Continuous Valuations of Temporal Logic Specifications with applications to Parameter Optimization and Robustness Measures

Aurélien Rizk[a] , Grégory Batt[a], François Fages[a], Sylvain Soliman[a]

[a] Project-team Contraintes, INRIA Paris-Rocquencourt,
BP105, 78153 Le Chesnay Cedex, France.
`http://contraintes.inria.fr`

Finding mathematical models satisfying a specification built from the formalization of biological experiments, is a common task of the modeler that techniques like model-checking help solving, in the qualitative but also in the quantitative case. In this article we define a continuous degree of satisfaction of temporal logic formulae with constraints. We show how such a satisfaction measure can be used as a fitness function with state-of-the-art evolutionary optimization methods in order to find biochemical kinetic parameter values satisfying a set of biological properties formalized in temporal logic. We also show how it can be used to define a measure of robustness of a biological model with respect to some temporal specification. These methods are evaluated on models of the cell cycle and of the MAPK signaling cascade.

1. Introduction

Temporal logics [16,10] have proven useful as specification languages for describing the behavior of a broad variety of systems ranging from electronic circuits to software programs, and more recently biological systems in either boolean [15,8], discrete [5], stochastic [6,26] or continuous [7,1,8,3] settings.

Because temporal logics allow one to express both qualitative (e.g. some protein is eventually produced) and quantitative (e.g. a concentration exceeds 10) information about time and systems variables, they provide a powerful specification language in comparison with the essentially qualitative properties considered in dynamical systems theory (e.g. multistability, existence of oscillations) or with the exact quantitative properties considered in optimization theory (e.g. curve fitting). In particular, these logics are well suited to the increasingly quantitative, yet incomplete, uncertain and imprecise information now accumulated in the field of quantitative systems biology.

Such a use of temporal logics relies on a logical paradigm for systems biology which consists in making the following identifications [19]:

biological model = transition system
biological properties = temporal logic formulae
biological validation = model-checking

This paradigm has been used in many applications, first for querying large interaction maps such as Kohn's map of the cell cycle [8,29] or gene regulatory networks [3], then for specifying the biological properties known or inferred [17] from experiments, validating models, discriminating between models and proposing new biological experiments [5], finding parameter values [7], and estimating robustness [4]. An important limitation of this approach is however due to the logical nature of temporal logic specifications and their boolean interpretation. A yes/no answer to a temporal logic query does not provide indeed any information on how far we are from satisfaction, or how to guide the search to satisfy a formula. A measure of how close a model is to satisfy a property is needed.

In this paper, we define a continuous violation degree that quantifies how far from satisfaction an LTL formula is in a given model. In order to accommodate the various kinds of quantitative models defined by either ordinary or stochastic differential equation systems [38,39], rule-based languages like SBML [27] or BIOCHAM [19], hybrid Petri nets [24,32], stochastic process calculi [35,34], etc..., we represent the behavior of the system simply by numerical traces [17,33,7,1], so our method is rather general. Our notion of violation degree is then used for two applications in systems biology: the search of kinetic parameter values in a model, and the quantitative estimation of the robustness of a model by adapting the general framework of Kitano [28] to our temporal logic setting.

Section 2 presents the quantifier free fragment of first-order linear time logic with constraints over the reals, QFLTL($\mathbb{R}$), studied in [17] and used in this paper. Section 4 defines a real-valued degree of satisfaction of a (QF)LTL formula on a trace using an aggregation function which composes the distances between the validity domains of some variables in a QFLTL formula and their objective value in the given specification.

Section 5 shows how such a continuous degree of satisfaction of an LTL formula can be used as a fitness function in local search methods for finding kinetic parameter values satisfying a temporal logic specification. We describe a general local search method and use the state-of-the-art Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [25] to evaluate our method on models of the budding yeast cell cycle with 8 parameters and of the MAPK signaling cascade with 30 parameters and 7 unknown initial conditions.

In section 6 we propose a definition for the robustness degree of a property w.r.t. a model and a set of perturbations weighted by probabilities. This definition is inspired by the abstract definition of robustness proposed by Kitano for systems biology [28]. We develop it here in our temporal logic setting and illustrate its relevance by applying it to the previous model of the cell cycle.

2. Preliminaries on Temporal Logics with Constraints over the Reals

2.1. LTL($\mathbb{R}$)

The *Linear Time Logic* LTL is a temporal logic [10] that extends classical logic with modal operators for qualifying when a formula is true in a *trace*, i.e. an infinite sequence of timed states. Temporal operators include **X** ("next", for at the next time point), **F** ("finally", for at some time point in the future), **G** ("globally", for at all time points in the future), **U** ("until") and **W** (" weak until"). These operators enjoy some simple duality properties, $\neg\ \mathbf{X}\ \phi =\ \mathbf{X}\ \neg\phi$, $\neg\ \mathbf{F}\ \phi =\ \mathbf{G}\ \neg\phi$, $\neg\ \mathbf{G}\ \phi =\ \mathbf{F}\ \neg\phi$, $\neg(\psi\ \ \mathbf{U}\ \ \phi) = (\neg\phi\ \ \mathbf{W}\ \ \neg\psi)$. These properties make it possible to eliminate negations by descending them to atomic propositions. Furthermore, we have $\phi\ \ \mathbf{W}\ \ \psi = (\phi\ \ \mathbf{U}\ \ \psi) \vee\ \mathbf{G}\ \ \phi$, hence

LTL can be defined with { **X** **F** , **G** , **U** } as basis of temporal operators.

A version of LTL with constraints over the reals, named LTL($\mathbb{R}$), has been proposed in [1,7] to express temporal properties about molecular concentrations. The atomic formulae of LTL($\mathbb{R}$) are formed with inequality relations and arithmetic operators over the real values of molecular concentrations (denoted by $[A]$, ...), possibly their derivatives (denoted by $d[A]/dt$, ...) and time. Since the relation symbols are closed by negations, and negations and implications can be eliminated by propagating the negations down to the atomic constraints in the formula, we assume that all LTL($\mathbb{R}$) formulae are in *negation free normal form.* The precise syntax of LTL($\mathbb{R}$) is given in Table 1.

$$\begin{array}{ll}
\textit{Formula ::=} & \textit{Atom} \mid \textit{Formula} \wedge \textit{Formula} \mid \textit{Formula} \vee \textit{Formula} \\
 & \mid \mathbf{X}\ \textit{Formula} \mid \mathbf{F}\ \textit{Formula} \mid \mathbf{G}\ \textit{Formula} \\
 & \mid \textit{Formula}\ \mathbf{U}\ \textit{Formula} \\
\textit{Atom ::=} & \textit{Term Op Term} \\
\textit{Op ::=} & < \mid > \mid \leq \mid \geq \\
\textit{Term} & \textit{Real} \mid \textit{Term + Term} \mid \textit{Term - Term} \mid \textit{- Term} \\
 & \mid \textit{Term} \times \textit{Term} \mid \textit{Term / Term} \mid \textit{Term}\ \hat{}\ \textit{Term} \\
\textit{Real ::=} & \textit{float} \mid [\textit{molecule}] \mid \mathrm{d}[\textit{molecule}]/\mathrm{dt} \mid \mathrm{Time}
\end{array}$$

Table 1
Syntax of LTL($\mathbb{R}$) formulae.

For instance, `F([A]>10)` expresses that the concentration of `A` eventually gets above the threshold value 10. `G([A]+[B]<[C])` expresses that the concentration of C is always greater than the sum of the concentrations of A and B. Oscillation properties, abbreviated as `oscil(M,K)`, are defined as a change of sign of the derivative of M at least K times:
`F((d[M]/dt > 0)` $\wedge$ `F((d[M]/dt < 0)` $\wedge$ `F((d[M]/dt > 0)...)))`

LTL($\mathbb{R}$) formulae are interpreted over infinite *traces* of the form

$$(< t_0, \boldsymbol{x}_0, d\boldsymbol{x}_0/dt >, < t_1, \boldsymbol{x}_1, d\boldsymbol{x}_1/dt >, ...)$$

which gives the concentration values $\boldsymbol{x}_i$ of the molecules, and the values of their first derivatives $d\boldsymbol{x}_i/dt$, at discrete time points t_i. We shall consider infinite traces obtained from finite traces by completing them with a loop on the last state. For instance, in a model described by a system of ordinary differential equations (ODE), and under the hypothesis that the initial state is completely defined, numerical integration methods (such as Runge-Kutta or Rosenbrock method for stiff systems) compute a finite simulation trace. To extend it to an infinite trace, a loop is added on the last state with the assumption that the finite time horizon considered for the numerical integration is sufficiently large to check the properties at hand.

It is worth noticing that the semantics of the "next" operator X refers to the next time point on the trace, and that in adaptive step size integration methods of ODE systems, the step size $t_{i+1} - t_i$ is not constant but determined through an estimation of the error made by the discretization. The X operator can be used to detect local optima in a trace for instance.

Formally, the truth value of an LTL($\mathbb{R}$) formula in a trace π is given in Table 2.

$\pi \models \phi$	iff	$s_0 \models \phi$
$s_i \models \alpha$	iff	α is an atomic formula and α is true at state s_i,
$s_i \models \phi \wedge \psi$	iff	$s_i \models \phi$ and $s_i \models \psi$,
$s_i \models \phi \vee \psi$	iff	$s_i \models \phi$ or $s_i \models \psi$,
$s_i \models \mathbf{X}\, \phi$	iff	$s_{i+1} \models \phi$,
$s_i \models \mathbf{F}\, \phi$	iff	there exists $j \geq i$ s.t. $s_j \models \phi$,
$s_i \models \mathbf{G}\, \phi$	iff	for all $j \geq i$, $s_j \models \phi$,
$s_i \models \phi\, \mathbf{U}\, \psi$	iff	there exists $j \geq i$ s.t. $s_j \models \psi$ and $s_k \models \phi$ for all $i \leq k < j$.

Table 2
Inductive definition of the truth value of an LTL($\mathbb{R}$) formula in a trace $\pi = (s_0, s_1, \ldots)$.

These truth values can be computed on traces by a model-checking algorithm [7].

2.2. QFLTL($\mathbb{R}$)

In [17], the quantifier free fragment of first-order LTL($\mathbb{R}$), named QFLTL($\mathbb{R}$), has been considered for the purpose of analyzing numerical data time series in temporal logic and computing automatically LTL($\mathbb{R}$) specifications from experimental traces. Syntactically, QFLTL($\mathbb{R}$) simply adds variables to the terms:

$$\textit{Real} ::= \quad \textit{variable} \;\mid \textit{float} \;\mid [\textit{molecule}] \;\mid \mathrm{d}[\textit{molecule}]/\mathrm{dt} \mid \ldots$$

For instance, the QFLTL($\mathbb{R}$) formula $G([A] < x)$ expresses the constraint that x is always greater than the concentration of A, or equivalently, that x is greater than the maximum of $[A]$. Obviously, an LTL($\mathbb{R}$) formula is a QFLTL($\mathbb{R}$) formula without variable.

As usual, the semantics of a QFLTL($\mathbb{R}$) formula containing variables is defined by its ground instances which are LTL($\mathbb{R}$) formulae. Given a trace π and a QFLTL($\mathbb{R}$) formula ϕ over a vector $\boldsymbol{x}$ of k real-valued variables, the *constraint satisfaction problem*, $\exists \boldsymbol{v} \in \mathbb{R}^k\ \phi[\boldsymbol{v}/\boldsymbol{x}]$, is the problem of determining the valuations $\boldsymbol{v}$ of the variables for which the formula ϕ is true. The domain of validity of a formula ϕ over variables $\boldsymbol{x}$ with respect to the trace π is defined as

$$\mathcal{D}_{\pi,\phi} = \{\boldsymbol{v} \in \mathbb{R}^k \;\mid\; \pi \models \phi[\boldsymbol{v}/\boldsymbol{x}]\}$$

LTL($\mathbb{R}$) *model-checking* has been generalized to QFLTL($\mathbb{R}$) *constraint solving* in [17], by showing how to compute the exact domain of validity $\mathcal{D}_{\pi,\phi}$ for any QFLTL($\mathbb{R}$) formula ϕ. This algorithm uses the following characterization of validity domains:

Proposition 2.1 (Inductive definition of validity domain) *Let* $\pi = (s_0, s_1, \ldots, s_n)$ *be a finite trace and* ϕ *be a QFLTL formula with* k *variables, we have*

- $\mathcal{D}_{\pi,\phi} = \mathcal{D}_{s_0,\phi}$,
- $\mathcal{D}_{s_i,\alpha} = \{\boldsymbol{v} \in \mathbb{R}^k \mid \ s_i \models \alpha[\boldsymbol{v}/\boldsymbol{x}]\}$ *for an atomic proposition* α,
- $\mathcal{D}_{s_i,\phi\wedge\psi} = \mathcal{D}_{s_i,\phi} \cap \mathcal{D}_{s_i,\psi}$,
- $\mathcal{D}_{s_i,\phi\vee\psi} = \mathcal{D}_{s_i,\phi} \cup \mathcal{D}_{s_i,\psi}$,
- $\mathcal{D}_{s_i,\mathbf{X}\phi} = \mathcal{D}_{s_{i+1},\phi}$,
- $\mathcal{D}_{s_i,\mathbf{F}\phi} = \cup_{j\geq i}\mathcal{D}_{s_j,\phi}$,
- $\mathcal{D}_{s_i,\mathbf{G}\phi} = \cap_{j\geq i}\mathcal{D}_{s_j,\phi}$,
- $\mathcal{D}_{s_i,\phi\mathbf{U}\psi} = \cup_{j\geq i}(\mathcal{D}_{s_j,\psi} \cap \cap_{k\in[i,j-1]}\mathcal{D}_{s_k,\phi})$.

The QFLTL($\mathbb{R}$) constraint solving algorithm computes the validity domains of the variables of the formula in each time point of the trace, by double induction on the subformulae of ϕ and on the states of π [17]. This algorithm is generalized in [18] to a fixpoint algorithm for solving Computation Tree Logic (CTL) constraints in branching structures over an arbitrary constraint domain.

3. Implementation and Complexity

The atomic propositions considered so far for QFLTL($\mathbb{R}$) include non-linear inequalities which would be too general to be computed efficiently without approximations. To get a handle on the complexity of QFLTL constraint solving, we first describe the restricted fragment of QFLTL formulae containing at most one variable per atomic constraint, for which solution sets are represented by finite unions of orthotopes as domains, and then describe our implementation of the linear constraint fragment of QFLTL for which the validity domains are finite unions of polytopes.

3.1. Orthotopes

We consider here the fragment of QFLTL($\mathbb{R}$), named QFLTL($\mathbb{R}_{\text{box}}$), in which each atomic formula contains at most one variable:

$$\begin{array}{ll} \textit{Atom} ::= & \textit{Term Op Term} \mid \textit{Term Op variable} \\ \textit{Real} ::= & \textit{float} \ \mid [\textit{molecule}] \ \mid \mathrm{d}[\textit{molecule}]/\mathrm{dt} \mid \ldots \end{array}$$

The validity domain of a conjunction of such constraints in one variable is an interval over the reals, and in higher-dimensions over v variables, is an *orthotope*, i.e. a cartesian product of v intervals over $\mathbb{R}$. These orthotopes are computed in the base case of the inductive definition 2.1 of validity domains. For the other cases, the QFLTL($\mathbb{R}_{\text{box}}$) constraint solving algorithm computes finite intersections and unions of validity domains. As a finite intersection of orthotopes is a finite union of orthotopes, the domains computed by the algorithm are always finite unions of orthotopes.

In [17], we have shown that the number of orthotopes needed to represent the validity domain of a QFLTL($\mathbb{R}_{\text{box}}$) formula may be exponential in the number of variables, but is polynomial in the size of the formula and in the size of the trace:

Proposition 3.1 *The validity domain of a QFLTL($\mathbb{R}_{box}$) formula of size f containing k variables, on a trace of length n is a union of at most $(nf)^{2k}$ orthotopes.*

One example necessitating $(nf)^k$ orthotopes is given in [17].

3.2. Polytopes

The linear fragment QFLTL($\mathbb{R}_{\text{lin}}$) generalizes QFLTL($\mathbb{R}_{\text{box}}$) by allowing linear constraints as atomic propositions. The validity domains of such linear constraints are polytopes, and since the intersection of two polytopes is a polytope, the QFLTL($\mathbb{R}_{\text{lin}}$) constraint solving algorithm computes finite unions of polytopes.

Proposition 3.2 *The validity domain of a QFLTL($\mathbb{R}_{lin}$) formula of size f on a trace of length n is a union of at most 2^{nf} polytopes.*

Proof. As a polytope is represented by a set of linear constraints, let us consider the number of possible linear constraints appearing in the validity domain $\mathcal{D}_{\pi,\phi}$ of a formula ϕ of size f on a trace π of size n. These constraints are only generated by the base case of the inductive definition 2.1 of validity domains. Each constraint in ϕ can be evaluated on each time point of the trace, thus creating at most n different constraints. As intersections and unions do not generate new constraints and as the number of constraints in ϕ is less than f, the number of constraints in $\mathcal{D}_{\pi,\phi}$ is less than nf.

Because a polytope is defined by an arbitrary number of constraints, with at most nf constraints one can form at most 2^{nf} polytopes. Therefore, validity domain $\mathcal{D}_{\pi,\phi}$ is a union of at most 2^{nf} polytopes.

□

This fragment QFLTL($\mathbb{R}_{\text{lin}}$) is what is currently implemented in BIOCHAM v2.8 [21] using the Parma Polyhedra Library [2] for computing union and intersection of sets of polytopes. The performance evaluation given in Section 5 refers to this implementation, and indicates that the upper complexity bound is not reached in practice where most domains are simple orthotopes.

4. Continuous Satisfaction Degree of Temporal Logic Formulae

In order to evaluate numerically the fitness of a model w.r.t. a temporal logic specification, we show in this section how QFLTL($\mathbb{R}$) formulae can be evaluated with a continuous satisfaction degree in the interval $[0, 1]$, instead of a Boolean value. This continuous valuation of temporal logic formulae opens up the field of model-checking to optimization, as illustrated in the following sections with applications to parameter optimization and robustness evaluation in systems biology.

4.1. Continuous Satisfaction of QFLTL formulae

Let π be a trace and ϕ be a QFLTL($\mathbb{R}$) formula that we wish to valuate continuously in the interval $[0, 1]$. For this, we introduce a QFLTL($\mathbb{R}$) *pattern formula* $\psi(x_1, ..., x_k)$ obtained by replacing some constants in ϕ by new variables $\{x_1, ..., x_k\}$: we have $\phi = \psi(v_1, ..., v_k)$ for some instantiation of the variables by real values $v_1, ..., v_k$. The satisfaction degree of ϕ in π is then defined using the distance between the validity domain of the variables $x_1, ..., x_k$ in ψ and the objective values $(v_1, ..., v_k)$ in $\mathbb{R}^k$.

Definition 4.1 *The* violation degree *$vd(\pi, \phi, \psi)$ of a QFLTL formula ϕ in a numerical trace π with respect to a pattern formula $\psi(\boldsymbol{y})$ such that $\phi = \psi(\boldsymbol{v})$ for some real values $\boldsymbol{v}$, is the Euclidean distance between $\boldsymbol{v}$ and the projection on variables $\boldsymbol{y}$ of domain $(D_{\pi,\psi})$, or $+\infty$ if $D_{\pi,\psi} = \emptyset$:*

$$vd(\pi, \phi, \psi) = min_{\boldsymbol{v'} \in D^{\boldsymbol{y}}_{\pi,\psi}} d(\boldsymbol{v'}, \boldsymbol{v})$$

The satisfaction degree *of ϕ in π w.r.t. ψ is*

$$sd(\pi, \phi, \psi) = \frac{1}{1 + vd(\pi, \phi, \psi)}$$

Example 4.1 *Let us consider the LTL formula $\phi = F([A] > 20)$ specifying it was observed in experiments that after some time the concentration of compound A becomes greater than 20. Take $\psi = F([A] > x)$ as QFLTL pattern formula and $\mathbb{R}$ as variable space indicating that we are interested in looking at the current values reached by $[A]$. The objective value is $v = 20$.*

Now, given a mathematical model of our system, let us suppose that the QFLTL constraint solving algorithm applied to ψ on simulation trace π computes $D^x_{\pi,\psi} =]-\infty, 15]$ as domain for variable x. Since $v = 20$ we get $vd(\pi, \phi, \psi) = 5$, i.e. the violation degree is 5 since the compound reaches a maximum of 15 whereas the formula expresses that the threshold 20 is reached. The satisfaction degree of ϕ is thus $1/6$.

Example 4.2 *Consider the QFLTL formula $\phi = F([A] \geq x) \wedge F([A] \leq y) \wedge x - y \geq 10$. The possible values of v range from $-\infty$ to the maximum value M reached by $[A]$ in the trace. Similarly, w ranges from the minimum m of $[A]$ to ∞. Consequently, the quantity $x - y$ ranges from $-\infty$ to $M - m$, such that the constraint $x - y \geq 10$ constrains the maximal amplitude of variations of $[A]$ in the trace.*

The formula $\psi = F([A] \geq x) \wedge F([A] \leq y) \wedge x - y \geq amp$ will allow us to reason on the amplitude of these variations. Since amp is the only variable of interest (the only one instantiated from ψ to ϕ) it is the only one used to define the objective that the amplitude should be at least 10.

Now suppose that the constraint solving computes the following validity domains for x and y: $D^x_{\pi,\psi} =]-\infty, 15]$ and $D^y_{\pi,\psi} = [10, +\infty[$. For formula ϕ, the maximum value of $D^x_{\pi,\psi}$ represents the maximum value of $[A]$ and the minimum value of $D^y_{\pi,\psi}$ its minimum value in the trace. The domain for variable amp is $D^{amp}_{\pi,\psi} =]-\infty, 5]$ since we know that $amp \geq x - y$, and thus, since the objective value is 10, we obtain $vd(\pi, \phi, \psi) = 5$, i.e. the amplitude of the curve is 5 whereas we wanted it to be at least 10.

Note that if the trace π is such that ϕ is satisfied then $vd(\pi, \phi, \psi) = 0$ since the objective value belongs to the validity domain $D_{\pi,\psi}$. When ϕ is not valid on π, the violation degree vd provides a quantitative measure of its degree of non satisfaction. The use of this measure is illustrated in the following sections to improve parameter search for biological models and to define a quantitative notion of robustness of a system w.r.t. a temporal logic formula.

4.2. Abstraction of constants by variables in LTL Formulae

Since an LTL($\mathbb{R}$) formula is a special case of QFLTL($\mathbb{R}$) formula, containing no variable, the definitions of the previous section directly apply to it. However, as it is more usual to write specifications in LTL rather than QFLTL, we present here an abstraction α from LTL to QFLTL formulae providing a default choice for the pattern formula ψ associated to an LTL specification formula ϕ. This abstraction α thus provides a definition of the satisfaction degree of an LTL formula in a trace, without referring explicitly to any QFLTL pattern formula.

An LTL specification ϕ (of the expected behavior of a system), can be transformed in a QFLTL formula $\alpha(\phi)$ by mapping the occurrences of the constants $c_1, \ldots, c_k$ (i.e. real numbers corresponding to concentration thresholds, amplitudes, etc.) appearing in ϕ, to distinct variables $x_1, \ldots, x_k$. The reverse mapping $\sigma = \alpha^{-1}$ is an instantiation of the variables $x_1, \ldots, x_k$ to real values $v_1, \ldots, v_k$ such that $\phi = \sigma(\psi)$.

Example 4.3 *Consider again the formula ϕ of example 4.1: $\phi = F([A] > 20)$ we obtain $\alpha(\phi) = F([A] > x)$, which corresponds to the ψ that was chosen since the threshold is the only occurrence of a constant in ϕ.*

The same abstraction process can also be applied to QFLTL formulae, resulting in QFLTL formulae having more variables.

Example 4.4 *Consider the formulae of example 4.2*

We have $\phi = F([A] \geq v) \wedge F([A] \leq w) \wedge v - w \geq 10$ and $\psi = F([A] \geq v) \wedge F([A] \leq w) \wedge v - w \geq amp$.

ϕ is not an LTL formula but the above abstraction still provides $\alpha(\phi) = \psi$.

Obviously, such a systematic abstraction of constants by variables may produce non-linear constraints and needs be restricted in some cases. For instance in example S5 of section 5.2, some constants are kept in ψ_4 in order to fit a curve to given time points. The abstraction α may also be inadequate when one wants to rescale the weight between variables. For instance if $\phi = F([A] > 20) \wedge\ F([B] < 0.001)$, α will put the same weight on both distances, whereas a direct formulation with ψ would permit a rescaling, as in $\psi = F([A]/20 < x) \wedge\ F([B]/0.001 > y)$. For these reasons, the continuous valuation degree has been defined for QFLTL specifications w.r.t. a QFLTL pattern formula in a completely general form in the previous section.

5. Parameter Optimization w.r.t. QFLTL($\mathbb{R}$) Properties

The violation degree provides a measure of how far a given numerical trace is from satisfying a temporal logic specification. This continuous measure of satisfaction opens up the field of model-checking to optimization, and can be used to guide the search for models satisfying a given specification. In particular, in dynamical models of molecular interaction networks, the search of kinetic parameters w.r.t. LTL specifications by scanning the space of parameter values described in [7] with time complexity in $O(n^k)$ where n is the size of the trace and k the number of variables, can be replaced by a much more efficient local search method which explores neighborhoods of parameter sets in higher dimensions, and progress by choosing the ones which minimize the violation measure.

5.1. Optimization Method

Let us consider a (QF)LTL formula ϕ, a QFLTL formula ψ, such that $\phi = \sigma(\psi)$ as in section 4.1, an SBML/BIOCHAM reaction model with initial conditions and known parameter values, a set of unknown parameters to explore and for each of those an interval of search. We consider the problem of finding a set of values of the unknown parameters such that the violation degree of the corresponding trace π obtained by numerical simulation is $vd(\pi, \phi, \psi) = 0$.

A generic local search evolutionary method for parameter search can now be described as follows:

Algorithm 5.1 *1. Set the current point in the parameter space to a random point belonging to the provided search box, compute a numerical simulation with trace π and the corresponding violation degree $vd(\pi, \phi, \psi)$;*

2. if $vd = 0$ jump to 5.

3. for each point in a defined neighborhood of the current point, compute a trace and its violation degree;

4. based on the violation degrees of the neighbors, determine the next point of the iteration, set the current point to this point, update current vd and go to 2.

5. Return the current point in the parameter space.

This procedure can be interrupted after a given number of steps, returning the best parameter set (minimizing the violation degree). It can also be restarted with a new initial point (step 1) several times in order to diversify the search.

A naive method would be to define as neighborhood of the current parameter state the parameter sets obtained by modifying one parameter by values $\pm\delta$; and to choose as next parameter set the best neighbor. More efficient instances of the previous algorithm can be obtained however, by using a probabilistic definition of the neighborhood at each step, and storing the information in a covariance matrix. This is achieved by the state-of-the-art nonlinear optimization method of Hansen and Ostermeier [25] named the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). This method generalizes the approximate gradient and Hessian of a quasi-Newton method to an evolutionary algorithm for optimization problems with a black box fitness function on which no assumption is made. In our implementation in Biocham, we use CMA-ES with the violation degree of temporal logic formulae as fitness function.

CMA-ES performs a parameter search given an initial solution, stop and restart criteria and a given search space. The search stops either when a given number of violation degrees have been computed or when the violation degree gets below a given threshold. When differences between successive evaluation of the violation degree are below a given value, i.e. the violation degree landscape is too flat, the search restarts from a randomly chosen solution inside the search space.

The search space can be defined by intervals or by sets of linear constraints on the searched parameters. In this case, the Parma Polyhedra library [2] is used to compute the polytope representing the search space, check that it is not empty, and randomly pick a value in this polytope for restarts.

5.2. Evaluation on Cell Cycle Models

In this section we present the application of the parameter search method outlined above to the budding yeast cell cycle model of [9]. This model displays how proteins `Cdc2` and `cyclin` interact to form the heterodimer `Cdc2-Cyclin~{p1,p2}` known as maturation promoting factor (MPF) and playing a key role in the control of mitotic cycles. The model consists of the following BIOCHAM [21] reaction rules with mass action (MA) kinetics and an autocatalytic kinetics for the dephosphorylation of the mitosis promotion factor `Cdc2-Cyclin~{p1}`:

```
MA(k1)        for                     _ => Cyclin.
MA(k3)        for  Cyclin + Cdc2~{p1} => Cdc2-Cyclin~{p1,p2}
MA(k4p)       for Cdc2-Cyclin~{p1,p2} => Cdc2-Cyclin~{p1}
AUTOCAT(k4)   for Cdc2-Cyclin~{p1,p2} => Cdc2-Cyclin~{p1}
MA(k6)        for Cdc2-Cyclin~{p1}    => Cyclin~{p1} + Cdc2
MA(k7)        for         Cyclin~{p1} => _
MA(k8)        for                Cdc2 => Cdc2~{p1}
MA(k9)        for           Cdc2~{p1} => Cdc2
```

`MA(k)` denotes Mass Action law kinetics with parameter `k` while `~{p1}` and `~{p1,p2}` denote phosphorylated forms of a molecule. The rate of reaction 4 is described by:
`AUTOCAT(k4)= k4*[Cdc2-Cyclin~{p1,p2}]*[Cdc2-Cyclin~{p1}]^2.`

We use as reference point $\boldsymbol{k}_{Tyson}$ the values of the kinetic parameters determined in [9]. The simulation for $\boldsymbol{k}_{Tyson}$ of the system of ODEs extracted from these rules is displayed in Figure 1. The total amount of cyclin presents oscillations of period 35 while MPF exhibits activity peaks with same period.

Using the optimization method CMA-ES together with our violation degree as a parameter search method we wonder whether it is possible to find values of the kinetic parameters corresponding to higher MPF peaks or oscillations with higher amplitudes or shorter periods.

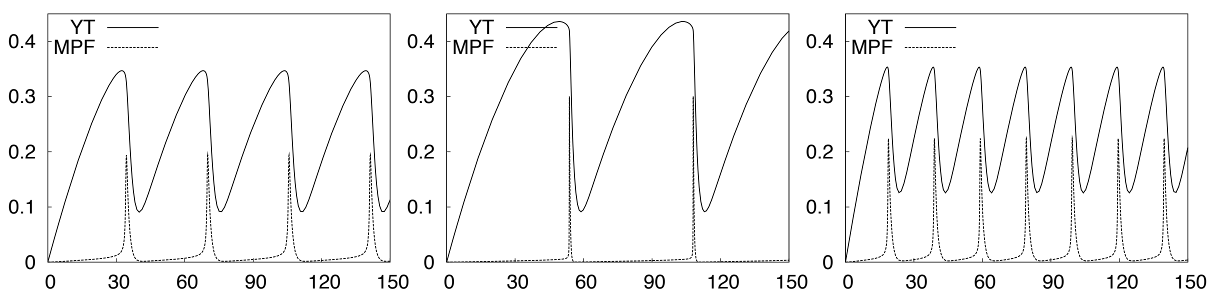


Figure 1. Dynamical behavior of the cell cycle model. The plots represent total cyclin (YT) and maturation promoting factor (MPF). (a) Oscillatory behavior obtained with parameter values $\boldsymbol{k}_{Tyson}$. (b) Higher MPF peaks obtained with $\boldsymbol{k}^*_{Tyson}$ (solution of problem S1). (c) Shorter oscillations period obtained with $\boldsymbol{k}^*_4$ (solution of problem S4).

5.2.1. Search problem S1 : higher MPF peaks (2 parameters unknown)

Two parameters, k4 and k6, have been found in [9] to play a particular role for the existence of oscillations. Depending on their values the system exhibits either a steady state behavior or limit cycle oscillations. We wonder whether it is possible to obtain higher MPF peaks by changing values of k4 and k6 only, all other parameters remaining at the value $\boldsymbol{k}_{Tyson}$ chosen in [9]. More precisely, we want to reach MPF peaks of 0.3 at least, the maximum amount of MPF for $\boldsymbol{k}_{Tyson}$ being 0.19.

Therefore we define the LTL specification : $\phi_1 = F([MPF] > 0.3)$ with the corresponding QFLTL formula being :

$$\psi_1 = F([MPF] > max)$$

The variable space associated to ϕ_1 and ψ_1 is $\mathbb{R}$ and corresponds to the sole variable max. The objective is $v = 0.3$, i.e the target peak value of MPF is 0.3. We have been able to find valid parameter values, denoted $\boldsymbol{k}^*_{Tyson}$, satisfying $vd(\pi, \phi_1, \psi_1) = 0$ where π is the corresponding simulated trace (see Figure 1b). $\boldsymbol{k}^*_{Tyson}$ is given in Table 3.

As the plot shows, for these parameter values essential features of the curve, especially repeated MPF peaks, are conserved although it was not enforced by the specification. In particular, a constantly growing amount of MPF would have also resulted in a null violation degree of this formula.

All computations have been performed on an Intel Core 2 Duo 2GHz with 2GB RAM. Note that as the optimization method CMA-ES uses a probabilistic neighborhood two consecutive runs can yield different results. In this example answers are typically obtained in less than 30 s after around 150 numerical simulations and violation degree computations.

5.2.2. Search problem S2 : amplitude of MPF oscillation (2 parameters unknown)

In this example we refine the previous query by constraining the minimum level of MPF. We search for k4 and k6 values that preserve at least two periods of MPF oscillations having same amplitudes as those observed for $\boldsymbol{k}_{Tyson}$.

In order to specify that the amplitude is at least 0.19, we use the variable space $\mathbb{R}$ corresponding to only one variable, amp, with objective value $v = 0.19$, i.e the target amplitude is 0.19. This value corresponds to the amplitude obtained for $\boldsymbol{k}_{Tyson}$.

$$\begin{aligned}
\phi_2 \;=\;& F([MPF] > max \wedge F([MPF] < min \\
&\quad \wedge F([MPF] > max \wedge F([MPF] < min)))) \\
&\wedge\, max - min > 0.19 \\
\psi_2 \;=\;& F([MPF] > max \wedge F([MPF] < min \\
&\quad \wedge F([MPF] > max \wedge F([MPF] < min)))) \\
&\wedge\, max - min > amp
\end{aligned}$$

Starting from a different value $\boldsymbol{k}_2$ for k4 and k6, we try to recover the behavior of $\boldsymbol{k}_{Tyson}$. We found such parameters, given in Table 3 and referred to as $\boldsymbol{k}^*_2$.

To illustrate the path followed during the search from $\boldsymbol{k}_2$ to $\boldsymbol{k}^*_2$ we computed the violation degree landscape in the k4, k6 parameter space. The resulting landscape is displayed in Figure 2. Note that as all constants of the formula have been abstracted by variables, the violation degree can only be finite. In particular when no oscillations are present in the trace *amp* will be equal to 0, thus leading to a violation degree of 0.19. Regions where the violation degree is 0.19 correspond to regions of steady state behavior whereas regions with a violation degree between 0 and 0.19 correspond to regions of oscillations.

Under mild assumptions Tyson determined linear equations defining a region in the k4, k6 plane where oscillations occurs, also represented in Figure 2. Our results are fully consistent with his analytical analysis, and provide more information on the amplitude of oscillation w.r.t. parameters k4 and k6.

5.2.3. Search problem S3 and S4 : amplitude and period of oscillations (all 8 parameters unknown)

To illustrate the scalability of the method we carry out two parameter searches on all 8 parameters of the model. The first one (problem S3) is the same query as above with formulae ϕ_2 and ψ_2 but with all parameters unknown.

The second one is a more complex query used to find shorter oscillation periods of `Cdc2`: to specify that the target period is 20, we use the variable space $\mathbb{R}$ corresponding to the variable *per* with objective value $v = 20$.

$$\begin{aligned}
\phi_3 \quad = \quad & F(d([\mathit{Cdc2}])/dt < 0 \wedge X(d([\mathit{Cdc2}])/dt > 0 \wedge \mathit{Time} > t1 \\
& \quad \wedge\ X(F(d([\mathit{Cdc2}])/dt > 0 \wedge X(d([\mathit{Cdc2}])/dt < 0 \wedge \mathit{Time} < t2)) \\
& \wedge\ t2 - t1 < 20 \\
\psi_3 \quad = \quad & F(d([\mathit{Cdc2}])/dt < 0 \wedge X(d([\mathit{Cdc2}])/dt > 0 \wedge \mathit{Time} > t1 \\
& \quad \wedge\ X(F(d([\mathit{Cdc2}])/dt > 0 \wedge X(d([\mathit{Cdc2}])/dt < 0 \wedge \mathit{Time} < t2)) \\
& \wedge\ t2 - t1 < \mathit{per}
\end{aligned}$$

Search problem S3 starts from parameter values $\boldsymbol{k_3}$ satisfying the constraints on their order of magnitude given in [9]. $\boldsymbol{k_3}$ does not give rise to oscillations. Search problem S4 starts from $\boldsymbol{k_4} = \boldsymbol{k_{Tyson}}$. In both cases parameter values are found satisfying the query (in around 350 s for S3 and 30 s for S4). Results are given in Table 3.

5.3. Evaluation on MAPK Signal Transduction Model

The MAPK signal transduction model of [31] is used to test the scalability of the parameter search method on a larger model, and illustrates the possibility of getting oscillations in a purely directional cascade of reactions [36,40,20]. The MAPK signaling network consists of a cascade of phosphorylation reactions from input membrane receptor activated molecule `RAF`, to output doubly phosphorylated molecule `MAPK~{p1,p2}` which enters the nucleus. The cascade of phosphorylations is depicted in Figure 3.

The model consists of the following reaction rules with mass action (MA) law kinetics with parameters $k1, \ldots, k_{15}$.

```
(MA(k1), MA(k2)) for RAF + RAFK <=> RAF-RAFK.
(MA(k3),MA(k4)) for RAF~{p1} + RAFPH <=> RAF~{p1}-RAFPH.
(MA(k5),MA(k6)) for MEK~$P + RAF~{p1} <=> MEK~$P-RAF~{p1}
   where p2 not in $P.
(MA(k7),MA(k8)) for MEKPH + MEK~{p1}~$P <=> MEK~{p1}~$P-MEKPH.
(MA(k9),MA(k10)) for MAPK~$P + MEK~{p1,p2} <=> MAPK~$P-MEK~{p1,p2}
```

	S1 (ϕ_1)		S2 (ϕ_2)		S3 (ϕ_2)		S4 (ϕ_3)	
	Initial values	Result	Initial values	Result	Initial values	Result	Initial values	Result
$vd(\pi, \phi_i, \psi_i)$	0.11	0	0.04	0	0.19	0	15.1	0
Parameters	$\boldsymbol{k}_{tyson}$	$\boldsymbol{k}^*_{tyson}$	$\boldsymbol{k}_2$	$\boldsymbol{k}^*_2$	$\boldsymbol{k}_3$	$\boldsymbol{k}^*_3$	$\boldsymbol{k}_4$	$\boldsymbol{k}^*_4$
k1	1.50e-2	1.50e-2	1.50e-2	1.50e-2	1.00e-2	**1.14e-2**	1.50e-2	**2.41e-2**
k3	2.00e2	2.00e2	2.00e2	2.00e2	1.00e2	**1.13e2**	2.00e2	**2.83e2**
k4p	1.80e-2	1.80e-2	1.80e-2	1.80e-2	1.00e-2	**8.77e-3**	1.80e-2	**2.24e-2**
k4	1.80e2	**8.99e2**	2.00e1	**1.94e2**	1.00e2	**1.82e2**	1.80e2	**2.28e2**
k6	1.00	**3.23**	0.25	**1.41**	1.00	**4.17e-1**	1	**1.13**
k7	0.60	0.60	0.60	0.60	1.00	**1.37**	0.60	**5.99e-1**
k8	1.00e2	1.00e2	1.00e2	1.00e2	1.00e3	**8.99e2**	1.00e2	**1.42e2**
k9	1.00e2	1.00e2	1.00e2	1.00e2	1.00e2	**8.44e1**	1.00e2	**6.94e1**
Total CPU time (s)	29.4		72.3		347.2		31.4	
Total simulation time (s)	26.1		17.3		56.3		12.5	
vd computation time (s)	2.7		49.7		271.2		16.8	
Number of *vd* evaluations	136		128		480		50	
Trace size	150		150		150		310	

Table 3

Parameter values and searching times for problems S1, S2, S3 and S4. In these search problems, the target specification is satisfiable and the procedure stops when the violation degree reaches 0. Computation times are the average values evaluated on 3 runs. Total simulation time is the total time spent for numerical simulations, while *vd* computation time is the total time spent to compute violation degrees.

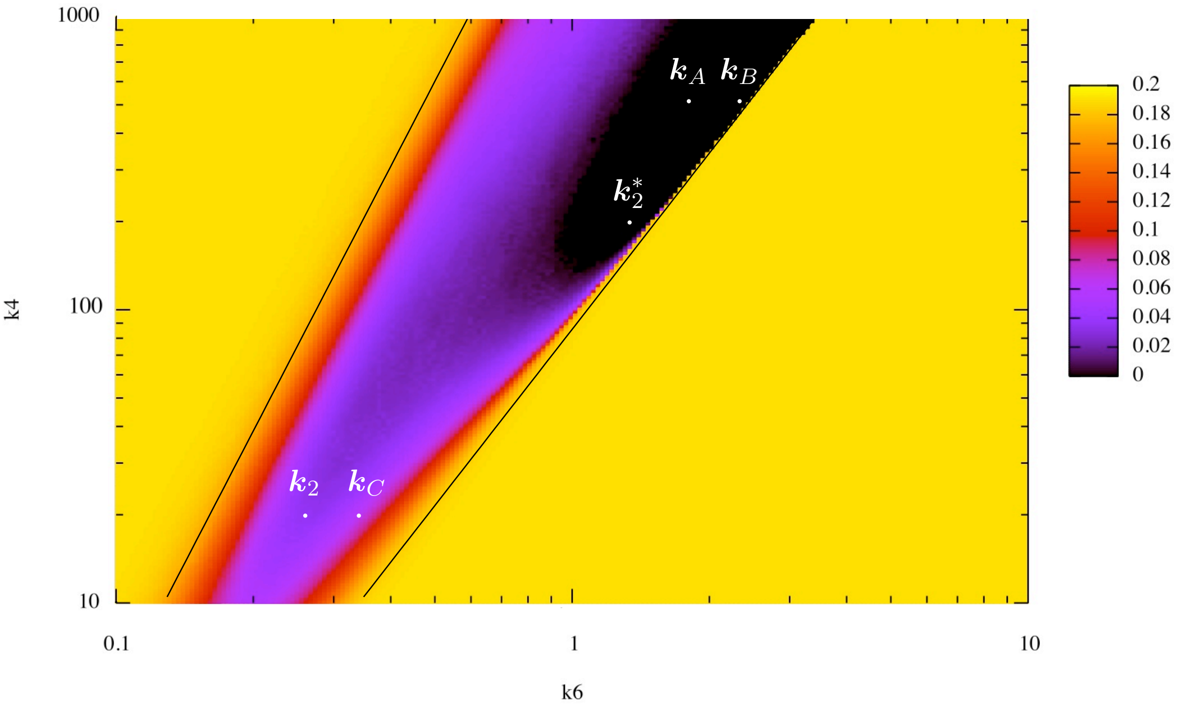


Figure 2. Violation degree landscape of problem S2. This violation degree measures amplitude of oscillations. Non oscillating regions have highest violation degree.

```
   where p2 not in $P.
(MA(k11),MA(k12)) for MAPKPH + MAPK~{p1}~$P <=> MAPK~{p1}~$P-MAPKPH.
MA(k13) for RAF-RAFK => RAFK + RAF~{p1}.
MA(k14) for RAF~{p1}-RAFPH => RAF + RAFPH.
MA(k15) for MEK~{p1}-RAF~{p1} => MEK~{p1,p2} + RAF~{p1}.
MA(k16) for MEK-RAF~{p1} => MEK~{p1} + RAF~{p1}.
MA(k17) for MEK~{p1}-MEKPH => MEK + MEKPH.
MA(k18) for MEK~{p1,p2}-MEKPH => MEK~{p1} + MEKPH.
MA(k19) for MAPK-MEK~{p1,p2} => MAPK~{p1} + MEK~{p1,p2}.
MA(k20) for MAPK~{p1}-MEK~{p1,p2} => MAPK~{p1,p2} + MEK~{p1,p2}.
MA(k21) for MAPK~{p1}-MAPKPH => MAPK + MAPKPH.
MA(k22) for MAPK~{p1,p2}-MAPKPH => MAPK~{p1} + MAPKPH.
```

We denote by k_{MAPK} the set of kinetic parameter values used as reference for this model.

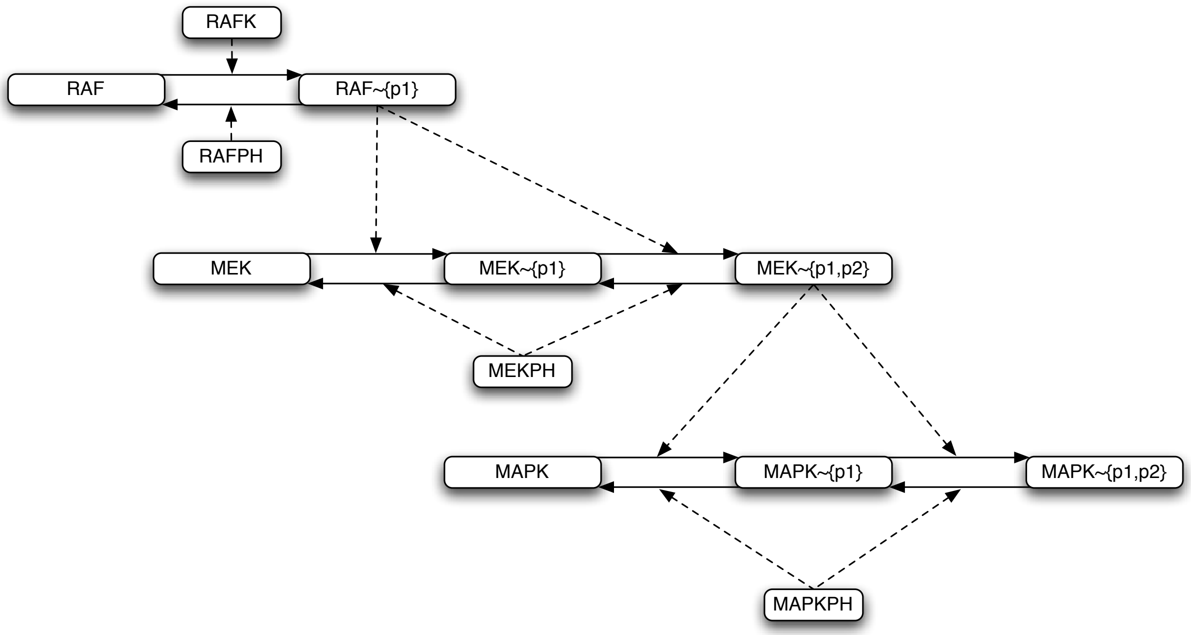


Figure 3. Diagram of the MAPK signal transduction cascade.

5.3.1. Search problem S5 : curve fitting at specific time points (22 parameters unknown)

In this example, we investigate the use of our parameter search method as a curve fitting tool at specific time points, on 22 parameter values. This is a typical scenario for fitting a model to experimental data. In order to express the classical distance between two curves at time points 30 and 60 for instance, we use the following pattern of formulae :

$$\begin{aligned}\psi_4 = G(\quad & \mathit{Time} = 30 \rightarrow [\mathit{MEK\text{-}RAF}\tilde{}\{p1\}] = x \\ & \wedge\ \mathit{Time} = 60 \rightarrow [\mathit{MEK\text{-}RAF}\tilde{}\{p1\}] = y)\end{aligned}$$

The parameter space of this formula is $\mathbb{R}^2$ and is defined by the two variables x and y. We set the objective $\boldsymbol{v}$ to the target values of $[\mathit{MEK\text{-}RAF}\tilde{}\{p1\}]$ at time 30 and 60, which defines ϕ_4.

Note that in theory, if the time points given in this formula are not present in the numerical simulated trace of the model the formula will always be true, resulting in a null violation degree. In order to effectively use this formula for curve fitting, we thus assume that the numerical simulations contain these specific time points. This can be done by computing concentration values at these time points by linear interpolation of neighboring values. Besides, this formula can be extended to any number of time points and molecules in order to perform a complete curve fitting, if it is relevant.

This pattern of formulae can be used to search the values of all the 22 parameters of the model to fit the concentration $[\mathit{MEK\text{-}RAF}\tilde{}\{p1\}]$ at six time points. The objective values for these time points are the values of the original model, obtained by simulation with the original parameters k_{MAPK}. The initial values for the search are some random altered values $k_{MAPK_{alt}}$. Numerical simulations obtained with k_{MAPK}, $k_{MAPK_{alt}}$ and the resulting parameter values are given in Figure 4. It took 290 s to obtain the result.

Note that a search problem with such dimensionality cannot be solved with parameter scanning, which has exponential time complexity in the number of parameters, and which is what can be hardly done with a binary valuation of temporal logic specifications [7]. By allowing continuous valuation of temporal logic formulae, scalability can be achieved by using state-of-the-art method for non-linear optimization for which the computation time depends on the type of problem (formula used and initial values of the parameters) and on the landscape of the violation degrees rather than on the number of parameters.. In our work we use CMA-ES [25] which has been shown to be applicable to searching problems of up to one or two hundred parameters on single processor machines.

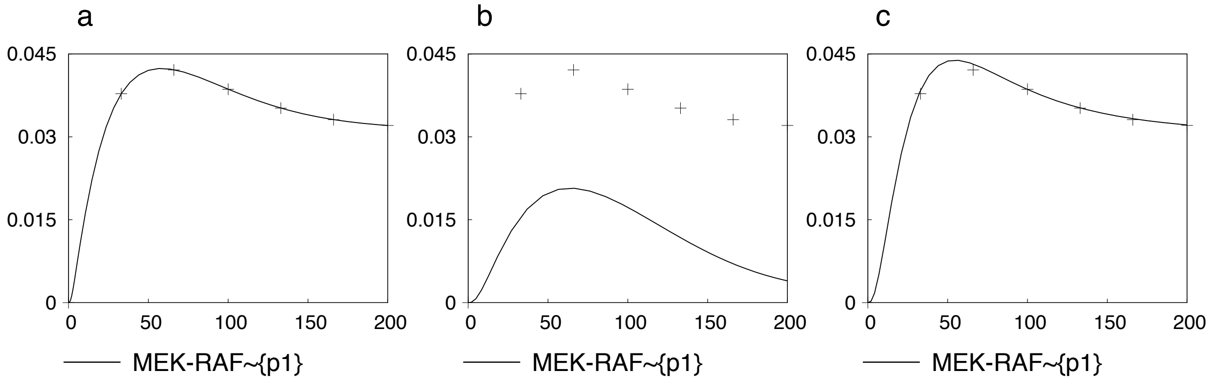


Figure 4. Dynamical behavior of the MAPK model. The curves display $[MEK\text{-}RAF\tilde{\ }\{p1\}]$. (a) Reference curve obtained with k_{MAPK} (b) Simulated curve obtained with altered parameter values $k_{MAPK_{alt}}$. Points are the reference values taken from curve (a). (c) Simulated curve obtained after curve fitting (solution of problem S5).

5.3.2. Search problem S6 : find oscillations (30 kinetic parameters and 7 initial conditions unknown)

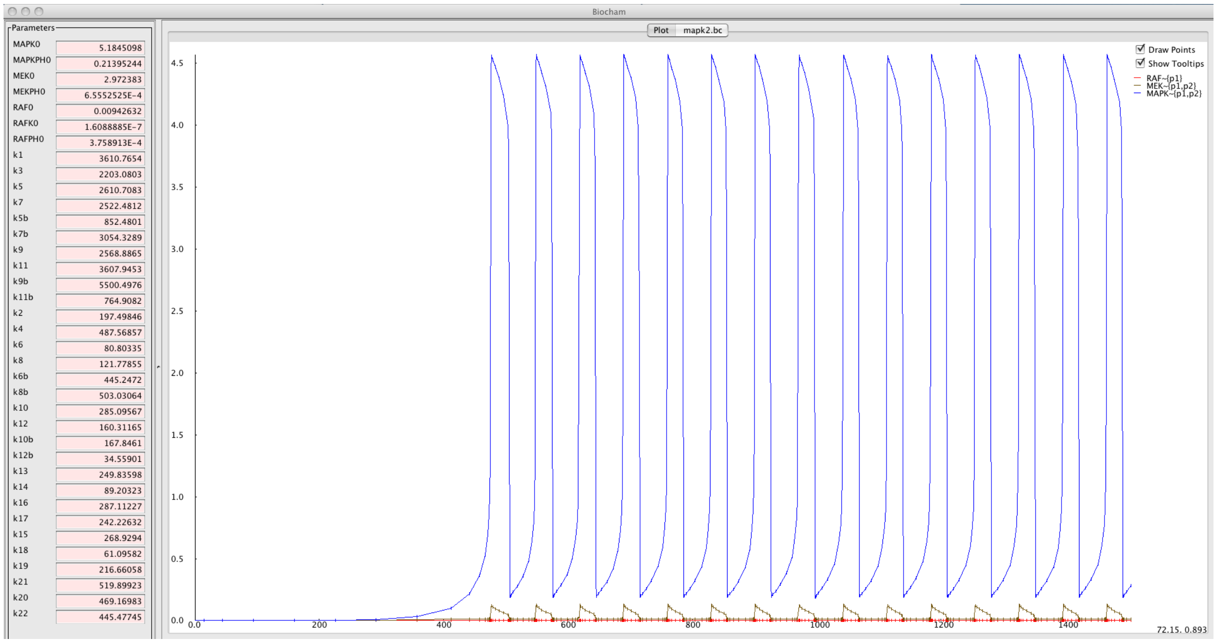
In [36], oscillations have been found in the MAPK cascade model of [31] although this model does not contain any negative feedback reaction. This does not contradict Thomas' necessary condition for sustained oscillations as such a purely directional cascade does contain negative feedback in its influence graph as shown in [20] and analyzed in [40]. However, to know whether these negative circuits in the influence graph are functional, one needs to search for kinetic parameter values and initial conditions that exhibit sustained oscillations.

Just by using the following formulae:

$$\phi_5 = F([MAPK\tilde{\ }\{p1,p2\}] > max \wedge F([MAPK\tilde{\ }\{p1,p2\}] < min)) \wedge \ max - min > 0.5$$

$$\psi_5 = F([MAPK\tilde{\ }\{p1,p2\}] > max \wedge F([MAPK\tilde{\ }\{p1,p2\}] < min)) \wedge \ max - min > amp$$

with the single variable $amp < max - min$ over $\mathbb{R}$, and by asking that setting an objective value $v = 0.5$ for this amplitude variable, parameter values leading to sustained oscillations are found in a few minutes, such as for instance the ones depicted in Figure 5.



Parameters

MAPK0	5.1845098
MAPKPH0	0.21395244
MEK0	2.972383
MEKPH0	6.5552525E-4
RAF0	0.00942632
RAFK0	1.6088885E-7
RAFPH0	3.758913E-4
k1	3610.7654
k3	2203.0803
k5	2610.7083
k7	2522.4812
k5b	852.4801
k7b	3054.3289
k9	2568.8865
k11	3607.9453
k9b	5500.4976
k11b	764.9082
k2	197.49846
k4	487.56857
k6	80.80335
k8	121.77855
k6b	445.2472
k8b	503.03064
k10	285.09567
k12	160.31165
k10b	167.8461
k12b	34.55901
k13	249.83598
k14	89.20323
k16	287.11227
k17	242.22632
k15	268.9294
k18	61.09582
k19	216.66058
k21	519.89923
k20	469.16983
k22	445.47745

Figure 5. Oscillations of MAPK found by parameter search in BIOCHAM

6. Robustness Measures w.r.t. QFLTL($\mathbb{R}$) Properties

6.1. Principle

The notion of satisfaction degree allows us to define in a mathematically precise way a degree of robustness of a system's behavior property described in temporal logic, with respect to a set of system's perturbations, and estimate it computationally. This robustness degree is defined as the average satisfaction degree of the property of interest over all admissible perturbations, possibly weighted by their probabilities:

Definition 6.1 *Let P be a set of perturbations, $prob(p)$ be the probability of perturbation p, $\pi(p)$ be the timed trace of the system under perturbation $p \in P$. The* robustness degree $R_{\phi,\psi,P}$ *of properties ϕ and ψ with respect to perturbation P is the real value*

$$R_{\phi,\psi,P} = \int_{p \in P} sd(\pi(p), \phi, \psi) \; prob(p) \; dp$$

If the set of perturbations is finite (eg, gene knock outs), the robustness degree is simply a finite weighted sum and can be exactly computed. If the set of perturbations is infinite, the robustness degree can be estimated by computing the violation degree between the behavior of the perturbed system $\pi(p)$ and the specification ϕ for sufficiently many perturbations.

	S5 (ϕ_4)		S6 (ϕ_5)	
	Initial value	Result	Initial value	Result
$vd(\pi, \phi_i, \psi_i)$	0.06	0.0025	0.5	0
Total CPU time (s)	290		372.4	
Total simulation time (s)	263.9		229.6	
vd computation time (s)	8.1		124.9	
Number of vd evaluations	906		1092	
Trace size	36		300	

Table 4
Total CPU time, simulation time and violation degree computation time (s) for search problems S5 and S6. As the violation degree associated to curve fitting problem S5 is always strictly positive, the search is stopped as soon as vd gets below arbitrary threshold value 0.005. S6 stops when the violation degree gets to 0.

This definition is an adaptation of the general definition given by Kitano [28] to our temporal logic setting. It is described in more detail with applications to synthetic genetic networks and sensitivity analyses in [37].

6.2. Evaluation on Cell Cycle Model

Using the same cell cycle model as in section 5.2, we compare the robustness of oscillation properties with regard to perturbations of parameter values k4 and k6 for different points in the parameter space.

We consider that parameter values for k4 and k6 are normally distributed around their reference value with coefficient of variation equal to 0.2. We also enforce that $k4 \geq 0 \wedge k6 \geq 0$. We examine the robustness of the property expressed by ϕ_2 and ψ_2, that is, MPF oscillations are of amplitude at least 0.19.

The robustness degree of this property is compared for three different values of k4 and k6. These three points in the parameter space of k4 and k6 are indicated by the three points $\boldsymbol{k}_A$, $\boldsymbol{k}_B$ and $\boldsymbol{k}_C$ in Figure 2. In all cases, the estimation of the robustness degree is done by computing the mean value of the violation degree for 500 samples.

The estimated degree of robustness for parameters $\boldsymbol{k}_A$, $\boldsymbol{k}_B$ and $\boldsymbol{k}_C$ are respectively 0.991, 0.917 and 0.932. This is consistent with the location of points $\boldsymbol{k}_A$, $\boldsymbol{k}_B$ and $\boldsymbol{k}_C$. Perturbations around point $\boldsymbol{k}_A$ have high probabilities of staying in the region satisfying the specification whereas perturbations around point $\boldsymbol{k}_B$ have high probabilities of moving the system to the region with no oscillation. $\boldsymbol{k}_C$ is more robust than $\boldsymbol{k}_B$ even though, as opposed to $\boldsymbol{k}_B$, its violation degree is non null. This is explained by the abrupt transition between oscillating and non oscillating regions near $\boldsymbol{k}_B$ compared to the smoother transition near $\boldsymbol{k}_C$.

The robustness degree can be estimated for perturbations on any number of parameters. For instance, by computing a robustness estimate for perturbations on all parameters, with coefficient of variation 0.2 for specification ϕ_2, ψ_2 and parameter values $\boldsymbol{k}_{Tyson}$ and $\boldsymbol{k}_3^*$, the estimated robustness degrees for $\boldsymbol{k}_{Tyson}$ and $\boldsymbol{k}_3^*$ are 0.962 and 0.970 respectively. This indicates that the oscillations are more robust to variations of the parameters values for $\boldsymbol{k}_3^*$ than for the parameters given in the original model of Tyson.

7. Related Work

Probabilistic temporal logics and probabilistic model checking have been used in systems biology [30], e.g. for an analysis of a probabilistic model of the MAP kinase signaling cascade. However these techniques provide information on the probability that a given property is exactly satisfied. They thus provide no quantitative information on unsatisfied formulae and cannot be compared to the satisfaction degree presented in this paper.

More closely related to our continuous satisfaction degree are the linear metrics for quantitative transition systems defined in [13]. These metrics apply to traces and can be characterized by quantitative LTL formulae. Quantitative LTL formulae include quantitative versions of the temporal operators and logical connectives, but not of the atomic propositions, and are interpreted on the $[0, 1]$ interval. Compared to QFLTL, these quantitative LTL formulae are thus limited to the time variable, and the applicability of this approach to solving optimization and robustness problems is not discussed.

Fainekos and Pappas have also proposed a satisfaction degree for temporal logic specifications [22]. Although the two approaches share some similarities, a significant difference is that in [22] the satisfaction degree corresponds to a distance between a trace and the set of traces satisfying a formula, whereas in our case the violation degree corresponds to a distance between a formula and the set of formulae satisfied by the trace. One major advantage of our approach is that the dimensionality of the formula space (number of constants abstracted by variables) is generally much lower than the dimensionality of the trace space (trace length). Reasoning in low dimensional space may strongly affect the practical applicability of these methods. Note however both approaches, handling sets of traces [23,11], and our approach, handling sets of formulae, are *a priori* compatible, and that their combination might combine their benefits.

Another notion of violation degree has been recently proposed by Donaldson and Gilbert [14], also based on the definition of a validity domain of temporal logic formulae. However, the computation of the (finite) validity domain is made by sampling the formula space rather than by constraint solving.

Concerning robustness, in [12], Chaves and colleagues propose a quantitative measure of robustness corresponding to the volume of the set of valid parameters in the parameter space. This measure thus reflects the proportion of parameters that satisfy exactly the property, as opposed to our measure that represents how close to satisfying the property the system is for various parameters. These two measures provide complementary information on robustness. In [4], robustness is similarly defined with respect to temporal logic specifications. However, it has a Boolean interpretation, since a property is defined as robustly satisfied by an ODE system if it is satisfied by the system for all possible perturbations. As stated earlier, obtaining a quantitative measure of robustness is more informative for many practical problems.

8. Conclusion

We have defined a continuous measure of satisfaction of an LTL($\mathbb{R}$) formula in a numerical trace and shown that it can be computed using the QFLTL($\mathbb{R}$) constraint solving algorithm of [17]. This measure is more informative than the Boolean interpretation of the formulae and can be used in many situations in systems biology to reason about numerical traces.

We have shown that this measure can be used as a fitness function in state-of-the-art optimization tools to efficiently guide the search of kinetic parameter values in biochemical reaction models – a central problem of computational systems biology. Several tenths of parameters could be found at the same time to satisfy non-trivial semi-qualitative semi-quantitative properties formalized in QFLTL($\mathbb{R}$).

We have also shown that this satisfaction measure can similarly be used to estimate the robustness of a model w.r.t. temporal logic specifications, in accordance to Kitano's notion of robustness for systems biology.

The generalization of model-checking to temporal logic constraint solving which is at the basis of our computation method thus seems to open new research avenues for the use of temporal logics in systems biology, and more generally for design problems from temporal specifications. In [18], the theory of temporal logic constraint solving presented here is generalized to branching structures and Computation Tree Logic (CTL) formulae over an arbitrary constraint domain.

Acknowledgements

This work is partly supported by the EU FP6 STREP project TEMPO and by the INRA Agrobi project INSIGHT. We acknowledge fruitful discussions with our partners of these projects.

REFERENCES

1. M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Model building and model checking for biochemical processes. *Cell Biochemistry and Biophysics*, 38:271–286, 2003.
2. R. Bagnara, P. M. Hill, and E. Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 72(1–2):3–21, 2008.
3. G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider. Validation of qualitative models of genetic regulatory networks by model checking : Analysis of the nutritional stress response in *Escherichia coli*. *Bioinformatics*, 21(Suppl.1):i19–i28, 2005.
4. G. Batt, B. Yordanov, R. Weiss, and C. Belta. Robustness analysis and tuning of synthetic gene networks. *Bioinformatics*, 23(18):2415–2422, 2007.
5. G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. A fruitful application of formal methods to biological regulatory networks: Extending thomas' asynchronous logical approach with temporal logic. *Journal of Theoretical Biology*, 229(3):339–347, 2004.
6. M. Calder, V. Vyshemirsky, D. Gilbert, and R. Orton. Analysis of signalling pathways using the continuous time markov chains. In G. Plotkin, editor, *Transactions on Computational Systems Biology VI*, volume 4220 of *Lecture Notes in BioInformatics*, pages 44–67. Springer-Verlag, Nov. 2006. CMSB'05 Special Issue.
7. L. Calzone, N. Chabrier-Rivier, F. Fages, and S. Soliman. Machine learning biochemical networks from temporal logic properties. In G. Plotkin, editor, *Transactions on Computational Systems Biology VI*, volume 4220 of *Lecture Notes in BioInformatics*, pages 68–94. Springer-Verlag, Nov. 2006. CMSB'05 Special Issue.
8. N. Chabrier and F. Fages. Symbolic model checking of biochemical networks. In C. Priami, editor, *CMSB'03: Proceedings of the first workshop on Computational Methods in Systems Biology*, volume 2602 of *Lecture Notes in Computer Science*, pages 149–162, Rovereto, Italy, Mar. 2003. Springer-Verlag.
9. K. C. Chen, A. Csikász-Nagy, B. Györffy, J. Val, B. Novàk, and J. J. Tyson. Kinetic

analysis of a molecular model of the budding yeast cell cycle. *Molecular Biology of the Cell*, 11:396–391, 2000.

10. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 1999.
11. T. Dang, A. Donze, O. Maler, and N. Shalev. Sensitive state space exploration. In *Proceedings of the 47th IEEE Conference on Decision and Control, CDC 2008*, pages 4049–4054, 2008.
12. A. Dayarian, M. Chaves, E. Sontag, and A. Sengupta. Shape, size and robustness: Feasible regions in the parameter space of biochemical networks. *PLoS Computational Biology*, 5(1):e1000256, Jan. 2009.
13. L. de Alfaro, M. Faella, and M. Stoelinga. Linear and branching metrics for quantitative transition systems. In J. Diaz, J. Karhumäki, A. Lepistö, and D. Sannella, editors, *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, volume 3142 of *Lecture Notes in Computer Science*, pages 97–109. Springer, 2004.
14. R. Donaldson and D. Gilbert. A model checking approach to the parameter estimation of biochemical pathways. In M. Heiner and A. Uhrmacher, editors, *CMSB'08: Proceedings of the fourth international conference on Computational Methods in Systems Biology*, volume 5307 of *Lecture Notes in Computer Science*, pages 269–287. Springer-Verlag, Oct. 2008.
15. S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and M. K. Sönmez. Pathway logic: Symbolic analysis of biological signaling. In *Proceedings of the seventh Pacific Symposium on Biocomputing*, pages 400–412, Jan. 2002.
16. E. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Sematics, pages 995–1072. MIT Press, 1990.
17. F. Fages and A. Rizk. On temporal logic constraint solving for the analysis of numerical data time series. *Theoretical Computer Science*, 408(1):55–65, Nov. 2008.
18. F. Fages and A. Rizk. From model-checking to temporal logic constraint solving. In *Proceedings of CP'2009, 15th International Conference on Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science. Springer-Verlag, Sept. 2009.
19. F. Fages and S. Soliman. Formal cell biology in BIOCHAM. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *8th Int. School on Formal Methods for the Design of Computer, Communication and Software Systems: Computational Systems Biology SFM'08*, volume 5016 of *Lecture Notes in Computer Science*, pages 54–80, Bertinoro, Italy, Feb. 2008. Springer-Verlag.
20. F. Fages and S. Soliman. From reaction models to influence graphs and back: a theorem. In *Proceedings of Formal Methods in Systems Biology FMSB'08*, number 5054 in Lecture Notes in Computer Science. Springer-Verlag, Feb. 2008.
21. F. Fages, S. Soliman, and A. Rizk. *BIOCHAM v8 user's manual*. INRIA, 2009. `http://contraintes.inria.fr/BIOCHAM`.
22. G. Fainekos and G. Pappas. Robustness of temporal logic specifications. In K. Havelund, M. Núñez, G. Rosu, and B. Wolff, editors, *Proceedings of the International Workshop on Formal Approaches to Software Testing and Runtime Verification, FATES/RV'06*, volume 4262 of *Lecture Notes in Computer Science*, pages 178–192. Springer, 2006.
23. G. Fainekos and G. Pappas. Robust sampling for MITL specifications. In J.-F. Raskin and P. Thiagarajan, editors, *Proceedings of the Fifth International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS'07*, volume 4763 of *Lecture Notes in Computer Science*, pages 147–162. Springer, 2007.
24. D. Gilbert, M. Heiner, and S. Lehrack. A unifying framework for modelling and analysing

biochemical pathways using Petri nets. In *CMSB'07: Proceedings of the fifth international conference on Computational Methods in Systems Biology*, volume 4695 of *Lecture Notes in Computer Science*. Springer-Verlag, 2007.

25. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
26. J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. In *Proc. Computational Methods in Systems Biology (CMSB'06)*, volume 4210 of *Lecture Notes in Computer Science*, pages 32–47. Springer-Verlag, 2006.
27. M. Hucka et al. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.
28. H. Kitano. Towards a theory of biological robustness. *Molecular Systems Biology*, 3:137, 2007.
29. K. W. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular Biology of the Cell*, 10(8):2703–2734, Aug. 1999.
30. M. Kwiatkowska, G. Norman, and D. Parker. Using probabilistic model checking in systems biology. *SIGMETRICS Performance Evaluation Review*, 35(4):14–21, 2008.
31. A. Levchenko, J. Bruck, and P. W. Sternberg. Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. *PNAS*, 97(11):5818–5823, May 2000.
32. H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano. Hybrid Petri net representation of gene regulatory network. In *Proceedings of the 5th Pacific Symposium on Biocomputing*, pages 338–349, 2000.
33. D. Nickovic and O. Maler. Amt: a property-based monitoring tool for analog systems. In *Proceedings of 5th International Conference on Formal Modelling and Analysis of Times Systems, FORMATS'07*, Lecture Notes in Computer Science, Salzburg, Austria, 2007. Springer-Verlag.
34. A. Phillips and L. Cardelli. A correct abstract machine for the stochastic pi-calculus. *Transactions on Computational Systems Biology*, to appear. Special issue of BioConcur 2004.
35. C. Priami, A. Regev, W. Silverman, and E. Shapiro. Application of a stochastic name passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, 2001.
36. L. Qiao, R. B. Nachbar, I. G. Kevrekidis, and S. Y. Shvartsman. Bistability and oscillations in the huang-ferrell model of mapk signaling. *PLoS Computational Biology*, 3(9):1819–1826, Sept. 2007.
37. A. Rizk, G. Batt, F. Fages, and S. Soliman. A general computational method for robustness analysis with applications to synthetic gene networks. *BioInformatics*, July 2009. to appear.
38. L. A. Segel. *Modeling dynamic phenomena in molecular and cellular biology.* Cambridge University Press, 1984.
39. Z. Szallasi, J. Stelling, and V. Periwal, editors. *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts.* MIT Press, 2006.
40. A. C. Ventura, J.-A. Sepulchre, and S. D. Merajver. A hidden feedback in signaling cascades is revealed. *PLoS Computational Biology*, 4(3):e1000041, Mar. 2008.