying CRNs before optimisation (all rates equal to 1.0; blue bars).

6   Discussion

In this paper, we presented a computational approach for the synthesis and parameter tuning of CRNs, given a specification of the system's correctness. We focused on the sub-class of bimolecular CRNs due to their importance
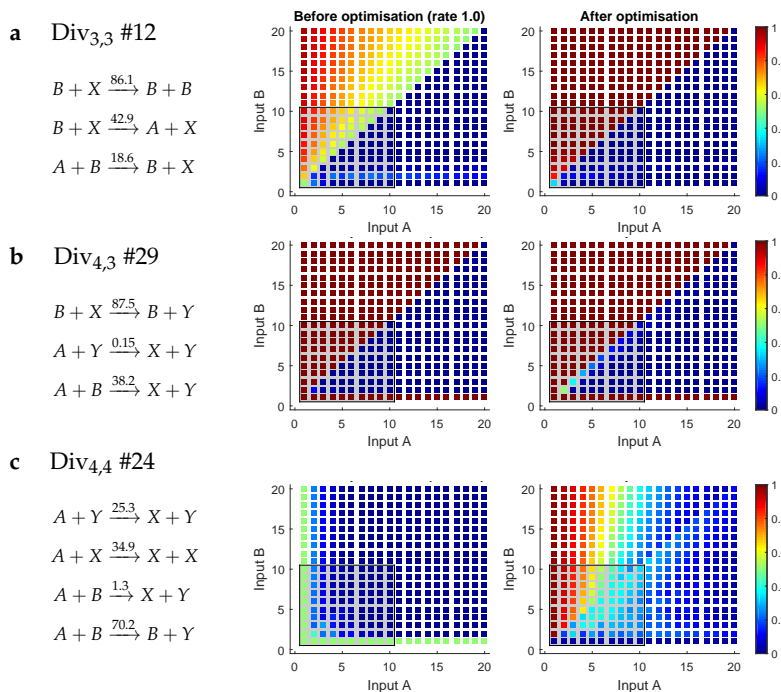
a Div$_{3,3}$ #12

$B + X \xrightarrow{86.1} B + B$

$B + X \xrightarrow{42.9} A + X$

$A + B \xrightarrow{18.6} B + X$

b Div$_{4,3}$ #29

$B + X \xrightarrow{87.5} B + Y$

$A + Y \xrightarrow{0.15} X + Y$

$A + B \xrightarrow{38.2} X + Y$

c Div$_{4,4}$ #24

$A + Y \xrightarrow{25.3} X + Y$

$A + X \xrightarrow{34.9} X + X$

$A + B \xrightarrow{1.3} X + Y$

$A + B \xrightarrow{70.2} B + Y$

Fig. 6. Response of Division algorithms to varied inputs. For each input combination, specified as initial copies of species A and species B, the probability that the molecule count of X is $\lfloor A/B \rfloor$ after 100 time units is reported. Results are shown for the top network in each combination of species and reactions (see Fig. 5). The performance of each CRN is compared both before optimisation (all rates equal to 1.0; left panels) and after optimisation (right panels).

as representations of various molecular algorithms and population protocols. However, our approach is more general and could also be applied directly to the synthesis of CRNs from other classes (e.g. unimolecular, trimolecular, etc.), which are defined through different stoichiometry constraints. The CRNs we synthesize can be converted into equivalent physical implementations, for example using DNA strand displacement (DSD) [2,3]. However, our approach could also be applied directly to synthesize DSD systems through additional structural constraints. This could lead to simpler designs than the ones obtained through direct translation of CRNs.

We considered simple reachability properties defined in terms of predicates on the initial and final states of a computation which are sufficient to express various logical and arithmetic functions and operations. More general specifications, for example where intermediate states along computations are specified, are also currently possible within our approach but extensions to more expressive lan-

14

guages, such as the probabilistic temporal logics used with other methods [14], remains a direction for future work.

An alternative approach to the problem of realising arbitrary behaviour in biochemical systems is to use directed evolution [25,26] *In silico* evolutionary search strategies might scale to larger CRNs and address the synthesis and parameter optimisation sub-problems using a single, combined procedure. However, this comes at the cost of completeness, where the absence of a solution does not mean a solution does not exist. In contrast, our method addresses the sub-problems separately and uses the SMT solver and theorem prover Z3 to identify CRNs that satisfy a given specification (kinetics are ignored at this first stage). Since the results provided by Z3 are complete (for a sufficiently large K), the termination of the procedure with no solutions is a "proof" that no CRNs exist in the given class. Thus, besides providing a practical tool for the identification of CRNs with given behaviour, the completeness property means our approach could also help explore the theoretical limits of CRN computation (e.g. no CRNs with less than M species and N reactions that compute a given function exists). For many applications, elements of our method could be complementary with evolutionary algorithms. For example, the exact CTMC methods we use to assess the probability of correct computations in a given CRN could provide a useful fitness function for evolutionary search, compared to alternative approximate methods based on stochastic simulation.

The fully automated generation of "good" CRNs is a challenging problem and certain scalability limitations of our current method must be addressed to provide a more complete solution. Firstly, the SMT-based synthesis procedure we propose may represent large or infinite state spaces and handle systems with large molecule numbers. However, currently this method is limited to relatively small CRNs with few reactions, species, and which have short computation paths. Secondly, the CTMC methods we apply require an explicit representation of the state space, which must be finite (which is always the case for biomolecular CRNs initialised with a finite number of molecules) and contain few reachable states — this makes the method suitable for systems involving relatively few species and numbers of molecules. To circumvent the need for an explicit representation of the state space, stochastic dynamical behaviour could be approximated by averaging multiple trajectories from Gillespie's stochastic simulation algorithm [27], using fluid or central limit approximations [28], or using ordinary differential equations. Depending on the specification, and the nature of the CRN, some of these approaches might be appropriate, but none are free of their own documented limitations. Finally, the large number of solutions identified at the synthesis stage of our approach makes the parameter tuning phase challenging and indicates that additional constraints describing more accurately the structure and dynamics of "good" solutions could improve the method.

For tuning reaction rates, alternative cost functions could be used that reward solutions that are "nearly" correct, e.g. using a mean-squared error. This would be most appropriate in high copy number situations, where a precise number of

molecules is not integral. Our approach is more appropriate for systems operating at low copy numbers, offering an exact characterisation of the probability that a specific predicate is satisfied. Our results were shown for calculations at $t_f = 100$ time units, a transient probability, rather than at the stationary distribution. While the selection of t_f is subjective, it allows a circuit programmer to specify how long they are willing to wait for a computation. Circuits that reach high probability at $t > t_f$ will not be rewarded. However, a natural extension to the presented method would be to reward circuits that reach high probability at $t < t_f$, both imposing an upper bound on time and optimising within that range. This could be achieved by integrating our metric over the interval $[0, t_f]$.

Automating the search for CRNs that compute the solution for a specified problem would be beneficial to both theoretical and experimental molecular programmers. Our method can be used to show the existence or absence of CRNs of a certain size and also suggest CRNs that can be tuned for a specific input range, and so become candidate designs for experimental construction. Prior to construction, more in-depth analysis of the candidate CRNs produced is beneficial, including parameter sensitivity/robustness analysis and bifurcation analysis (where appropriate). Future work could also incorporate notions of robustness into the proposed method, for example by using interval-based methods [14]. Our results illustrate the potential of this approach on several examples, including the majority and division functions discussed here.

Acknowledgements

References

1. T. Wilhelm, "The smallest chemical reaction system with bistability.," *BMC Syst Biol*, vol. 3, p. 90, 2009.
2. D. Soloveichik, G. Seelig, and E. Winfree, "DNA as a universal substrate for chemical kinetics," *PNAS*, vol. 107, pp. 5393–5398, 2010.
3. Y.-J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig, "Programmable chemical controllers made from DNA," *Nature Nanotechnology*, vol. 8, no. 10, pp. 755–762, 2013.
4. T. Fujii and Y. Rondelez, "Predator-prey molecular ecosystems.," *ACS Nano*, vol. 7, no. 1, pp. 27–34, 2013.
5. J. Kim and E. Winfree, "Synthetic in vitro transcriptional oscillators," *Mol. Syst. Biol.*, vol. 7, no. 1, 2011.
6. M. Cook, D. Soloveichik, E. Winfree, and J. Bruck, "Programmability of chemical reaction networks," in *Algorithmic Bioprocesses*, Natural Computing Series, pp. 543–584, Springer, 2009.
7. D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta, "Computation in networks of passively mobile finite-state sensors," *Distributed Computing*, vol. 18, no. 4, pp. 235–253, 2006.

8. H.-L. Chen, D. Doty, and D. Soloveichik, "Deterministic function computation with chemical reaction networks," in *DNA Computing and Molecular Programming*, vol. 7433 of *LNCS*, pp. 25–42, Springer, 2012.

9. D. Angluin, J. Aspnes, and D. Eisenstat, "Stably computable predicates are semilinear," in *PODC 2006*, pp. 292–299, 2006.

10. D. Angluin, J. Aspnes, and D. Eisenstat, "Fast computation by population protocols with a leader," *Distributed Computing*, vol. 4167, pp. 61–75, 2006.

11. B. Yordanov, C. M. Wintersteiger, Y. Hamadi, A. Phillips, and H. Kugler, "Functional analysis of large-scale DNA strand displacement circuits," in *DNA Computing and Molecular Programming*, vol. 8141 of *LNCS*, pp. 189–203, Springer, 2013.

12. L. M. de Moura and N. Bjørner, "Z3: An Efficient SMT Solver," in *TACAS*, vol. 4963 of *LNCS*, pp. 337–340, Springer, 2008.

13. T. Han, J. Katoen, and A. Mereacre, "Approximate parameter synthesis for probabilistic time-bounded reachability," in *Real-Time Systems Symposium, 2008*, pp. 173–182, IEEE, 2008.

14. M. Češka, F. Dannenberg, M. Kwiatkowska, and N. Paoletti, "Precise parameter synthesis for stochastic biochemical systems," in *CMSB*, pp. 86–98, Springer, 2014.

15. D. Angluin, J. Aspnes, and D. Eisenstat, "A simple population protocol for fast robust approximate majority," *Distributed Computing*, vol. 21, no. 2, pp. 87–102, 2008.

16. E. Perron, D. Vasudevan, and M. Vojnovic, "Using three states for binary consensus on complete graphs," in *IEEE Infocom 2009*, IEEE Communications Society, 2009.

17. L. Cardelli, "Morphisms of reaction networks that couple structure to function," *BMC Systems Biology*, vol. 8, no. 1, p. 84, 2014.

18. A. Biere, A. Cimatti, E. M. Clarke, and Y. Zhu, "Symbolic Model Checking without BDDs," in *TACAS*, (London, UK, UK), pp. 193–207, Springer, 1999.

19. C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2nd ed., 2004.

20. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

21. J. R. Norris, *Continuous-time Markov Chains*. Cambridge University Press, 1997.

22. G. B. Mertzios, S. E. Nikoletseas, C. Raptopoulos, and P. G. Spirakis, "Determining majority in networks with local interactions and very small local memory," in *ICALP*, pp. 871–882, 2014.

23. L. Cardelli and A. Csikász-Nagy, "The cell cycle switch computes approximate majority," *Scientific Reports*, vol. 2, p. 656, 2012.

24. A. Scialdone, S. T. Mugford, D. Feike, A. Skeffington, P. Borrill, and *et al.*, "Arabidopsis plants perform arithmetic division to prevent starvation at night," *eLife*, vol. 2, 2013.

25. O. S. Soyer and S. Bonhoeffer, "Evolution of complexity in signaling pathways," *PNAS*, vol. 103, no. 44, pp. 16337–16342, 2006.

26. H. Dinh, N. Aubert, N. Noman, T. Fujii, Y. Rondelez, and H. Iba, "An effective method for evolving reaction networks in synthetic biochemical systems," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 3, pp. 374–386, 2014.

27. D. Gillespie, "Exact Stochastic Simulation of Coupled Chemical Reactions," *The Journal of Physical Chemistry*, vol. 81, pp. 2340–2361, 1977.

28. S. N. Ethier and T. G. Kurtz, *Markov processes: characterization and convergence*, vol. 282. John Wiley & Sons, 2009.