Novel domain expansion methods to improve the computational efficiency of the solution of Chemical Master Equation for large biological networks.

Rahul Kosarwal
Lincoln University
Don Kulasiri (don.kulasiri@lincoln.ac.nz)
Lincoln University, New Zealand
Sandhya Samarasinghe
Lincoln University

Research article

Keywords: Biochemical reaction network, chemical master equation, stochastic, intelligent state projection, Bayesian likelihood node projection, mathematical modelling, ordinary differential equations

Posted Date: April 28th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-23887/v1

Version of Record: A version of this preprint was published on November 11th, 2020. See the published version at https://doi.org/10.1186/s12859-020-03668-2.

Novel domain expansion methods to improve the computational efficiency of the solution of Chemical Master Equation for large biological networks.

Authors: Rahul Kosarwal[1], Don Kulasiri[*2], Sandhya Samarasinghe[3]

Abstract

Background: Numerical solutions of the chemical master equation (CME) are important to understand the stochasticity of biochemical systems. However, solving CMEs is a formidable task due to the nonlinear nature of the reactions and size of the networks that result in different realisations and, most importantly, the exponential growth of the size of the state-space with respect to the number of different species in the system. When the size of the biochemical system is very large in terms of the number of variables, the solution to the CME becomes intractable. Therefore, we introduce the intelligent state projection (ISP) method to use in the stochastic analysis of these systems. For any biochemical reaction network, it is important to capture more than one moment to describe the dynamic behaviour of the system. ISP is based on a state-space search and the data structure standards of artificial intelligence (AI) to explore and update the states of a biochemical system. To support the expansion in ISP, we also develop a Bayesian likelihood node projection ($BLNP$) function to predict the likelihood of the states.

Results: To show the acceptability and effectiveness of our method, we apply the ISP to several biological models previously discussed in the literature. According to the results of our computational experiments, we show that ISP is effective in terms of speed and accuracy of the expansion, accuracy of the solution, and provides a better understanding of the state-space of the system in terms of blueprint patterns.

Conclusions: The ISP is the de-novo method to address the accuracy as well as the performance problems for the solution of the CME. It systematically expands the projection space based on predefined inputs, which are useful in providing accuracy in the approximation and an exact analytical solution at the time of interest. The ISP was more effective in terms of predicting the behaviour of the state-space of the system and in performance management, which is a vital step towards modelling large biochemical systems.

Keywords: Biochemical reaction network, chemical master equation, stochastic, intelligent state projection, Bayesian likelihood node projection, mathematical modelling, ordinary differential equations.

Background

In systems biology, it is of great interest to understand the dynamics of large and complicated biochemical reaction networks based on recent advances in computing and mathematical techniques. These advances in techniques have made it easier for biologist to deal with enormous amounts of experimental data down to the level of a single molecule of a species. Such information reveals the presence of a high level of stochasticity in the networks of biochemical reactions. Stochastic models in biochemical reaction networks have made significant contributions towards the fields of systems biology [1,2], neuroscience [3], and drug modelling [4].

Biochemical reactions in a complex system are often modelled as reaction rate equations (RREs) using the ordinary differential equations (ODEs) such as in the biochemical networks of Alzheimer's disease (AD) [5]; in the pathways in the fungal pathogen *Candida albicans* [6]; in the COVID-19 coronavirus pathogen network [7], where the behaviour of different pathogens in pathways is still largely unknown. But all these models contain the species with small copy numbers and widely different reaction rates; therefore, the probabilistic descriptions of time evolution of molecular concentrations (or numbers) are much suited to understand the dynamics of such systems. One probabilistic approach to model a biochemical reaction network is to deduce a set of integro-differential equations known as chemical master equations (CMEs) [8,9]. CMEs describe the evolution of the probability distribution over the entire state-space of a biochemical system that jumps from one set of states to another set of states in continuous time: they are a continuous time version of Markov chains (CTMCs) [8,10] with discrete states. By defining the Markov chain [10,11], we can consider the joint and marginal probability densities of the species in a system that change over time [12].

In such cases, the inability to define the nonlinearity at the molecule count level in $RREs$ can become very important. The biochemical reaction network can be defined in terms of the discrete state $X \equiv (x_1, \dots, x_{\tilde{\mathrm{N}}})^T$ vector of non-negative integers $x_{\tilde{\mathrm{N}}}$ for the given the initial conditions, where $\tilde{\mathrm{N}} \geq 1$. $\{X(t) : t \in K;\ \varphi\}$ defines a stochastic process, where K is some indexing scheme and φ is a sample space. Following the derivation in [9], for every reaction, there exists a reaction channel, R_M, which determines a unique reaction in the system with a propensity function k_M. The specific combinations of the reactant species in R_M will react during an infinitesimal $[t, t + dt)$ time interval. The average probability $a_{\mu}(X(t))dt$ of a particular R_M fires within $[t, t + dt)$ is the multiplication of the numbers of reactant species, denoted by square brackets, by k_M. For example,

$$\begin{aligned} &\boldsymbol{R_1}\colon\ C \overset{k_1}{\rightarrow} A, && \boldsymbol{a_1}\colon\ k_1([C]) \\ &\boldsymbol{R_2}\colon\ A + E \overset{k_2}{\rightarrow} T, && \boldsymbol{a_2}\colon\ k_2([A].[E]) \\ &\boldsymbol{R_3}\colon\ T \overset{k_3}{\rightarrow} C, && \boldsymbol{a_3}\colon\ k_3([T]) \end{aligned}$$

In the case when the reactants are of the same type; for example $A + A \overset{k_2}{\rightarrow} T$, then $a_2\colon\ k_2(\frac{[A][(A-1)]}{2})$. The set consisting of all the reaction channels, R_M is the union of sets of *fast* reactions and *slow* reactions, and they are categorised into sets of $R_{M(fs)}$ and $R_{M(sr)}$ reactions, respectively, based on their propensity values. Therefore,

$$R_M = R_{M(fs)} \bigcup R_{M(sr)}. \tag{1}$$

A reaction is faster than others if its propensity is of several orders of magnitude larger than the other propensity values. (See the list of abbreviations and notations at the end.)

Chemical Master Equation

In this paper, we consider a network of biochemical reactions at a constant volume that consists of $\tilde{\mathrm{N}} \geq 1$ different species $\{S_1, \dots S_{\tilde{\mathrm{N}}}\}$ that are spatially homogeneous and interact through $M \geq 1$ reaction channels in thermal equilibrium. The number of counts of each different species defines the state of the system. If all the species are bounded by S, then the

approximate number of states in the system would be $S^{\tilde{N}}$ [13]. Each state $X \equiv (x_1, \dots x_{\tilde{N}})^T$, and $x_{\tilde{N}}$, denotes the number of molecules (counts) of each species. For every state, X, the probability satisfies the following CME [8],

$$\frac{\partial P^{(t)}(X)}{\partial t} = \sum_{\mu=1}^{M} a_{\mu}(X - v_{\mu})\, P^{(t)}(X - v_{\mu}) - \sum_{\mu=1}^{M} a_{\mu}(X)\, P^{(t)}(X) \tag{2}$$

where $P^{(t)}(X)$ = the probability function representing the time-evolution of the system, given that $t \geq t_0$ and the initial probability is, $P^{(t_0)}(X_0)$,

M = elementary chemical reaction channels R_1,..... R_M,

a_{μ} = chemical reaction propensity of channel $\mu = \{1,2, \dots . M\}$, and

v_{μ} = stoichiometric vector that represents a change in the molecular population of the chemical species by the occurrence of one R_M reaction. The system transitions to a new state by $X + v_{\mu}$ recording the changes in the number of counts of different species when the reactions occur.

We note that $a_{\mu}(X - v_{\mu})dt$ is the probability for state $(X - v_{\mu})$ to transition to state X through chemical reaction, R_M, during $[t, t + dt)$, and $\sum_{\mu=1}^{M} a_{\mu}(X)dt$ is the probability for the system to shift from state X through any reaction during dt. Let $\mathbf{X}_J = \{X_1, \dots \dots . X_{S^{\tilde{N}}}\}$ be the ordered set of possible states of the system indexed by $\{1,2 \dots K\}$ having $S^{\tilde{N}}$ elements, then Eq. (2) represents the set of ordinary differential equations (ODEs) that determines the changes in probability density $P^{(t)}$= $(P^{(t)}(X_1), \dots P^{(t)}(X_{S^{\tilde{N}}}))^T$. Once $\mathbf{X}_J$ is selected, the matrix-vector form of Eq. (2) is described by an ODE:

$$\frac{\partial P^{(t)}}{\partial t} = A.P^{(t)}, \tag{3}$$

where the transition rate matrix is $A = [a_{i,j}]$. If each reaction leads to a different state, $X_{i'}$, then the elements in submatrix $A_{i,j}$ are given as:

$$A_{i,j} = \begin{cases} -\sum_{\mu=1}^{M} a_{\mu}(X_i), & \text{if } i = j \\ a_{\mu}(X_i), & \text{if } X_{i'} = X_i + v_{\mu} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

This represents the infinitesimal generator of the Markov process [10,12,14], such that rows and columns are ordered according to lowercase, i and j respectively. The entry of $a_{i,j}$ of the matrix gives the propensity for the chemical system to transition from one state to another state, given that $i \neq j$, are non-negative. The diagonal terms of the matrix are defined by a_{jj}, when $i = j$ and has a zero-column sum, so its probability is conserved. From Eq. (3) we can derive the $P^{(t_f)}$ probability vector at the final time, t_f, of interest given an initial density of $P^{(t_0)}$:

$$P^{(t_f)} = exp(t_f A).P^{(t_0)}, \tag{5}$$

where the matrix exponential function is defined by the convergent Taylor series as [15,16]

$$exp(t_f A) = I + \sum_{n=1}^{\infty} \frac{(t_f A)^n}{n!}. \quad (6)$$

However, algorithms, such as in [13,17–19] truncate Eq. (6) infinite summation to approximate Eq. (3) at the cost of a truncation error.

Initial Value Problem

If v_μ or v_M, for μ or $M = \{1,2, \ldots . M\}$ be the stoichiometric vectors for R_M reaction channels, then we will define the stoichiometric matrix for the system by V_μ or $V_M = [v_1; v_2; \ldots\ldots v_\mu]^T$. Let φ be the sample space and $X_0 \in \varphi$ be the initial state of the system if $\mathbf{X}_J$ denotes the only set of states in φ. To solve $P^{(t)}(X)$ in Eq. (2) for $X \in \varphi$, we define the $P^{(t)}$ vector to be $(P^{(t)}(X))_{X \in \varphi}$ or $(P^{(t)}(X))_{X \in \mathbf{X}_J}$ for a finite set of states, then $\frac{\partial P^{(t)}}{\partial t}$ be defined as a vector $(\frac{\partial P^{(t)}}{\partial t})_{X \in \varphi}$. Therefore, solving the CME is to find the solution of the initial value problem over a time period by the differential equation Eq. (3) when $t > 0$, whereas, $P^{(t_0)}$ is the initial distribution at $t = 0$. Here, the sample space φ can be infinite for large biochemical systems, so finding the solution of Eq. (3) for the given parameters with finite set of states $\mathbf{X}_J$ is cited as the CME problem of large biochemical systems because the size of A will be extremely large.

For example, consider an enzymatic reaction network [13] described by reactions $R_1: S + E \xrightarrow{k_1} C$, $R_2: C \xrightarrow{k_2} S + E$, $R_3: C \xrightarrow{k_3} P + E$. This network of reactions involves four species: namely, $S -$ substrate, $E -$ enzyme, $C -$ complex and $P -$ product molecules. The $X \equiv (x_1, x_2, x_3, x_4)^T \equiv (S, E, C, P)^T$ represents any state of the system, given $X_0 \equiv (S_0, E_0, C_0, P_0)$ as the initial state. The stoichiometric vectors are given by $v_1 = (-1, -1, 1, 0)$, $v_2 = (1, 1, -1, 0)$, $v_3 = (0, 1, -1, 1)$. Therefore, for (x_1, x_2, x_3, x_4) $x_{\tilde{N}=4}$, the propensity functions are:

$$R_1: \quad a_1([x_1], [x_2], [x_3], [x_4]) = k_1 \times x_1(t) \times x_2(t)$$

$$R_2: \quad a_1([x_1], [x_2], [x_3], [x_4]) = k_2 \times x_3(t)$$

$$R_3: \quad a_1([x_1], [x_2], [x_3], [x_4]) = k_3 \times x_3(t)$$

The set of states reachable from X_0 is finite in number. With the multiple explosions of the number of states in a large model, the size of A increases multiple folds.

As seen in Eq. (5), the difficulty of solving Eq. (2) is a problem when the dimension of the model grows with the number of species present in the system – especially for large biochemical models. The approximate estimate of $S^{\tilde{N}}$ shows how the size of the problem increases and this explosion in size is known as the *curse of dimensionality* [9,13]. The solution of CME given in Eq. (5) has two major parts: (a) the expansion of state-space, (b) the approximation of the series. For the expansion of state-space, Finite State Projection (FSP) [20] and Sliding Windows (SW) [17] are used to find the domain. Whereas, methods like Krylov subspace [13] and Runge Kutta [21] are popularly used for approximation (of the series) of the CME Eq. (5).

Although, CME has been employed and solved explicitly for relatively small biological systems [13,17–19,22,23], computationally complaisant but accurate solutions are still unknown for most significant systems and also for large systems having infinite number or a very large number of states. This deprivation of closed-form solution has driven the system biology research towards *Monte-carlo Algorithms* (MC) [24] to capture dynamics. One of such algorithm is *Stochastic Simulation Algorithm* (SSA) by Gillespie [9] used for processes in the CME. On other hand, the original FSP state-space expansion has been put into practice in [20], [25] has some drawbacks [20]. The FSP [20] and its variants [19,23,25,26] are based on *r-step reachability* [25], whereas, SW [17] is also a FSP based method but employ stochastic simulation algorithm (SSA) for finding the domain which proves better than FSP and suitable for stiff problems. Add-on weighting functions like $GORDE$ [27] and likelihoods [23] methods are used to improve the expansion. It is reported that $FSP\ GORDE$ [27] removes the states with small probabilities before the calculation of Eq. (5) which saves the computational time and performs faster than conventional FSP *r-step reachability.* However, removing the probabilities before the calculation of Eq. (5) increases the steps error and affects the accuracy of the final solution at t_f even when the state-space is small or large. If one is interested in solving stiff and/or large systems, it will greatly affect the solution.

It is noted that FSP variant called *Optimal Finite State Projection* $(OFSP)$ [19] based on *r-step reachability* performs better in terms of producing optimal order domain and perform faster than FSP as well as $FSP\ GORDE$. It is also infeasible to use SW for large CME problems either as creating hyper-rectangles is a very difficult task. At least four-times the number of SSA simulations required to minimize the error to half, because the convergence rates of routines in MC is very slow. The original SSA takes long time since one simulation may have number of different R_M. Recently, the efficiency of the SSA have greatly enchanced by the researchers through various schemes such as τ leaps (adaptive) [28,29]. Therefore, we will compare the $OFSP$ and SSA (τ leaps adaptive) with the ISP in terms of finding the domain, accuracy and computational efficiency. In such cases, the crucial part of solving the CME remains in finding the right projection size (domain) for large models which would then make the approximation efficient.

In this paper, we mainly focus on developing the expansion strategy, namely *Intelligent State Projection* (ISP) method to mitigate the problems of accuracy of the solution, performance of the method and projection size. The ISP has two variants namely – *Latitudinal Search* (LAS) and *Longitudinal-Latitudinal Search* $(LOLAS)$. It treats the Markov chain of a biochemical system as Markov chain tree structure and state as objects of class *node*. Based on the dimension of the system, search is performed in latitudinal way for different size of models using $ISP\ LAS$ method. Whereas, bidirectional search is applied using $ISP\ LOLAS$, which quickly expands the state-space up to a specified bound limit. To support the expansion strategy, we also develop *Bayesian Likelihood Node Projection* $(BLNP)$ function, based on Bayes' theorem [30,31]. It is adjoined with ISP variants to find the likelihood of the events at any interval at molecular population level. $BLNP$ provides confidence to the expansion strategy by assigning probability values to the occurrence of future reactions and prioritizing the direction of expansion. The ISP embedding $BLNP$ function inductively expands the multiple states with likelihood of occurrence of the fast and slow reactions. It also defines the complexity of the system by predicting the pattern of state-space updation, and depth of the end state from the initial state. In ISP, the advantage of the LAS is, when used for any size of

biological networks, the amount of memory usage is proportional to the entire width of expansion which is less compared to $ISP\ LOLAS$, and because of this, both the methods quickly becomes feasible and differentiated for various types of biological networks. However, computational time of both the variants depend on the nature of model and size of time step used. At any point, the amount of memory in use is directly proportional to the neighbouring states reachable through single R_M reaction. On the other hand, $ISP\ LOLAS$ uses much less memory even though when it retracts to initial node to track new reaction followed by revisiting the depth many times.

Results

In this section, we discuss the modelling and integration of the biochemical reaction systems for ISP methods as well as the assumptions underlying these methods. During ISP setting, we tested its ability to reproduce the model to measure dynamics of the key parameters in the models. We found that ISP method is a novel easy-to-use technique to model and expand the state-space of biochemical systems, which also shows several improvements in modelling and computational efficiency.

Computational experiment (Initializing, and solving the model) was conducted on carbon-neutral platform of Amazon® Web Service Elastic Computing (EC2) instance type large (m5a) running on HVM (hardware virtual environment) virtualization with variable ECUs, multicore environment 16vCPU @ 2.2GHz, AMD EPYC 7571 running Ubuntu 16.04.1 with relevant dependencies, 64GB memory with 8GB Elastic Block Storage (EBS) type General Purpose SSD (GP2) formatted with Elastic File System (EFS). The performance mode was set to General Purpose with input-output per second (IOPS = 100/3000) and throughput mode of type bursting is set (also see Supplementary Information (SI) 1).

Intelligent state projection

The main idea of the proposed algorithm is to expand the $\mathbf{X}_K$ iteratively such that $\mathbf{X}_K$ contains a minimum number of states carrying the maximum probability mass of the system. To create the sample space for ISP, Markov chain tree $\mathcal{H}$ [32] is used to visualize a biochemical system to exhibit transition matrix as directed trees [10,11] of its associated graph. Additionally, Markov chain tree $\mathcal{H}$ generates sample space of the system to represent Markov processes associated with the Markov chain and the transition matrices of biochemical reaction networks. In following section, we will visualize the Markov chain of the biochemical system as Markov chain graph (tree) for ISP compatibility.

(a) Markov chain as a Markov chain tree

We define the Markov chain tree, $\mathcal{H}$, [32] as infinite, locally finite, connected to a special type of graph with a prominent vertex called a parent node without loops or cycles. Let graph G_{mc} be a state-space of the finite state Markov chain with $\{P(X_i, X_{i'}) \mid X_i, X_{i'} \in G_{mc}\}$ transition probabilities meeting the condition $\sum_{X_{i'}} P(X_i, X_{i'}) = 1$, then the induced Markov chain tree is a combination of valued G_{mc} random variables with the distributions inductively defined from $P(X_i, X_{i'})$ with an initial state, $X_i \in G_{mc}$. That being the case, it is easy to expand this

class of Markov random field through a Markov chain tree structure for biochemical systems. Also, the Markov chain tree and Markov processes can be equated as explained in [33] for the stochastic analysis.

Since we are interested in aperiodic states in the expansion of state-space, we shall assume the reducibility or simplification of the G_{mc}; namely for each $X_i, X_{i'} \in G_{mc}$ through $\mathbb{K}$. Therefore, let us concentrate on the case where G_{mc} is considered as a locally finite connected graph, but the transition probabilities of each state are not equal due to the propensities and parameters of different reactions in the biochemical system. Consequently a Markov chain tree, $\mathbb{K}$, can be used to visualise a biochemical system process to exhibit a transition matrix as directed trees of its associated graph [10,11], and to generate a sample space for the system to represent the Markov processes and the transition matrices of biochemical reaction networks. Further in this section, we will discuss the details needed to represent Markov models on trees and working with graphs for state-space.

Let $\mathbf{X}_J$ be the finite set of cardinality $\{1,2 \ldots \ldots K\}$ of a Markov chain $\mathbb{K}_c$, and A be the transition probability matrix associated with $\mathbf{X}_J$. A state-space is, substantially, a class of a set of states containing the unique state of the system, and the arcs between the states represent the transitions from the initial state to the end state. This transition defined as transient and communicating class in graphs. When all the transitions are combined, every state-space takes the form of a graph and creates the state-space of the system, as shown in Figure 1 below.

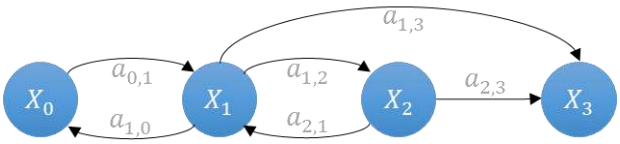

Figure 1. Markov chain graph showing forward and reversible reactions through four different states.

We can now associate chain $\mathbb{K}_c$ with the directed graph $G_{mc} = (\mathbf{X}_J, V_\mu)$, where $V_\mu = [v_1; v_2; \ldots\ldots v_\mu]$ and v_μ defines the transition from state X_i to $X_{i'}$ and is denoted as $v_\mu = \{(X_i, X_{i'});\ a_{i,j} > 0\}$. For every transition $(X_i, X_{i'}) \in \mathbf{X}_J$, then weight $\omega(X_i, X_{i'})$ is $a_{i,j}$.

Suppose G_{mc} has a cycle, which starts and terminates at some state, $X_i \in \mathbf{X}_J$. If there is a transition from X_i to $X_{i'}$, but we add a unique transition by creating a cycle from X_i back to itself and then consider the original transition from X_i to $X_{i'}$. This contradicts the uniqueness of the walk in tree [34]. When relating this to the CTMC of a biochemical system process, the change in molecular population is defined by a stoichiometric vector, so, in G_{mc}, there must be at least one intermediate state that will send the system back to the previous state to create the cycle. This process categorises the *forward* and *backward* reactions given the initial state, X_0, of the system. The transient class of the transition leads the system to a unique state that defines the *forward* reaction in the system. Whereas, the communicating class of a transition defines the reversible reaction in the system. In this section, we will define such systems as transient class systems and communicating class systems. Large biochemical systems are usually a combination of both classes.

A biochemical system is visualised as tree $\mathbb{T}$ [32] for the expansion of the state-space. A tree, $\mathbb{T}$, is a special form of graph in data structure constituting set of nodes and a collection of edges (or arcs) that each connects with an ordered pair of nodes. G_{mc} is considered as a directed tree, $\mathbb{T}$, and is rooted with $N_0 = (X_0, \mathfrak{d}_l)$ if it contains a unique walk to $N_i = (X_i, \mathfrak{d}_l + 1)$ and does not contain any cycles, while $X_i \in \mathbf{X}_J \backslash \{X_0\}$ has exactly one outgoing transition away from X_0 – in which case, it is called an arborescence, or making its transition towards $N_0 = (X_0, \mathfrak{d}_l)$ – in which case, it is called as anti-arborescence. An arborescence is a subset $\subseteq V_\mu$ that has one edge out of every node containing no cycles and having a maximum cardinality. For example, set $U = \{5,7,8,10\}$ contains 4 elements, then cardinality of $|U|$ is 4.

If X_i and $X_{i'}$ be the states other than the initial X_0 state. There is a transition from X_i to $X_{i'}$, so X_i has at least one transition. Now, suppose X_i has two walks, (X_i, X_{i+1}) and (X_i, X_{i+2}). Concatenating these walks to the walks $(X_{i+1}, X_{i'})$ and $(X_{i+2}, X_{i'})$, respectively, we have two distinct changes in state from X_i to $X_{i'}$ in G_{mc} but in $\mathbb{T}$, this concatenation is not considered, which makes them Directed Acyclic Graphs (DAG) as shown in Figure 2. Most of the biochemical models G_{mc} can be visualised as DAGs irrespective of the nature of reactions present in the model. Figure 2 shows the equivalent tree of G_{mc} shown in Figure 1. The trees are less complex as they have no cycles, no self-loops, and are still connected to depict the state-space.

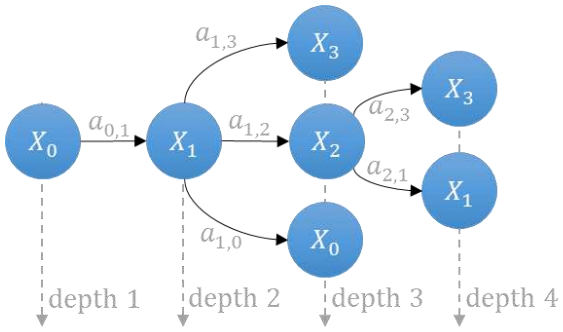

Figure 2. Equivalent tree of Markov chain graph, as shown in Figure 1. It is a special form of graph that has no cycle, no self-loops and depicts the state-space of the system in the form of a tree (DAG).

The weight of the tree containing all e edges is defined by $\omega(\mathbb{T}) = \prod_{e \in \mathbb{T}} \omega(e)$, where $\omega(e) = \omega(X_i, X_{i'}) = a_{i,j}$ is the weight of an edge starting from X_i and ending at $X_{i'}$ when $e \in \mathbb{T}$ [35]. For the systems having both *forward* and *backward* reactions, if $\mathbf{n}_J$ is the total number of nodes indexed by $\{1,2 \ldots K\}$ same as states, $\mathbf{n}_K$ be the set of nodes carrying $\mathbf{X}_K$, and $\mathbf{n}'_K$ be the set of nodes carrying $\mathbf{X}'_K$ given N_0 root node of the tree $\mathbb{T}$, then the walk from one node to another node is given by:

$$\{f(N_i, N_{i'}), f(N_{i'}, N_i) \mid N_0\} \in \mathbf{X}_J, \tag{7}$$

and $\mathbb{T}$ is formed by superimposing the forward transitions between the states X_i and $X_{i'}$, with the reverse orientation, $X_{i'}$ and X_i, for backward reactions but these are graphically denoted as an individual edge from $N_i = (X_i, \mathfrak{d}_l)$ to $N_{i'} = (X_{i'}, \mathfrak{d}_l + 1)$ to $N_i = (X_i, \mathfrak{d}_l + 2)$ in a tree. The N_i of $\mathfrak{d}_l + 2$ can be renamed to a new node, N_{i+1}, as it is at a different depth from N_i of $\mathfrak{d}_l$ but storing the same state X_i. In the expansion, repeated states are not

considered in the domain; therefore, any node carrying a similar state will be considered the same regardless of the level and indexing. Consideration of trees for the state-space expansion in ISP will not only help in reducing the complexity but also in improving the accuracy of the solution of Eq. (5) by identifying nodes carrying probable states. If the Markov chain graph starts in state $X_i \in \mathbf{X}_J$, then the mean number of transits to any state $X_{i'}$ converging to $\overline{a_{X_i,X_{i'}}}$ is given by the $(i,i')^{\text{th}}$ value of

$$\bar{\mathrm{A}} = \lim_{n\to\infty} \left(\frac{1}{n}\right) \sum_{k=0}^{n-1} A^k. \tag{8}$$

Let U be the set of all arborescences. Let $U_{X_i,X_{i'}}$ be the set of all arborescences which have a transition from X_i to $X_{i'}$ and $\left\| U_{X_i,X_{i'}} \right\|$ be the sum of the weights of the arborescences in $U_{X_i,X_{i'}}$ then according to Markov chain tree theorem [32],

$$\overline{a_{X_i,X_{i'}}} = \frac{\left\| U_{X_i,X_{i'}} \right\|}{\|U\|} \tag{9}$$

$\overline{a_{X_i,X_{i'}}}$ is probabilistic in nature. This nature is not only restricted to the systems having irreducible Markov chains in which graph G_{mc} is strongly connected while carrying probable state-spaces but also for the systems that can be simplified by converting to a Markov chain tree and then by reducing the tree by ignoring the states having low probabilities in space φ.

(b) Expansion criterion for state space

As mentioned before, the states are indexed by $\{1,2 \ldots \ldots . K\}$ in the domain denoted by set $\mathbf{X}_J$. To derive the time based on the state-space expansion conditions, the probability exponential form of the CME Eq. (5) is evaluated for approximation up to the desired final time t_f in steps. To focus on the probable states that contribute most to the probability mass in the domain, we first define the set of non-probable states (having the least probability mass) as $\mathbf{X}'_K$, which are to be bunked. Number of states will usually be infinite without selecting probable states for the domain. By doing this we can avoid unnecessary recalculation of probabilities and decrease the computational efforts by keeping the domain small. This bunking can also be applied to the initial distribution of the system at t_0. If submatrix A'_j contains the non-probable set $\mathbf{X}'_K$ of states, then probability of set will be,

$$P^{(t)}(\mathbf{X}'_K) = exp\left(t.A'_j\right).P^{(t)}(X_0). \tag{10}$$

The criterion for defining the non-probable states is determined by the τ_m tolerance value. A'_j will only be considered to have non-probable states if,

$$A'_j = \begin{cases} \text{nonprobable states,} & \text{if } P^{(t)}(\mathbf{X}'_K) < \tau_m \\ \text{else,} & \\ \text{probable states,} & \text{if } P^{(t)}(\mathbf{X}'_K) \geq \tau_m \end{cases} \tag{11}$$

Similarly, submatrix A_j has a probable set $\mathbf{X}_K$ of states if $P^{(t)}(\mathbf{X}_K) \geq \tau_m$ otherwise, the states from $\mathbf{X}_K$ are bunked to $\mathbf{X}'_K$ if $P^{(t)}(\mathbf{X}_K) < \tau_m$. For any iteration, if $P^{(t)}(\mathbf{X}'_K) \geq \tau_m$ then (from Eq. (11)) some states from $\mathbf{X}'_K$ return to $\mathbf{X}_K$ in the next iteration to increase the accuracy of

the approximate solution (*Æ*). The column sum of the approximate solution (*Æ*) of these states is defined as:

$$Æ = I^T exp\left(t_f A_j\right).P^{(t)}(X_0), \tag{12}$$

where, I = $(1, \ldots 1)^T$ is of an appropriate length. Declaring some states as non-probable and removing them before calculation of the probabilities as seen in [27] will decrease the accuracy of *Æ* with the cumulative step errors. This can be validated from the state probabilities that have been ignored in the domain:

$$Æ = 1 - P^{(t)}(\mathbf{X}'_K). \tag{13}$$

We define the step error in terms of the probabilities bunked. If $e_{rror} \propto P^{(t)}(\mathbf{X}'_K)$ then,

$$e_{rror} = 1 - I^T exp\left(t_f A_j\right).P^{(t)}(X_0) \tag{14}$$

$$e_{rror} = 1 - Æ \tag{15}$$

Every expansion step explores at least one new state and change $\{\mathbf{X}_K\}$ but not necessarily $\{\mathbf{X}'_K\}$ as long as:

$$P^{(t)}(\mathbf{X}_K) \geq \tau_m > P^{(t)}(\mathbf{X}'_K), \tag{16}$$

is satisfied. For ideal systems with a given initial probability of $P^{(t_0)}(X_0)$, the $\{\mathbf{X}'_K\}$ should be *null* and so $P^{(t_f)}(\mathbf{X}'_K) = 0$. For such systems $\{\mathbf{X}_K\}, \{\mathbf{X}'_K\}\ \epsilon \left\{\mathbf{X}_J\right\}$ for final projection and,

$$P^{(t_f)}\left(\mathbf{X}_J\right) = P^{(t_f)}(\mathbf{X}_K) + P^{(t_f)}(\mathbf{X}'_K), \tag{17}$$

$$P^{(t_f)}\left(\mathbf{X}_J\right) = P^{(t_f)}(\mathbf{X}_K) + 0. \tag{18}$$

$P^{(t_f)}\left(\mathbf{X}_J\right)$ in Eq. (18) is the solution of Eq. (3) after the state-space is expanded to $\mathbf{X}_K$. However, for large biochemical systems, Eq. (18) may not hold completely true due to the nature (*fast* ($R_{M(fs)}$) and *slow* ($R_{M(sr)}$)) of some reactions present in the system; therefore, the condition in Eq. (11) will pass the states from $\mathbf{X}'_K$ to $\mathbf{X}_K$ and then the states with lowest probabilities will be bunked when:

$$P^{(t)}(\mathbf{X}'_K) << P^{(t_f)}(\mathbf{X}_K), \tag{19}$$

which improves the solution. Removing without calculating the probabilities of some states is one of the lag, the current methods [17,19,20,23,25–27] are facing over the cost of achieving the truncated domain and saving computation time. To address this, we set a $P^{(t)}(\mathbf{X}'_K)$ leakage point based on:

$$P^{(t)}(\mathbf{X}_K) \geq \tau_m(leak) > P^{(t)}(\mathbf{X}'_K), \tag{20}$$

where, $\tau_m(leak)$ for systems will reform the Eq. (16) as:

$$P^{(t)}(\mathbf{X}_K) \geq \tau_m * 0.4 > P^{(t)}(\mathbf{X}'_K), \tag{21}$$

which would then zip the $\mathbf{X}'_K$ further by leaking the highest probabilities to $\mathbf{X}_K$ so the probability sum is no longer conserved. The motivation of setting this ration is to reconsider (up to 40% of $\mathbf{X}'_K$) the bunked states to improve the Æ solution and decrease the expansion step error. While modelling the biochemical system, if *slow* and *fast* reaction criterion is considered during expansion, then $\tau_m(leak)$ will be defined as,

$$= \begin{cases} \tau_m * \dfrac{(\text{no. of } R_{M(sr)})}{(\text{no. of } R_{M(fs)})}, & \text{if no. of } R_{M(sr)} < \text{no. of } R_{M(fs)} \\ \text{else,} & \\ \tau_m * \dfrac{(\text{no. of } R_{M(fs)})}{(\text{no. of } R_{M(sr)})}, & \text{if no. of } R_{M(sr)} > \text{no. of } R_{M(fs)} \\ \text{else,} & \\ \tau_m * 0.4, & \text{if no. of } R_{M(fs)} = \text{no. of } R_{M(sr)}. \end{cases} \tag{22}$$

We consider Eq. (21) criterion throughout the computational experiments in this study. The conditions in Eqs. (21) and (22) will then lead to an optimal set of states as,

$$\mathbf{X}_K \longleftarrow \mathbf{X}_K - \mathbf{X}'_K, \tag{23}$$

at t_d in the domain. When $\mathbf{X}_K$ is updated at every t_{step} before reaching t_f, it creates several intermediate domains which we define as $Bound$. At t_0, the domain only has the initial state of the system; therefore, we define the $Bound$ as:

$$Bound_{lower} = \{domain, \mathfrak{d}_{l=1}\}. \tag{24}$$

After a single t_{step} of expansion, if $\mathbf{X}_K$ is updated with a new state or set of states, it creates:

$$Bound_{upper} = \{domain, \mathfrak{d}_l\} \tag{25}$$

at t_d. Here, $\mathfrak{d}_l$ denotes the depth level of the latest state or set of states that has been added in the domain to form $Bound_{upper}$. Further, this $Bound_{upper}$ is re-labelled and considered as $Bound_{lower}$ for the next t_{step} of the expansion. If the expansion is to be limited in the number of $Bounds$, then every $count(\mathfrak{b}_{limit})$ leads to:

$$count(\mathfrak{b}_{limit}) = \mathfrak{b}_{limit}, \tag{26}$$

where, $\mathfrak{b}_{limit}$ is the bound limit. For example, if $\mathfrak{b}_{limit} = 2$, then $count(\mathfrak{b}_{limit})$ will be from $0 \overset{\text{to}}{\rightarrow} 1 \overset{\text{to}}{\rightarrow} 2$. If the $count(\mathfrak{b}_{limit})$ is increased up to $\mathfrak{b}_{limit}$ for $I_{tr}{}^{\text{th}}$ iterations, then $Bound_{upper}$ in the current iteration will be $Bound_{lower}$ for the next iteration. Every $Bound_{lower}$ state will be the strict subset of every consecutive $Bound_{upper}$ given as:

$$Bound_{lower}(Z) \subset Bound_{upper}(Z). \tag{27}$$

and the upper bound as:

$$Bound_{upper}(Z) = \{domain\ at\ Z^{th}\ iteration, \mathfrak{d}_l\}, \tag{28}$$

where Z is the number of $Bounds$ (or intermediate domains). The *2D pyramid* domain in Figure 3 showcases the increase in the population of states in the domain with the increase in iterations (I_{tr}). The apex of the pyramid represents the initial state X_0 of the system at $Bound_{lower}(1)$ at t_0, whereas the base represents the deepest level where the system ends with the final domain carrying set $\mathbf{X}_K$ with maximum probability mass.

For large biochemical systems, the number of $Bounds$ created are based on I_{tr} and have million/billions of states. The expansion can be terminated by defining time t_f at which the

solution is required. In order to have auto break-off point in the expansion, it is important to define the criteria that limits I_{tr} when no more new states can be searched. Therefore, in the next section (c), we define this criterion which also fits biochemical systems having *fast* and *slow* reactions.

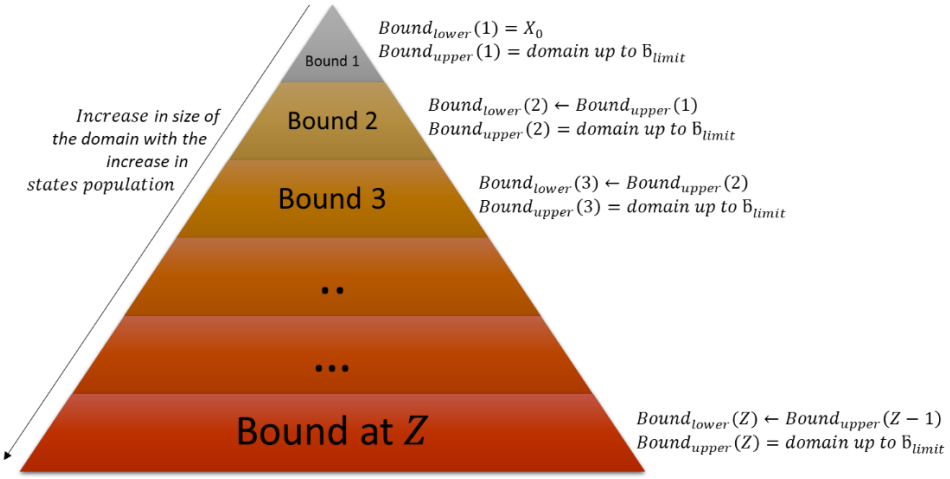

Figure 3. General framework of *2D pyramid* domain showing the increase in the size of the domain with the increase in state with an increase in the bounds. $\boldsymbol{Bound_{lower}(1)}$ represents the initial condition, whereas $\boldsymbol{Bound_{upper}(Z)}$ represents the final domain carrying explored set of states of the system.

(c) Cease of criterion after updating

In every expansion step, the domain is validated by Eq. (21) and new states are added in $\mathbf{X}_K$ as long as:

$$1 - I^T exp(t_f A_j).P^{(t)}(X_0) \geq \tau_m, \tag{29}$$

is satisfied for probable states and stops if it is not. This led to a point at t_f where $e_{rror} < \tau_m$, but the expansion can be extended further to meet the accuracy by re-considering the criteria as:

$$1 - I^T exp(t_f A_j).P^{(t)}(X_0) \geq \tau_m * \frac{(\text{no. of } R_{M(sr)})}{(\text{no. of } R_{M(fs)})}, \tag{30}$$

$$1 - I^T exp(t_f A_j).P^{(t)}(X_0) \geq \tau_m * \frac{(\text{no. of } R_{M(fs)})}{(\text{no. of } R_{M(sr)})}, \tag{31}$$

$$1 - I^T exp(t_f A_j).P^{(t)}(X_0) \geq \tau_m(leak), \tag{32}$$

before steps to t_f. However, the size of $\mathbf{X}_K$ obtained through Eqs. (30), (31) and (32) at t_f will be greater compared to the size of $\mathbf{X}_K$ obtained by Eq. (29) at t_f, as the latter will have fewer number of states. In Eqs. (30), (31) and (32), with the increase in size of A_j, the value of the left-hand side also increases resulting in an improvement in Æ. When considering any Markov

process of a biochemical system of any size in which the probability density expands according to Eq. (3) then Eqs. (30), (31) and (32) will approximate the solution within $\tau_m * \frac{(\text{no. of } R_{M(sr)})}{(\text{no. of } R_{M(fs)})}$, $\tau_m * \frac{(\text{no. of } R_{M(fs)})}{(\text{no. of } R_{M(sr)})}$ and $\tau_m(leak)$, respectively, of the true solution of the CME, which is Eq. (3).

Computational Experimental results

The ISP method is initialized and parameterized using given initial conditions of the models. Due to large number of math operations and equations, simultaneous parameter prediction with limited amount of experimental values at any instance is often complicated for dynamic systems. Therefore, the consistency with the available experimental data was ensured at each step of the ISP as this method has led to successful development of several functions that integrates large number of processes supporting extensive expansion of the state-space.

To demonstrate the $ISP\ LAS$ algorithm, we first consider the catalytic reaction system [36] defined by the reactions

$$S \overset{k_1}{\rightarrow} B \overset{k_2}{\rightarrow} C, \quad B + P \overset{k_3}{\rightarrow} B + E \tag{33}$$

depicted as a network in Figure 4 as:

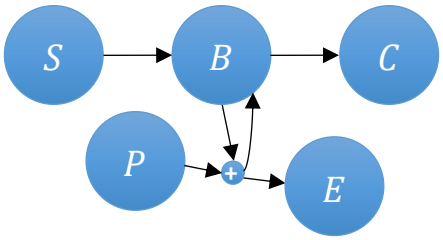

Figure 4. Catalytic reaction network having five $\tilde{N} = 5$ species S, B, C, P, and E in a network defining reactions, as given in Eq. (33).

In this biochemical system (dimension = 5), reactant P will transform into product E via complex B when reactant S acts as a catalyst for the reaction and produces C. We rewrite this catalytic reaction system as a network of three reactions:

$$\boldsymbol{R_1}: \quad S \overset{k_1}{\rightarrow} B, \tag{34}$$

$$\boldsymbol{R_2}: \quad B \overset{k_2}{\rightarrow} C, \tag{35}$$

$$\boldsymbol{R_3}: \quad B + P \overset{k_3}{\rightarrow} B + E\ ; \tag{36}$$

with the initial copy counts $S_0 = 50$, $P_0 = 80$, $B_0 = C_0 = E_0$ and the reaction rate parameters $k_1 = 1$, $k_2 = 1000$, $k_3 = 100$. These species counts are used as a state-space to define the model and these copy counts are tracked as $([S], [B], [C], [P], [E]) \in \tilde{N} := (x_0, x_1, x_2, x_3, x_4)$.

In reaction R_1, the copy count of S is reduced by 1, which increases the copy count of B by 1. In reaction R_2 the copy count of B is reduced by 1, which increases the copy count of C by 1.

Whereas, reaction R_3 decreases the count of B and P by 1 and increases the counts of B and E by 1. As in R_3, B act as a catalyst to convert P to E and B is retained in the same reaction. We can now define the transitions associated with R_1, R_2, R_3 in stoichiometric vector V_M matrix as:

$$V_M = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}. \tag{37}$$

For LAS method compatibility, the associated Markov chain of this model is converted into a Markov chain tree, and the states in terms of the nodes with additional information such as the number of R_M reactions required to reach the state. In the growing Markov chain tree, the transition between the nodes:

$$N_i \xrightarrow{v_\mu(X_0(t),X_1(t),\ldots,X_K(t))} N_{i+1}\,, \tag{38}$$

is defined in the typical form of the dictionary $Dict$. We express the propensity functions of the three reactions in terms of the states $([S],[B],[C],[P],[E]) \in \tilde{\mathrm{N}}$. Node $N_1 = (X_0, \mathfrak{d}_1)$ carries the initial state X_0 of the system at an initial depth of level 1. Further $\mathbf{n}_J = (\mathbf{X}_K, \mathfrak{d}_{1,2,\ldots})$ is expanded and the states updated by following the order of LAS. The corresponding propensities $\Delta a_{i,j}$ are updated in $A_{i,j}$ matrix in every iteration based on the LAS updating trend (for example see SI 2). Initially, the system started with $S_0 = 50$, $P_0 = 80$ and gradually all of the reactants are transformed to products, E and B, resulting in the system ending in $\mathbf{n}_J = (X_{1,2,\ldots\ldots 14666}, \mathfrak{d}_l)$.

Figure 5 shows the response of the LAS method when solved with $\tau_m = 1e-6$ for $t_f = 0.5$ *sec*. Due to the nature of the model reaction rates, small steps $t_{step} = 0.01$ *sec* are taken to capture the moments based on non-negative non-zero states for the domain. LAS successfully creates the domain of an optimum order with 14666 states at t_f by introducing the new states to the domain with time, as shown in Fig (a) in Figure 5. This pattern also depicts that the frequency (number of states at any time t) of expansion increases in depth when number of active reactions increase in the system. With the addition of probable states, the domain contains enough probability mass to approximate the solution up to t_f. The states are updated in sets as seen in Fig (b) of Figure 5, for the catalytic system after every iteration.

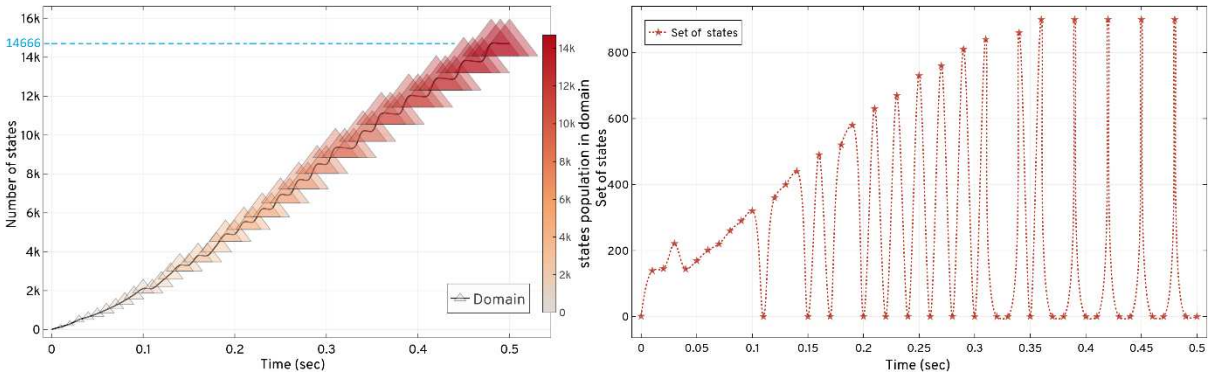

Fig (a) Fig (b)

Figure 5. Expansion and updation of the states and set of states explored for the catalytic reaction system based on the LAS method. Fig (a) depicts that state-space expansion

increases the number of additions of new states in the domain. The size and colour of ▲ shows the increase in the size of the domain with the states population. In Fig (b), LAS unfolds the state-space pattern to update the states in the domain and expands 14666 probable states in 0.5 *sec*.

The state-space pattern in Fig (b) of Figure 5 can be used as a *blueprint* of the catalytic systems' state-space to compare with other model's *blueprints* for their characteristics and occurrence of the reactions. Such a pattern is considered to predict the behavior of large network state-space expansions when the set of occurrences of the initial reactions are similar in different systems. The solution of Eq. (5) up to t_f, for the domain created by LAS is shown in Table 1 and the system's conditional probabilities based on its species are, as shown in Figure 6.

Table 1. LAS expansion response and solution at t_f for the catalytic system.

$t_f = 0.5,$ $t_{step} = 0.01$	Run-time (sec)	Domain	Expansion time (sec)	Error at t_f
$ISP\ LAS$	4677	14666	0.5	$1.865\text{e}-05$

In three test runs, the run time of $ISP\ LAS$ for the catalytic system was 4677 *secs* when solving the Eq. (5) with 14666 states. The probability of the species in Figure 6 shows the nature of the reactions affecting the counts of each species in the system. The involvement of species B in all the reactions results in its highest probability at t_f. Species B also acts as a catalyst for R_3, converting species P to E; therefore, both have equal probabilities at the time of solution.

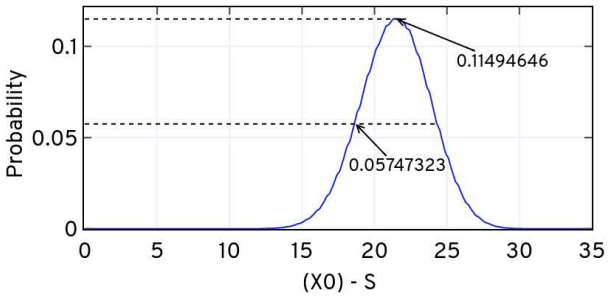

Fig (A). Probability of S over t_f

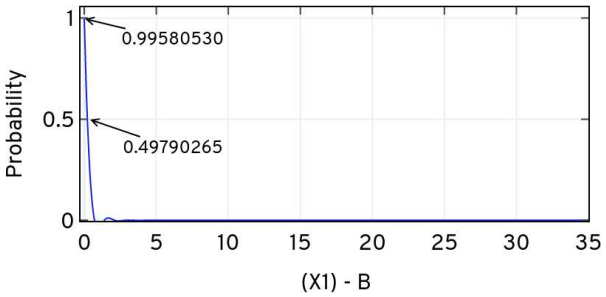

Fig (B). Probability of B over t_f

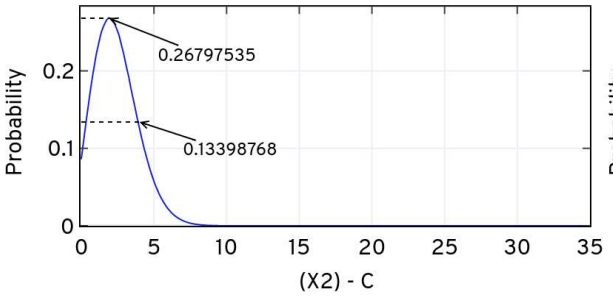

Fig (C). Probability of C over t_f

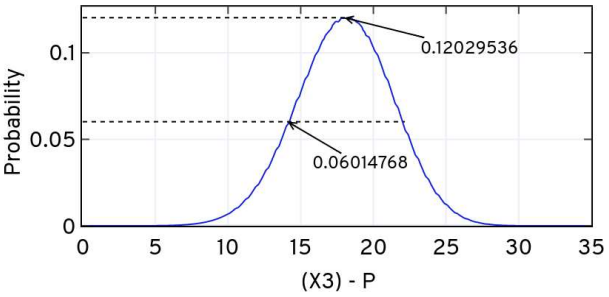

Fig (D). Probability of P over t_f

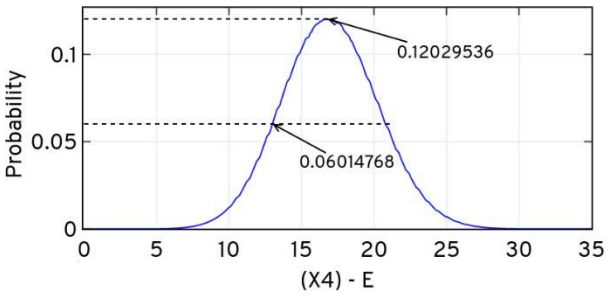

Fig (E). Probability of E over t_f

Figure 6. Conditional probability of the catalytic system evaluated at $t_f = 0.5$ *sec*, $t_{step} = 0.01$ using LAS. Fig (A) is the probability of the species S over t_f, Fig (B) is the probability of the species B over t_f, Fig (C) is the probability of the species C over t_f, Fig (D) is the probability of the species P over t_f, Fig (E) is the probability of the species E over t_f.

Figure 7 shows the total probability bunked at t' while progressing with the expansion. Bunking produces an error (w.r.t approximation) with time when the number of states increased with the expansion and provided that LAS produces minimal error of order 10^{-5}, as given in Table 1.

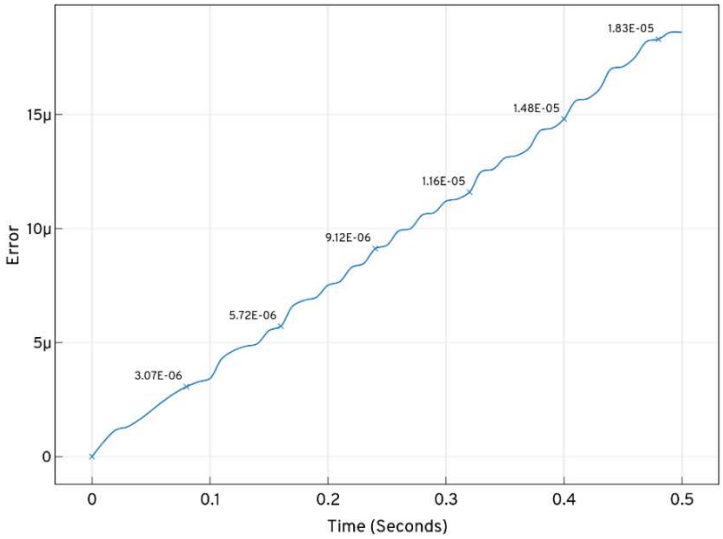

Figure 7. Total probability of states bunked at t' from the domain of catalytic system produced by $ISP\ LAS$ iteration while expansion and solving the CME.

Further to demonstrate the $ISP\ LOLAS$ algorithm, we consider the coupled enzymatic reactions defined by the reactions

$$S + E_1 \overset{k_1}{\rightarrow} C_1 \overset{k_2}{\rightarrow} S + E_1, \qquad C_1 \overset{k_3}{\rightarrow} P + E_1 \tag{39}$$

$$P + E_2 \overset{k_4}{\rightarrow} C_2 \overset{k_5}{\rightarrow} P + E_2, \qquad C_2 \overset{k_6}{\rightarrow} S + E_2 \tag{40}$$

depicted as a network in Figure 8 as:

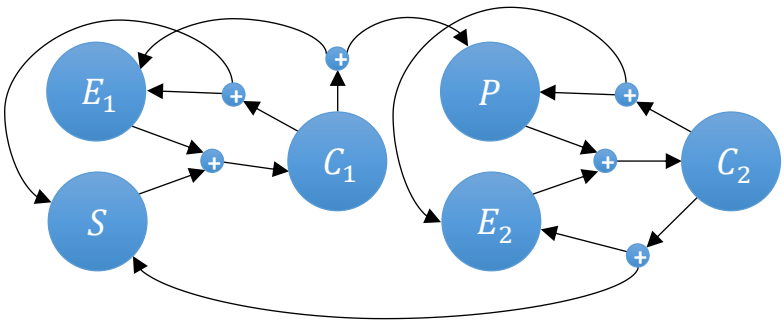

Figure 8. Coupled enzymatic reactions network. Showing six $\tilde{\mathrm{N}} = \mathbf{6}$ species, $\boldsymbol{S}$, $\boldsymbol{E_1}$, $\boldsymbol{C_1}$, $\boldsymbol{P}$, $\boldsymbol{E_2}$, $\boldsymbol{C_2}$, in a network defining reactions, as given in Eqs. (39) and (40).

This biochemical system (dimension = 6) describes two sets of enzymatic reactions transforming species S into species P and transforming species P back into S. We rewrite these enzymatic reactions system as a network of six reactions:

$$\begin{aligned} &\boldsymbol{R_1}\colon\ S + E_1 \overset{k_1}{\rightarrow} C_1,\quad \boldsymbol{R_2}\colon\ C_1 \overset{k_2}{\rightarrow} S + E_1 \\ &\boldsymbol{R_3}\colon\ C_1 \overset{k_3}{\rightarrow} P + E_1,\quad \boldsymbol{R_4}\colon\ P + E_2 \overset{k_4}{\rightarrow} C_2 \\ &\boldsymbol{R_5}\colon\ C_2 \overset{k_5}{\rightarrow} P + E_2,\quad \boldsymbol{R_6}\colon\ C_2 \overset{k_6}{\rightarrow} S + E_2 \end{aligned} \tag{41}$$

with initial copy counts $S = 50$, $E_1 = 20$, $E_2 = 10$, $C_1 = C_2 = P = 0$ and reaction rate parameters of $k_1 = k_4 = 4$, $k_2 = k_5 = 5$, $k_3 = k_6 = 1$. These species counts are used as a state-space to define the model and these copy counts are tracked as:

$$([S], [E_1], [C_1], [P], [E_2], [C_2]) \in \tilde{\mathrm{N}} := (x_0, x_1, x_2, x_3, x_4, x_5).$$

As in the previous example, we can now define the transitions associated with $R_1, R_2, R_3, R_4, R_5, R_6$ in stoichiometric vector V_M matrix as:

$$V_M = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}. \tag{42}$$

For the $LOLAS$ method, the associated Markov chain of this model is converted to a Markov chain tree, and the states in terms of nodes with additional information, such as number of R_M reactions required to reach the state. In growing Markov chain tree, the transition between the nodes:

$$N_i \xrightarrow{v_\mu(X_0(t), X_1(t), \ldots, X_K(t))} N_{i+1}\,, \tag{43}$$

is defined in the typical form of the dictionary $Dict$. We express the propensity functions of the six reactions in terms of the states $([S], [E_1], [C_1], [P], [E_2], [C_2]) \in \tilde{\mathrm{N}}$.

Node $N_1 = (X_0, \mathfrak{d}_1)$ carries the initial state X_0 of the system at initial depth level 1. Then $\mathbf{n}_J = \left(\mathbf{X}_K, \mathfrak{d}_{1,2,\ldots\ldots}\right)$ is further expanded and the states updated by following the order of $LOLAS$. The corresponding propensities $\Delta a_{i,j}$ are updated in the $A_{i,j}$ matrix in every iteration based on the given $LOLAS$ updation trend (for example see SI 2). Initially, the system started with

$S = 50$, $E_1 = 20$, $E_2 = 10$ and gradually all reactant species are transformed to products resulting in the system ending in $\mathbf{n}_J = \left(X_{1,2,\ldots\ldots 8296}, \beth_l\right)$.

Figure 10 shows the response of the $LOLAS$ method when solved with $\tau_m = 1e-6$ for $t_f = 2.0\ sec$. Due to the nature of the model reaction rates, small steps $t_{step} = 0.01\ sec$ are taken to capture the moments based on non-negative non-zero states for the domain. $LOLAS$ successfully creates the domain of an optimum order with 8296 states at t_f by introducing the new states to the domain with time, as shown in Fig (a) of Figure 9. In Fig (b) of Figure 9, pattern depicts that the frequency (number of states at any time t) of expansion increases in depth when the number of active reactions increases in the system. With the addition of probable states, the domain contains enough probability mass to approximate the solution up to t_f.

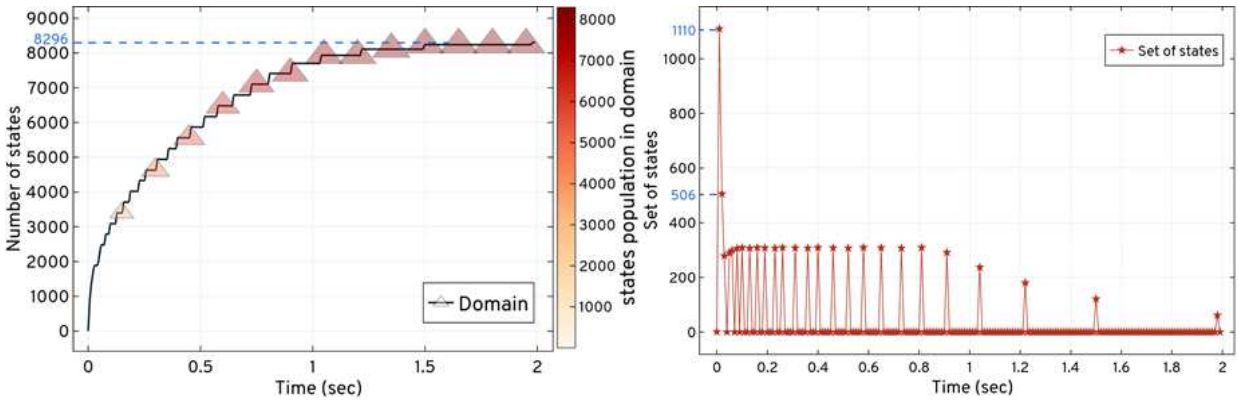

Fig (a) Fig (b)

Figure 9. Expansion and updating of the states and set of states explored for the dual enzymatic reaction network based on the *ISP LOLAS* method. Fig (a) depicts state-space expansion increases the number of additions of new states in the domain. The size and colour of ▲ shows the increase in size of the domain with the states' population. In Fig (b), ***ISP LOLAS*** unfolds the state-space pattern to update states in the domain and expands 8296 probable states in 2.0 *sec*. ★ shows the time point where new set of states is explored and updated in the domain.

The state-space pattern in Fig (b) of Figure 9 can be used as a blueprint of the dual enzymatic reaction network state-space to compare with other model blueprint for their characteristics and the occurrence of reactions. Such a pattern is considered to predict the behavior of a large network state-space expansion when the set of occurrences of the initial reactions are similar in different systems. The solution of Eq. (5) up to t_f for the domain created by $LOLAS$ is shown in Table 2 and the system's conditional probabilities based on species is shown in Figure 10.

Table 2. *LOLAS* expansion response and solution at t_f for the dual enzymatic reaction network.

$t_f = 2.0$, $t_{step} = 0.01$	Run-time (sec)	Domain	Expansion time (sec)	Error at t_f
ISP LOLAS	1614.22	8296	2.0	5.953e − 05

In three test runs, the run time of $ISP\ LOLAS$ for the dual enzymatic reaction network was ≈1614 *secs* when solving the Eq. (5) with 14666 states. The probability of the species in Figure 12 shows the nature of the reactions affecting the counts of each species in the system. At t_f, the probabilities of E_2 and C_2 remain high compared to E_1 and C_1 at different molecular counts, which results in a low probability of P compared to S. We know that this network transforms species S into species P and then transforms the species P back into S. Therefore, based on the current probabilities of the species at t_f, the future probability of P will increase and, for S, it will remain same or decrease. With this change, the probabilities of E_2 and C_2 decreased compared to E_1 and C_1.

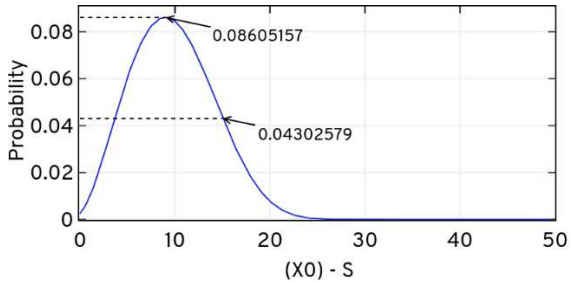

Fig (A). Probability of S over t_f

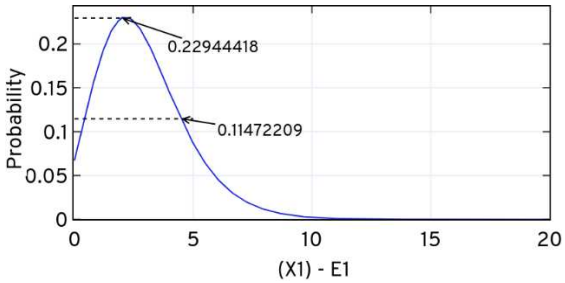

Fig (B). Probability of E_1 over t_f

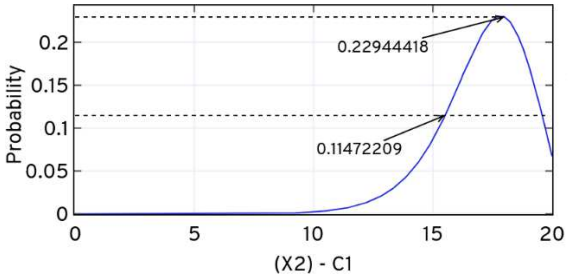

Fig (C). Probability of C_1 over t_f

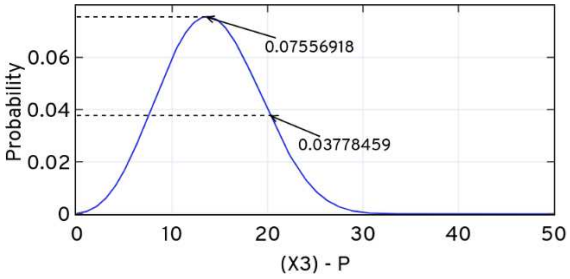

Fig (D). Probability of P over t_f

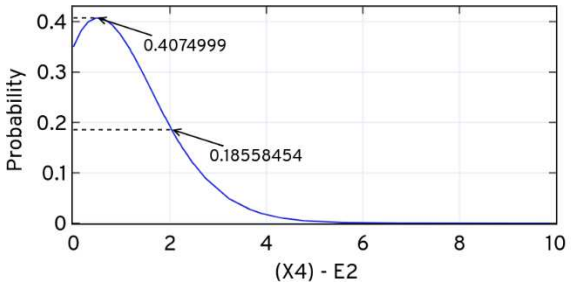

Fig (E). Probability of E_2 over t_f

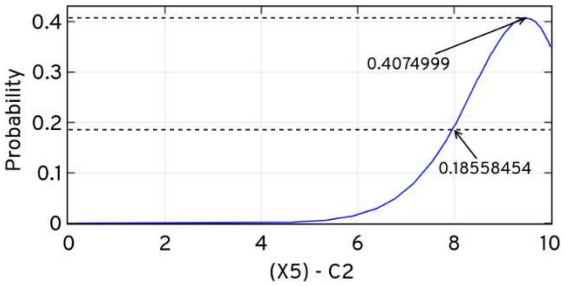

Fig (F). Probability of C_2 over t_f

Figure 10. Conditional probability of the dual enzymatic reactions system evaluated at $t_f = 2.0$ *sec*, $t_{step} = 0.01$ using $LOLAS$. Fig (A) is the probability of the species S over t_f, Fig (B) is the probability of the species B over t_f, Fig (C) is the probability of the species C over t_f, Fig (D) is the probability of the species P over t_f, Fig (E) is the probability of the species E over t_f.

Figure 11 shows the total probability bunked at t' while progressing with the expansion. The bunking produces an error (w.r.t approximation) with time when the number of states increases with the expansion and provided that, $LOLAS$ produces a minimal error of order, 10^{-5}, as given in Table 2.

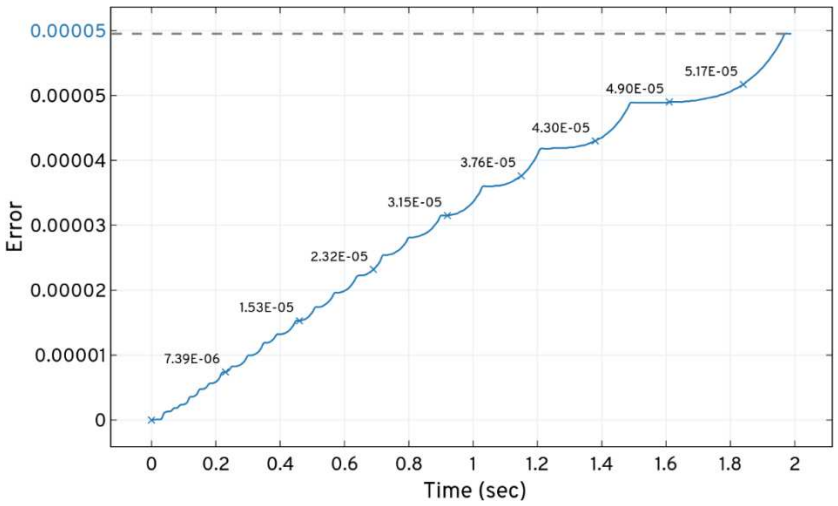

Figure 11. Total probability of states bunked at t' from the domain produced by dual enzymatic reactions system in $ISP\ LOLAS$ iteration while expansion and solving the CME.

Further, we extend the application of our ISP method to very large model which is our main aim. We simulate large model of the G1/S network [37] under DNA-damage condition to study the number of states at different time points and predict the conditional probabilities of the protein species based on events leading to the formation of different complexes in the system.

The G1/S model (dimension = 28) with a DNA-damage signal transduction pathway is considered to be very stiff in nature, so molecular counts of certain proteins increase very rapidly and some do so slowly, which makes it tough to solve, even for a short time. The model is solved for $t_f = 1.5\ sec$ with $\mathfrak{b}_{limit} = 1,\ \tau_m = 1e-6,\ t_{step} = 0.1$. The systematic exploration of nodes carrying probable states are undertaken in a similar way, as discussed in Table SI 4 of SI 3 and depicted (see Figure SI 7 of SI 3) in six stages (denoted as $\hat{\mathrm{S}}$), representing R_M reactions with propensity, a_μ, with the arcs as transitions.

The nodes are expanded up to t_f for the identifying the reaction channels responsible for variations in the proteins. It is observed from the transitioning factor of the *2nd-tier* that every node has an average of at least ≈97 possible child nodes carrying states and; further, $Dict$ is expanded for *n*-tiers of child nodes to add more states to the domain. Further, $\mathbf{n}_J = \left(\mathbf{X}_K, \mathfrak{a}_{l=1,2,\ldots\ldots}\right)$ is expanded and updated as per the trend of $ISP\ LOLAS$ (see Table SI 5 of SI 3).

The response from the $ISP\ LOLAS$ method between a number of states in the domain and time, t, is shown in Figure 12. The initial response suggests that only a few reactions were active until $t = 0.4\ sec$ and after that more reactions triggered that explosively take the exploration above 0.5 million states in 0.5 *sec*. For such a large model this combination of explosion states was expected because proteins undergo several excursions due to the number of reactions in fractions of time, t. The second explosion of states occurs after 1.0 *sec* when almost all the reactions (involving the species, given in SI 4.1) become active in the network. The size and colour of the *2D pyramid* in Figure 14 shows the increase in size of the domain with the state explosions. The number of set of states that create the bounds at t are shown in Fig (b) of Figure 12. With the exploration of the set of 517584 states, the $Bound(3)_{upper} = \{X_{0,1,2\ldots\ldots 604677}\}$ is formed at 0.5 *sec* carrying 604677 states. Some states were bunked at 0.5 *sec* resulting in approximation errors that reach $2.42\mathrm{e}-06$ at 0.6 *sec*. At

t_f, the $LOLAS$ ends up with a domain defined by $Bound(4)_{upper} = \{X_{0,1,2\ldots\ldots3409899}\}$ carrying 3409899 states with $3.52e-06$ approximation errors.

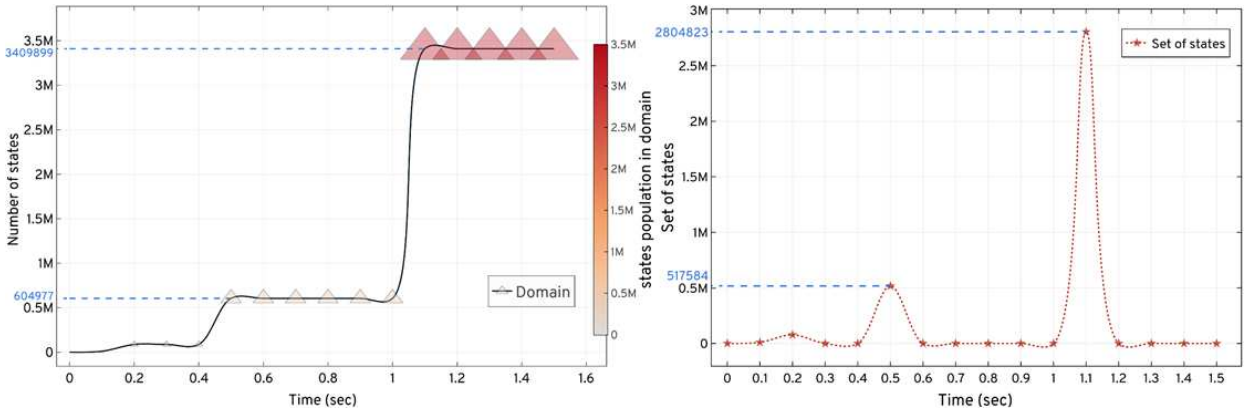

Fig (a) Fig (b)

Figure 12. Showing the expansion and updating of the states and set of states explored for the G1/S model based on the $ISP\ LOLAS$ method. Fig (a) depicts the state-space expansion increases the number of additions of new states in the domain. $ISP\ LOLAS$ quickly expands the state-space up to ≈3.5 million states in 1.5 *sec*. In Fig (b), $ISP\ LOLAS$ unfolds the state-space pattern to update states in the domain and expands 3409899 states up to t_f. ★ shows the time point where new set of states is explored and updated in the domain.

The corresponding propensities, $\Delta a_{i,j}$, are updated in the $A_{i,j}$ matrix in every iteration, based on the $ISP\ LOLAS$ update trend. The system started with the initial state of the protein species and gradually, when protein level changes in the system, it exploits the copy counts that shift the system to a new state. The change in protein levels causes the system to shift to new states, which manifest the Markov process of the system. The $ISP\ LOLAS$ captures this process and defines several bounds of the domain at different time intervals, as defined by pyramid in Figure 3. To investigate the expansion of states closely, the order of bounds at different time intervals, and the number of states present in the bounds, are given in Table 3. The size of bound created in each duration reveals that, for every step, the growth of the domain is eight-to-ten times the previous size of the domain.

Table 3. Lower and upper Bounds of the domain for G1/S model given by $ISP\ LOLAS$ trend based on bound limit $б_{limit}$.

Z	$Bound(Z)_{lower}$	$Bound(Z)_{upper}$	States	Duration
1	$Bound(1)_{lower} = \{X_0\}$ formed at $t = 0.0$ *sec* $Approximation = 1$	$Bound(1)_{upper} = \{X_{0,1,2\ldots\ldots9808}\}$ formed at $t = 0.1$ *sec* $Approximation = 0.999999867$	9808	0.0 – 0.1 *sec*
	$б_{limit} = 1, count(б_{limit}) = 0,1,$			
2	$Bound(2)_{lower} = Bound(1)_{upper}$ formed at $t = 0.1$ *sec* $Approximation = 0.999999847$	$Bound(2)_{upper} = \{X_{0,1,2\ldots\ldots87393}\}$ formed at $t = 0.2$ *sec* $Approximation = 0.999999173$	87393	0.1 – 0.2 *sec*
	$б_{limit} = 1, count(б_{limit}) = 0,1$			
3	$Bound(3)_{lower} = Bound(2)_{upper}$ formed at $t = 0.4$ *sec* $Approximation = 0.999999157$	$Bound(3)_{upper} = \{X_{0,1,2\ldots\ldots604677}\}$ formed at $t = 0.5$ *sec* $Approximation = 0.99999701$	604677	0.4 – 0.5 *sec*
	$б_{limit} = 1, count(б_{limit}) = 0,1$			
4	$Bound(4)_{lower} = Bound(3)_{upper}$ formed at $t = 1.1$ *sec*	$Bound(4)_{upper} = \{X_{0,1,2\ldots\ldots3409899}\}$ formed at $t = 1.5$ *sec*	3409899	1.1 – 1.5 *sec*

	$Approximation\ = 0.99999699$	$Approximation\ = 0.99999648$		
	$Б_{limit} = 1, count(Б_{limit}) = 0{,}1$			

The set of nodes $N_1, N_2, \ldots . N_{3409900}$ carries unique states representing the set of $state(n_{3409900}) = (X_{0,1,2,\ldots 3409899})$ that forms the state-space of the model. It is important to note that some proteins are synthesised and promoted by the network itself, as evidenced by some reactions of the pathway, which increase the frequency of the repeated states. However, $ISP\ LOLAS$ validation does not consider them for the domain. The solution of Eq. (5) up to t_f for the domain created by $ISP\ LOLAS$ is shown in Table 4.

Table 4. $ISP\ LOLAS$ expansion response and solution at t_f for the G1/S model.

$t_f = 1.5\ sec$, $t_{step} = 0.1$	Run-time (sec)	Domain	Expansion time (sec)	Error at t_f
$ISP\ LOLAS$	1372	3409899	1.5	$3.52e-06$

Over three test runs, the run time of $ISP\ LOLAS$ for the G1/S model was 1372 *secs* in solving the Eq. (5) with the optimal domain having 3409899 states. The response of $ISP\ LOLAS$ given in Figure 13, shows the system's probabilities bunked at t' while expansion (w.r.t approximation) when the number of states increases with the expansion and provided that, $ISP\ LOLAS$ produces minimal errors of the order of 10^{-6}, as given in Table 4 and Fig (a) of Figure 17. We have set the checkpoint to examine the initial state probability over time. The response in Fig (a) of Figure 13 indicate that the probability of the system remaining in the initial (normal) state decreases with time in the presence of DNA damage, which triggers the change in protein levels.

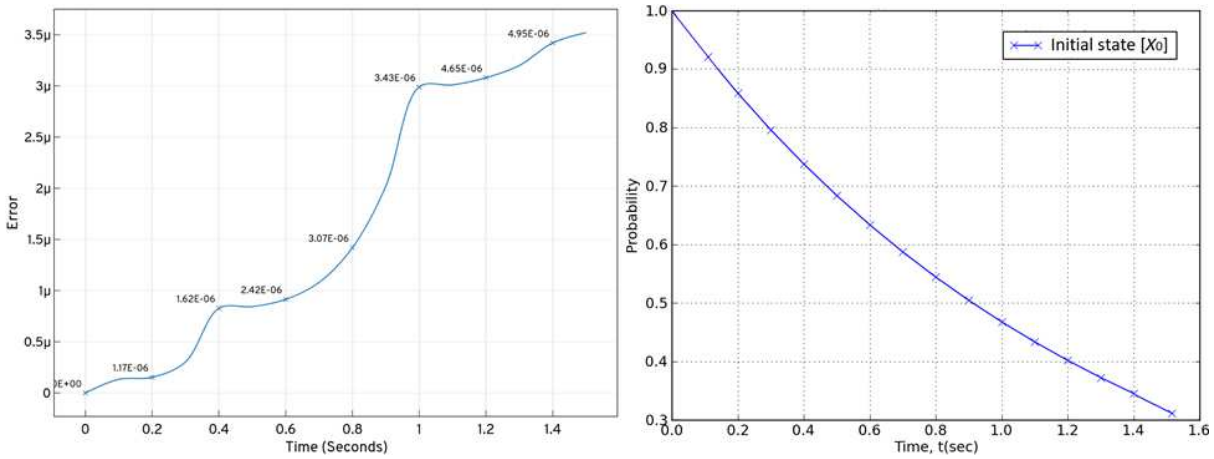

Fig (a) Fig (b)

Figure 13. Response of the $ISP\ LOLAS$ for total probability bunked at t' from the domain and checkpoint for examining the initial state probability over time. Fig (a) shows, how the accuracy of the result is maintained by the ISP for keeping low errors. Fig (b) shows the decline in probability of the system to remain in the initial state in presence of DNA damage.

The conditional probabilities of the species' systems are given in Figure 14. In the case of the DNA-damage situation, large numbers of the most notable parameters increase compared with the normal condition (in cell cycle progression). The increase is predominantly related to x_{14} (p21) having a high initial probability, see Fig(14) in Figure 14, and the feedback (negative)

of x_{24} (p53) increases its probability, see Fig(24) in Figure 14, such as association rate of x_{16} (p21/CycE/CDK2 − P), rate of synthesis of x_{14} (p21) by x_{24} (p53), rate of degradation of x_{14} (p21), rate of synthesis of x_{24} (p53) by DNA-damage signal. The conditional probabilities of the two key proteins, x_{10} (p27) and x_1 (CycE), are affected by the change in the response of cells to the level of the DNA-damage signal, see Fig (10) and Fig (3) in Figure 14. The parameters related to x_{10} (p27) as well as x_1 (CycE) greatly affect the probability of x_{21} (E2f) with time, see Fig (21) in Figure 14. The impact of x_1 (CycE) involves additional parameters related to CycA because the release of supplementary x_{21} (E2f) depends on x_{20} (Rb − PP/E2f) hyperphosphorylation by the activation x_7 (CycE/CDK2 − P), which affects the probability of x_{21} (E2f).

When the release of x_{21} (E2f) is affected, the probability of x_1 (CycE) increases, see Fig (3) in Figure 14, that leads to progression to the S-phase followed by the temporary suspension of cell cycle progression. The increase in probability of x_{24} (p53) shows support to cells to repair the DNA damage. The parameters and its probabilities relating to x_{14} (p21) and x_{24} (p53) become important in the case of DNA damage. When combined, the conditional probability of these parameters indicates the involvement of the DNA-damage signal in the transition of G1/S.

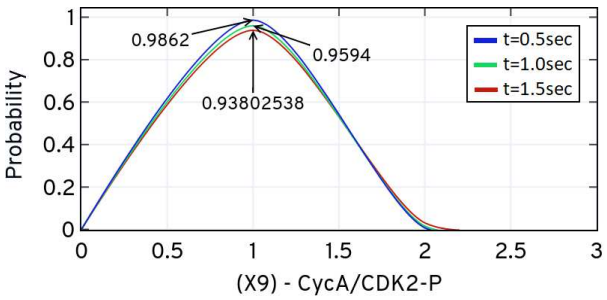

Fig (1). Probability of CycA/CDK − P over t_f

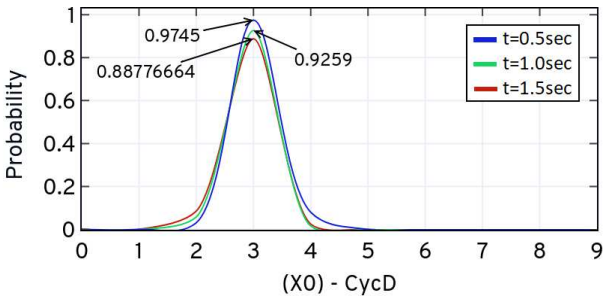

Fig (2). Probability of CycD over t_f

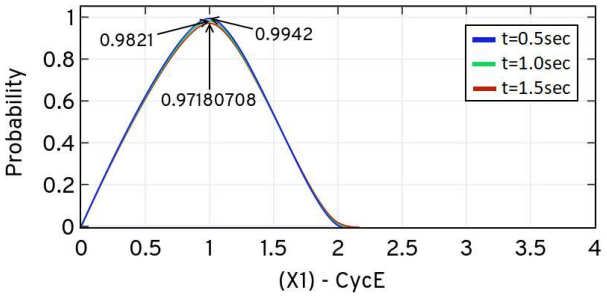

Fig (3). Probability of CycE over t_f

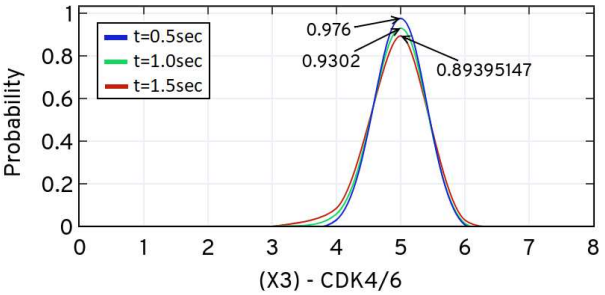

Fig (4). Probability of CDK4/6 over t_f

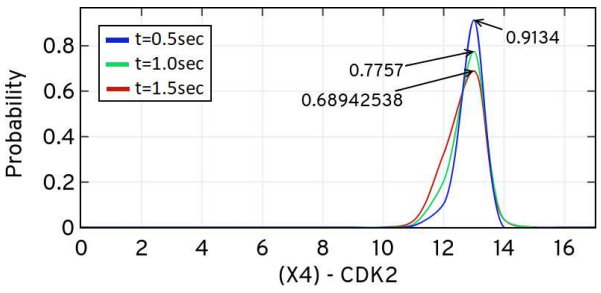

Fig (5). Probability of CDK2 over t_f

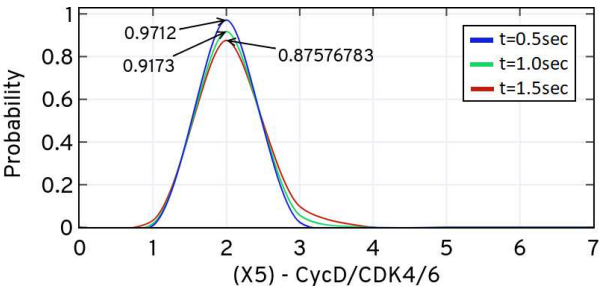

Fig (6). Probability of CycD/CDK4/6 over t_f

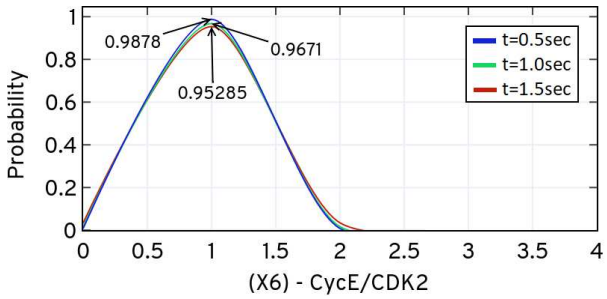

Fig (7). Probability of $\mathbf{CycE/CDK2}$ over t_f

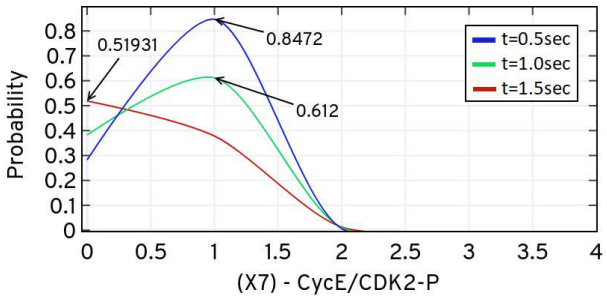

Fig (8). Probability of $\mathbf{CycE/CDK2-P}$ over t_f

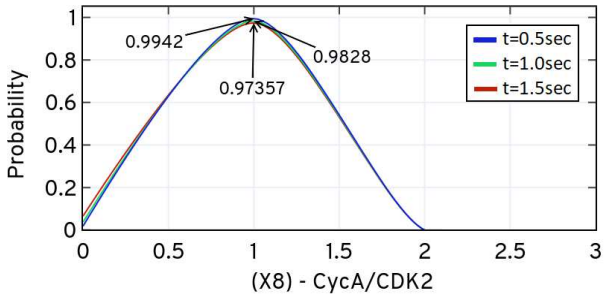

Fig (9). Probability of $\mathbf{CycA/CDK2}$ over t_f

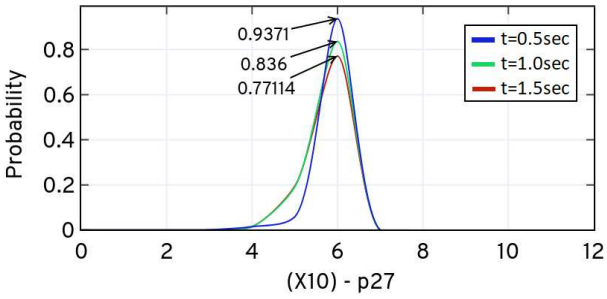

Fig (10). Probability of $\mathbf{p27}$ over t_f

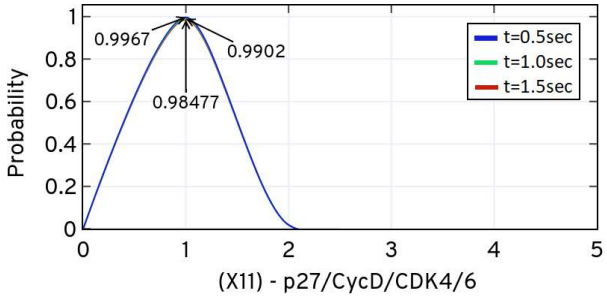

Fig (11). Probability of $\mathbf{p27/CycD/CDK4/6}$ over t_f

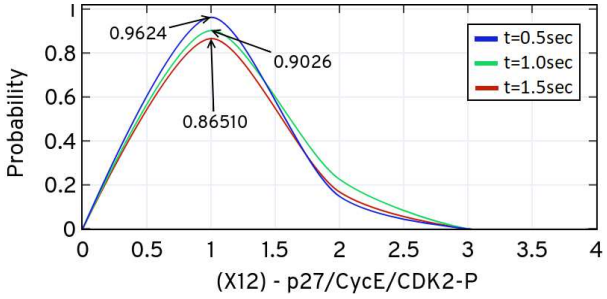

Fig (12). Probability of $\mathbf{p27/CycE/CDK2-P}$ over t_f

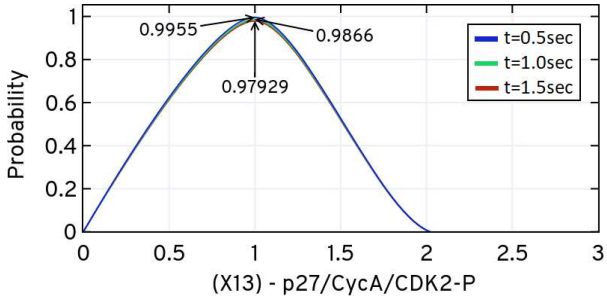

Fig (13). Probability of $\mathbf{p27/CycA/CDK2-P}$ over t_f

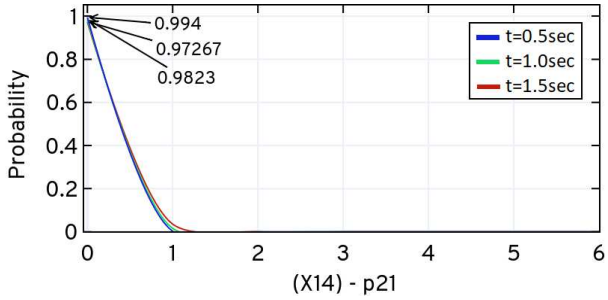

Fig (14). Probability of $\mathbf{p21}$ over t_f

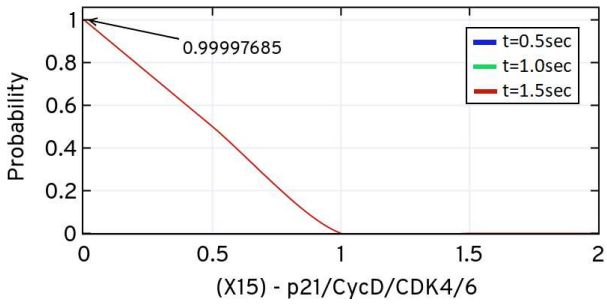

Fig (15). Probability of $\mathbf{p21/CycD/CDK4/6}$ over t_f

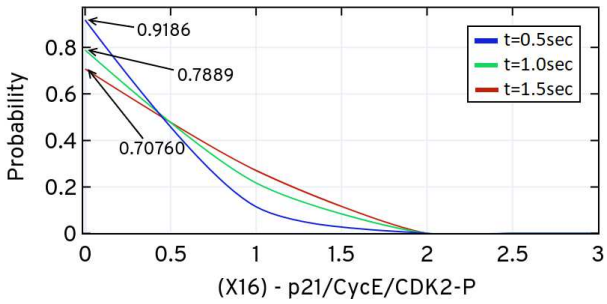

Fig (16). Probability of $\mathbf{p21/CycE/CDK2-P}$ over t_f

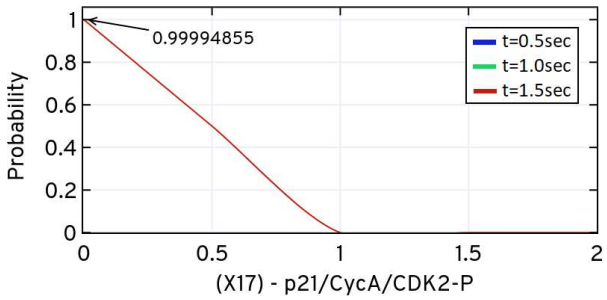

Fig (17). Probability of $\mathbf{p21/CycA/CDK2-P}$ over t_f

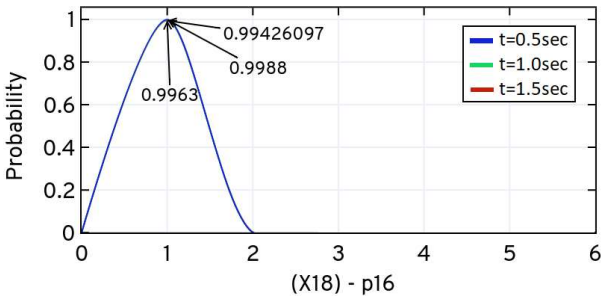

Fig (18). Probability of $\mathbf{p16}$ over t_f

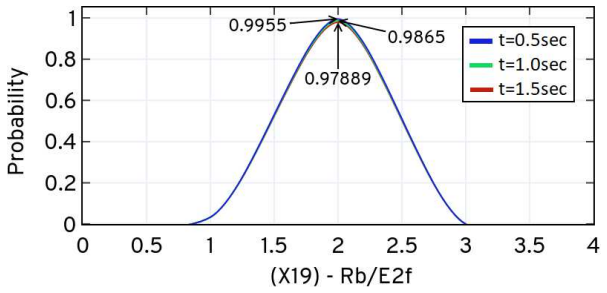

Fig (19). Probability of $\mathbf{Rb/E2f}$ over t_f

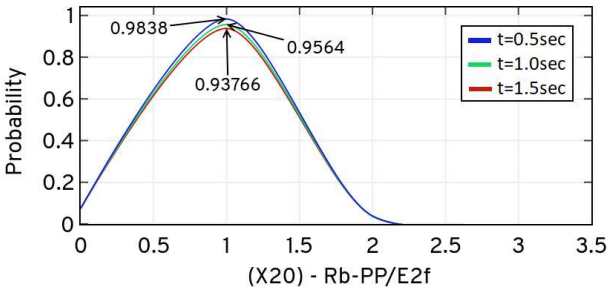

Fig (20). Probability of $\mathbf{Rb-PP/E2f}$ over t_f

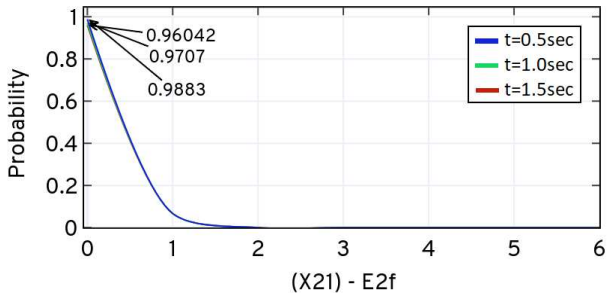

Fig (21). Probability of $\mathbf{E2f}$ over t_f

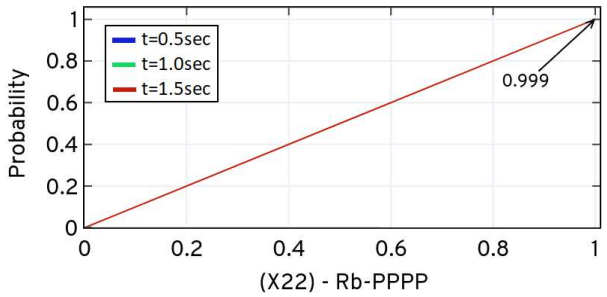

Fig (22). Probability of $\mathbf{Rb-PPPP}$ over t_f

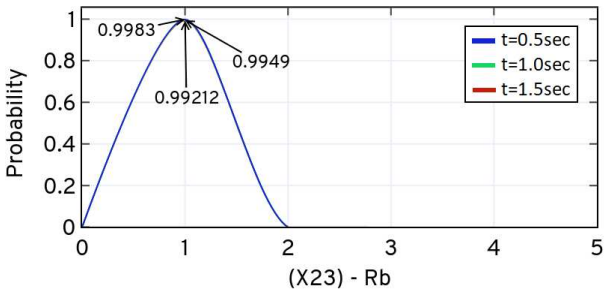

Fig (23). Probability of Rb over t_f

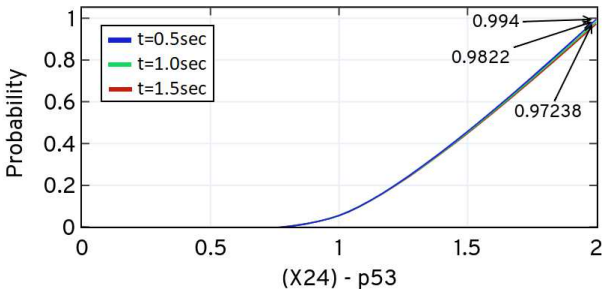

Fig (24). Probability of p53 over t_f

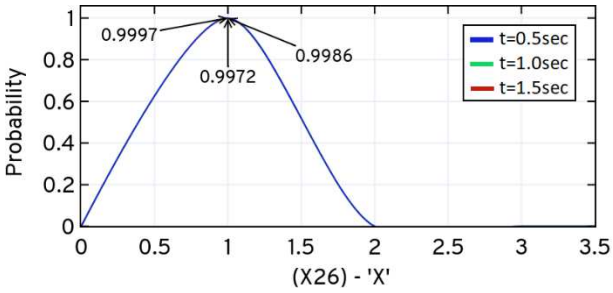

Fig (25). Probability of 'X' over t_f

Figure 14. Conditional probability of the G1/S model evaluated at $t_f = 1.5$ *sec*, $t_{step} = 0.1$ using ISP $LOLAS$.

Discussion

In this section, we discuss ISP performance based on speed and accuracy of the expansion, domain size and accuracy of the solution by comparing with other existing methods.

Comparison with other methods

An approximation of 10^{-5} is considered to find the approximate number of realisations required by the SSA for 10^{-4} global error. Realisations were computed until the difference is less than 10^{-4} between the known distribution and the empirical distribution.

Therefore, approximately 10^6 and 10^5 runs were required to obtain the right distribution for catalytic and dual enzymatic reaction network, respectively. In catalytic system, we observe (see Table 5) that both versions of ISP are faster than the $OFSP$ of *r-step reachability* and the SSA of sliding windows, and the improvement was attributed to the $LOLAS$ having fewer states and less computational time than $OFSP$ method and with better accuracy at t_f. Similary, the ISP was much faster than the SSA as the total number of realisations required to have an empirical distribution while the error at t_f was ≈10 times more than the domain produced by the ISP. Whereas, in dual enzymatic, we observe (see Table 5) that both versions of ISP are faster than $OFSP$ of *r-step reachability* and the SSA of sliding windows; the improvement is attributed to both the variants of ISP having an efficient domain with a small approximation error and less computational time than that of $OFSP$ method and with better accuracy at t_f. Similarly, both variants of ISP were much faster than SSA as the total number

of realisations required to have an empirical distribution with the error at t_f is ≈12 times more than the domain produced by ISP.

Table 5. Comparison of the solution of the catalytic reaction system based on ISP, $OFSP$ and SSA.

$t_f = 0.5$, $t_{step} = 0.01$	ISP		$OFSP$	SSA
	LAS	$LOLAS$		
Catalytic reaction system ($t_f = 0.5$, $t_{step} = 0.01$)				
Run-time (sec)	4677	2706	8767	17428
Domain at t_f	14666	13089	14665	10^6 Runs
Expansion time	0.5	0.5	0.5	-
Error at t_f	$1.865e-05$	$1.532e-05$	$1.917e-05$	$\approx 9.81 \times 10^{-3}$
Dual enzymatic reaction system ($t_f = 2.0$, $t_{step} = 0.01$)				
Run-time (sec)	2386	1614	2804	6374
Domain at t_f	8282	8296	8266	10^5 Runs
Expansion time	2.0	2.0	2.0	-
Error at t_f	$7.470e-05$	$5.953e-05$	$1.060e-04$	$\approx 9.94 \times 10^{-3}$

We also compared the error at t_f to note the efficiency of the solution. As seen in the results the increase in step error in $OFSP$ affected the solution at t_f. Figure 15 (see Fig (a) and (b)), shows a comparison of the ISP (LAS and $LOLAS$) with $OFSP$ on the basis of the approximation error at t during the expansion of the catalytic and dual enzymatic reaction network, respectively. Addressing the step error in ISP and the selection of the probable states results in an efficient solution at t_f as compared to $OFSP$.

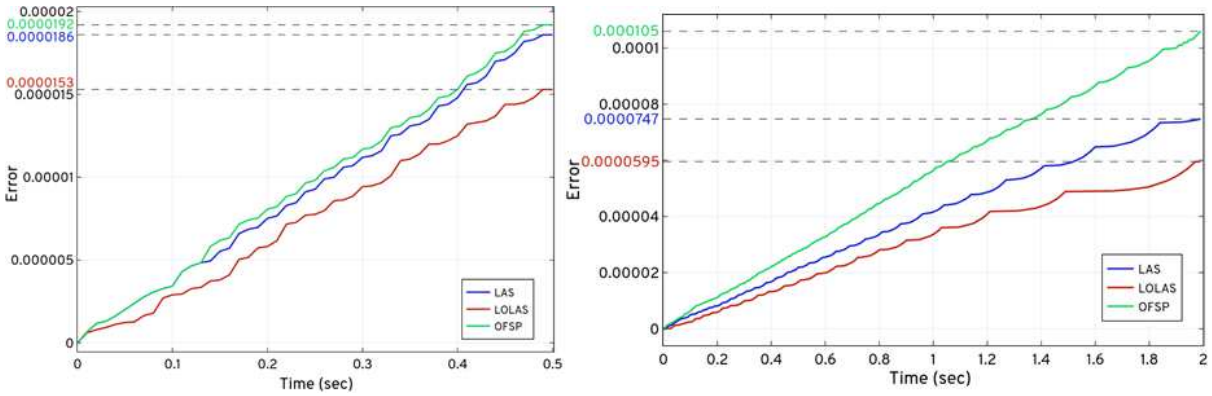

Fig (a) Fig (b)

Figure 15. The comparison of ISP (LAS and $LOLAS$) with $OFSP$ based on the solution of the catalytic and dual enzymatic reaction networks.

The typical firing nature of reactions in catalytic system makes them stiff and; therefore, the selection of states becomes difficult for approximation. This is due to some species in the system tending to increase abruptly in the population while others do so very slowly because the kinetic parameters ($k_1 = 1$, $k_2 = 1000$, $k_3 = 100$) have large differences and this triggers the reactions at different rates. Reaction R_1, is categorised as a *slow* reaction in the network and that affects the fast reaction, R_2. In the computation results of Table 5, the ISP identified that only 13089 probable states were required to solve the system up to t_f, this saves computational time (see Figure 16) compared to $OFSP$ and SSA, as well as improves the

accuracy of the solution. In $OFSP$, applying the compression at every step or after a few steps is still computationally expensive for a model like the catalytic reaction system, as seen in Table 5 and Fig (a) of Figure 16.

The network shown in Figure 8 consists of two enzymatic reaction system interlinked that transforms species S and P into each other via the other species, making the system stiff in nature and therefore, the selection of states becomes difficult for approximation. This is due to some species (S and E_1) in the system tending to increase in population abruptly while others very slowly, because some of the kinetic parameters ($k_1 = k_4 = 4$, $k_2 = k_5 = 5$) have large differences from other kinetic parameters ($k_3 = k_6 = 1$) and this triggers the reactions at different rates. Reaction R_1, categorised as the fastest reaction of the network that affects species S, C_1 and E_1 is followed by other reactions involving other species. From the computation results in Table 5, the $ISP\ LAS$ identified that only 8282 probable states and $ISP\ LOLAS$ identified that only 8296 probable states are required to solve the system up to t_f which saves the computational time (see Figure 16), compared to $OFSP$ and SSA, as well as improves the accuracy of the solution. In $OFSP$, applying the compression at a defined step or after a few steps is still computationally expensive for models like dual enzymatic reaction system, as seen in Table 5 and Fig (b) of Figure 16**.**

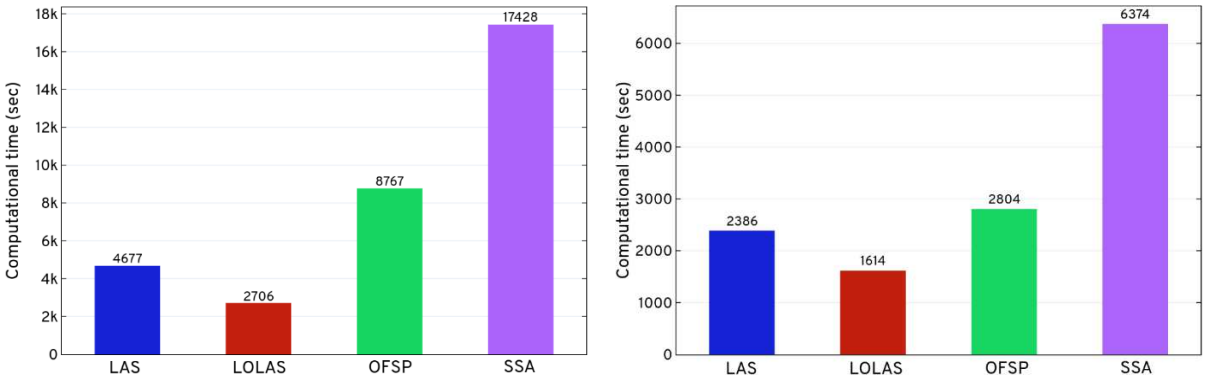

Fig (a) Fig (b)

Figure 16. The comparison of ISP (LAS and $LOLAS$) with $OFSP$ and SSA by computational time. All methods were applied to the catalytic and dual enzymatic reaction network, respectively, that was previously integrated in Experimental results section.

This also leads to the justification that the total computation effort required at every step when compressing the number of states up to t_f is approximately equal to the total computation effort required when the compression is applied in the gaps in some steps on a set of states up to t_f. Moreover, the state-space will remain the same, at t_f regardless of when the compression is applied.

A comparison of the computational times in Table 5 shows that both versions of ISP are significantly faster than the other methods. Figure 17 shows the CPU utilisation (%) of $LOLAS$ and $OFSP$ with respect to run-time (minutes). The dedicated throughput (see SI 1.1) between EC2 and EBS was not used for solving the model. The average CPU exertion is about 60%, which is a considerable workload for such a configuration for a given model. The expansion and approximation started when CPU use was at ≈1.6422% in catalytic reaction system and ≈1.23% in dual enzymatic reaction system, at $t = 0$ *sec* and increases up to 60.0% and then goes down to zero at t_f.

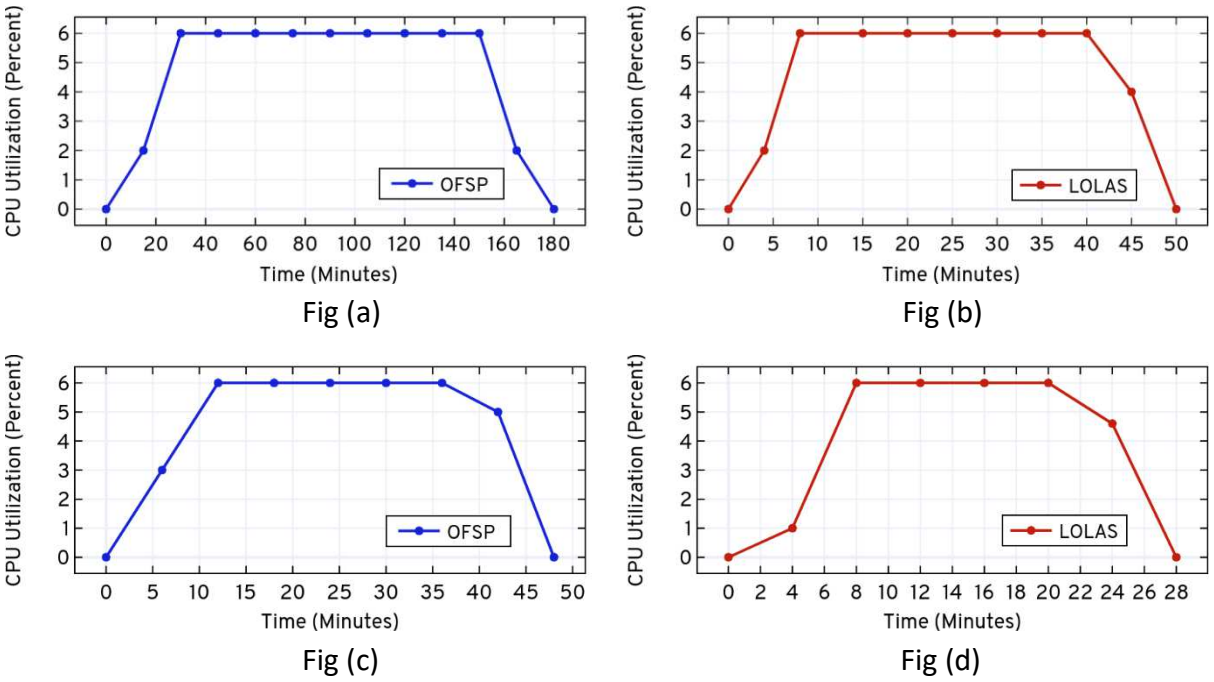

Figure 17. AWS® CPU utilisation percentage, when the catalytic reaction system is solved up to $t_f = 0.5$ *sec* and dual enzymatic reaction system solved up to $t_f = 2.0$ *sec* using $OFSP$ and $LOLAS$. The performance analysis was carried out using CloudWatch® (Statistic: Average, Time Range: Hour, Period: 5 Minutes).

Conclusions

This paper introduced a novel approach ISP to model biochemical systems that address the performance as well as the accuracy problems for the solution of the CME. Variants of ISP (LAS and $LOLAS$) provide systematic ways of expanding the state-space as long as all probable states are not added into the domain, up to the desired t_f. We have effectively demonstrated the effectiveness of our methods with several experiments using real biological models: the catalytic reaction system, dual enzymatic reaction system, and G1/S model (large model). The results and the response of the algorithm shows improvements in the way how different size of biological networks can be modelled and even state-space of 3409900 nodes (see Table 3) carrying states up to ≈3.5 million can be explored in a reasonable time. The results also show that the domain laid out by ISP had an optimal order and was successful in finding probable states of the system while maintaining good accuracy and computational timing.

We compared the results of the two popular methods: $OFSP$ (*r-step reachability*) and SSA (τ leaps adaptive) based on their accuracy and efficiency. In the comparative study results, we have seen that ISP outperformed the other methods, in case of computational expense, accuracy as well as projection size. The ISP was more effective in terms of predicting the behaviour of the state-space of the system and in performance management, which is a vital step towards modelling large biochemical systems. Unlike other methods, the ISP keeps the lowest states probabilities in the bunker without removing (as removed in $OFSP$) them before its calculation (as removed without calculation in $FSP\ GORDE$) and then computes the probabilities at t without computing large numbers of realizations (as done in SSA).

The diverging nature of the ISP response with respect to $OFSP$ in Figure 19 also showed that the solution improved with t and at a higher t_f. For example, in the large model in case study 2, the computation time was 1372 *sec* and the solution was $3.52\mathrm{e}-06$ at t_f, which was lower compared to the results from the small model (catalytic reaction system). This also shows the compatibility of ISP with the distinct size of the biochemical models.

These examples showed that the ISP is a very promising technique for system's biology. For stiff models, such as the G1/S and *Candida albicans* models, the nature of the ISP yielded plenty of information and opened opportunity for stochastic analysis of large models. It is often sufficient to employ the ISP to compute the probabilities of the species up to the required time. One could also use ISP to effectively conduct *robustness* and *sensitivity analysis* on the dynamics of biochemical systems and to keep track of what reactions are more active in the system at a particular time. It also determines the complexity of the system by defining the bounds with number states and keep track of the nested state-space patterns (called as ISP model blueprint) that were updated at the end of each step. Outlining the patterns of expansion of states to predict the projection folds can be used for updating of the new states.

We anticipate that the current structure of the ISP variants can be efficiently used for different classes and varieties of biological systems, and for computing the configurations with many reactions, as long as the notable part of the state-space density was present between $Bound(Z)_{lower}$ and $Bound(Z)_{upper}$. When there was a high probability of the molecular population of the species undergoing several excursions in a fraction of time, then ISP use small t_{step} to capture the moments. Such computations were still challenging in the expansion phase for typical models but can be addressed more closely in combination with the second part of the solution to the CME, i.e. the approximation phase. There are several methods available to address these challenges.

Approximation methods, such as the *Krylov sub-space*, can be used to effectively compute the matrix exponential times of a vector. It was mathematically attractive to aggregate the states or decompose the large sparse matrix into a small dense matrix using the *Krylov sub-space* but they may not be computationally efficient for many in the absence of an efficient domain. On other hand, the performance can also be enhanced by employing the fast math functions compatible with the multicore environment. We have clearly outlined the core ideas behind the ISP variants by highlighting the differences and similarities between them and other methods that cover the computational and theoretical considerations that are essential before any of the approximation methods becomes feasible for an efficient CME solution.

Methods

To understand and predict the dynamics of state-space response in biochemical systems, we have developed an analytical numerical method called ISP that integrates the reactions propensities describing the Markovian processes through set of nodes governing set of states of the system. The two variants of ISP as LAS and $LOLAS$ consisting of several modules that incorporate set of inputs and functions within several compartments. Figure 18 depicts all the modules of the ISP, and integrated form is discussed later in Table 7, 8.

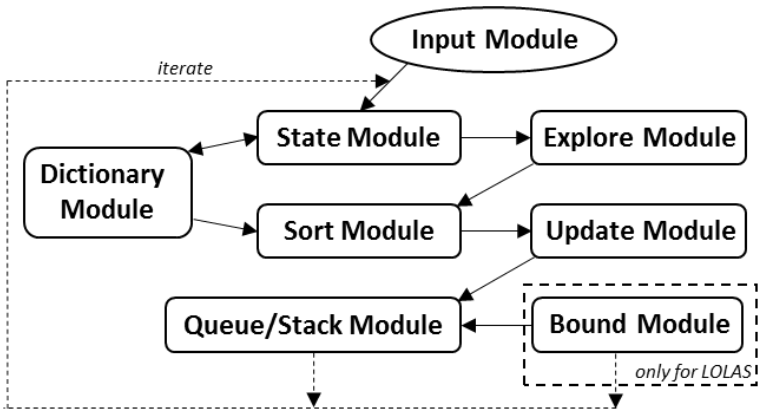

Figure 18. Comprehensive ISP method flow chart. A description of the modules (*steps*), sub-modules and the list of components are discussed in SI 5.

These modules and sub-modules constitute the ISP method such that they track key changes in the components that follows change in the reaction propensities by population and activation of the species and describe the dynamics of the biochemical system. The method also permits the time form quantification of state-space based on the size, dimension of the model.

The ISP states expansion strategy is based on the standards of *Artificial Intelligence* (AI) [38–41], state-space search and relative successor (S_{uc}) operator or function which perform operations on inputs. AI is the field of study of intelligent agents [42] of a system that perceive and take action to successfully achieve the goals. Most of the problems can be formulated as searches and solved by reducing to one of searching a graph or a tree. To use this approach, we must specify the successor operator which defines the sequence of actions leading from initial to goal state at different time interval that leads to the solution.

In terms of AI, we will now define the *state-space* as a set of states in the system we can get to by applying S_{uc} to explore new states of a biochemical network. S_{uc} can be applied explicitly, which maps the current state to the output states or defined implicitly that act on current state and transform into a new state. In the *state-space* graph for biochemical networks, we will not define the goal state (or end state) explicitly but it is required to be defined by S_{uc} implicitly in intervals based on nature (*fast, slow, reversible and irreversible*) of reactions in the system and duration of expansion to introduce the stochasticity of the system. This should systematically expand the state-space from $\mathbf{X}_K$ at t to $\mathbf{X}_{K+1}$ at $t+1$ by going through each node $\mathbf{n}_J$ at depth $\mathfrak{d}_l$ of the Markov chain graph to evaluate the Markov processes, which aims to occupy most of the probability mass during $[\mathbf{X}_K + \mathbf{X}_{K+1}]$, and can be solved for probability distribution at $t+1$.

Let $\mathbf{X}_J$ be the finite set of states and $G_{mc} = (\mathbf{X}_J, V_\mu)$ be the Markov chain graph on $\mathbf{X}_J$ associated with $A = [a_{i,j}]$, given X_0 as the initial state and $\mathbf{X}_K$ as the set of the explored state, where $X_0 \in \mathbf{X}_K$ then implicit successor is defined as,

$$S_{uc} \longrightarrow V_\mu(\mathbf{X}_K(t)), \tag{44}$$

defines the new states of the system, where, V_μ is the set of stoichiometric vectors v_μ function defining the state transitions from any present state $X_i \in \mathbf{X}_K$ to new state $X_{i'} \notin \mathbf{X}_K$. The

sample space in the graph contains the unique state of the system stored in a transition matrix, which satisfy the condition in Eq. **Error! Reference source not found.**. This transition matrix is a compressed row format (CSR) [43,44] based on index of row ⟶ column delimited by commas generating the dictionary $Dict$ of the model which defines the transitions between nodes in the state-space and mapping of states. Through S_{uc}, we can know nothing more than the neighbours (child nodes) of the current node (states reachable through a single reaction), then we consider these neighbours (child nodes) as our only goal states and there can be many in numbers. In such a situation, we call our search trails as *blind* or *uninformed search*. In following section, we will work on the infrastructure of an *uninformed search* to define the type of data structure we will be dealing with.

Infrastructure for searching

A data structure is required to retain the search track in the graph for *problem state-space* expansion. For each node, N_i, of the tree, we create a structure consisting of five elements:

(1) N_i.State: represents state X_i in the state-space corresponding to N_i;

(2) N_i.Parent: represents the parent node of child node N_i;

(3) N_i.Depth: represents the depth of state state X_i;

(4) N_i.Cost: represents the cost $\mathfrak{C}_{N_i,N_i'}$ of the transition from N_i to N_i' in the state-space;

(5) N_i.Action: represents the action applied via S_{uc} on parent node to reach N_i.

To explore the new states in the system, we will consider the initial state $state(N_1) = (X_0, \mathfrak{d}_l)$ as input to the successor, S_{uc}. Once the expansion is initiated, the $Dict$ will temporarily (in run-time) store the information for the transition from one node to another in the state-space that binds to the reaction propensities a_μ. This shift is denoted by an arrow ⟶, which shows multiple transitions from the parent nodes to child nodes containing the end state. The set of nodes $\mathbf{n}_J = \{N_1, N_2, \ldots . N_{S^{\tilde{N}}}\}$ is a data structure that incorporates the Markov chain graph G_{mc}. We will explore all the nodes that store the set of states $\mathbf{X}_K$ as well as some additional information about the state, such as depth and transition cost, from one state to another in the system. If a set of $states(\mathbf{n}_J) = \mathbf{X}_J$, then $\mathfrak{C}_{1,N_i'}$ is the transition cost to reach $state(N_i') = X_{i'}$ from $state(N_1) = X_1$ and $depth(\mathbf{n}_J) = \mathfrak{d}_l$ defines the depth of the set of nodes in G_{mc}, then the standard relation between a set of nodes and a set of states is given by $\mathbf{n}_J = (\mathbf{X}_J, \mathfrak{d}_l)$ or $\mathbf{n}_J = (\mathbf{X}_J, \mathfrak{d}_l, \mathfrak{C}_{N_i,N_i'})$ and the standard relation between a single node and a single state is given by $N_i = (X_i, \mathfrak{d}_l)$ or $N_i = (X_i, \mathfrak{d}_l, \mathfrak{C}_{N_i,N_i'})$ if the transition cost is considered.

For example, Fig (a) of Figure 19 shows the Markov chain graph, G_{mc}, with $\mathfrak{n}_J = 10$, $\mathfrak{d}_l = 4$ and its equivalent tree $\mathbb{K}$ is shown in Fig (b) of Figure 19 with $\mathfrak{n}_J = 15$, $\mathfrak{d}_l = 5$. In the tree nodes $N_1 = N_{11} = N_{12}$ carries the same state, X_1 at $\mathfrak{d}_l = 1, 2$ and 3, respectively, where walk $N_2 \to N_{11}$ and $N_7 \to N_{12}$ represents the backward reaction of the forward reaction represented by walk $N_1 \to N_2$ and $N_1 \to N_7$, respectively.

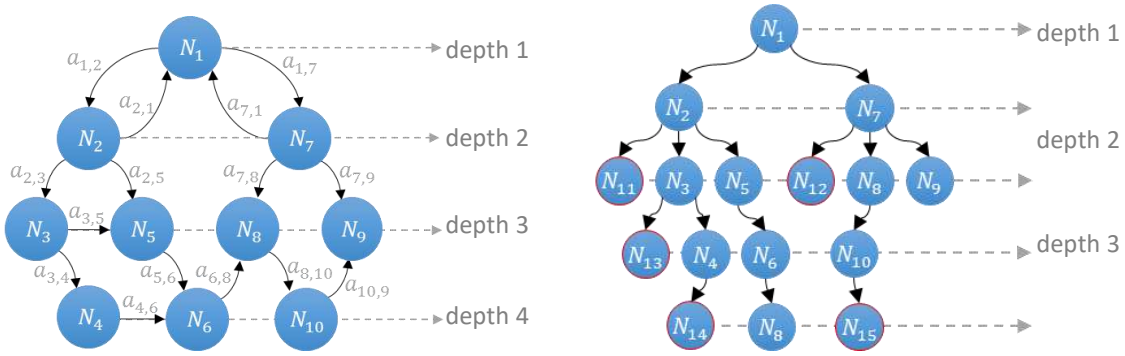

Fig (a) Fig (b)

Figure 19. **Represents the Markov chain graph and its equivalent tree.** Fig (a) depicts Markov chain graph (G_{mc}) with n_J nodes carrying $\mathbf{X}_J$ states and the arcs showing transitions between them and all together forming a Markovian process, and Fig (b) depicts equivalent tree $\mathbb{T}$ of G_{mc} as DAG representing state-space of the system.

The set of nodes with states are represented as

$$\mathbf{n}_{1,2,\ldots.10} = \left(\mathbf{X}_{1,2\ldots\ldots K}, \mathfrak{a}_{1,2,3,4}\right) \text{ or} \tag{45}$$

$$\mathbf{n}_{1,2,\ldots.10} = \left(\mathbf{X}_{1,2\ldots\ldots K}, \mathfrak{a}_{1,2,3,4}, \mathfrak{C}_{N_i,N_i'}(min)\right) \tag{46}$$

In general, the transition cost, $\mathfrak{C}_{N_i,N_i'}$, is defined as:

$$\mathfrak{C}_{N_i,N_i'} = a_{1,2} + a_{2,3} + \cdots + a_{N-1,N}. \tag{47}$$

$\mathfrak{C}_{N_i,N_i'}$ is the summation of all the propensities a_{μ} of the R_M reactions that takes the system to its final state. For example, $\mathfrak{C}_{N_1,N_{10}}$ to expand to $state(N_{10}) = X_{10}$ of Fig (a) of Figure 23 is given by

$$\mathfrak{C}_{N_1,N_{10}} = \begin{cases} \textbf{\textit{Path}}\ \mathbf{1}\text{:}\ a_{1,2} + a_{2,3} + a_{3,4} + a_{4,6} + a_{6,8} + a_{8,10} \\ \textbf{\textit{Path}}\ \mathbf{2}\text{:}\ a_{1,2} + a_{2,3} + a_{3,5} + a_{5,6} + a_{6,8} + a_{8,10} \\ \textbf{\textit{Path}}\ \mathbf{3}\text{:}\ a_{1,2} + a_{2,5} + a_{5,6} + a_{6,8} + a_{8,10} \\ \textbf{\textit{Path}}\ \mathbf{4}\text{:}\ a_{1,7} + a_{7,8} + a_{8,10} \\ \textbf{\textit{Path}}\ \mathbf{5}\text{:}\ 0, \text{if not reachable} \end{cases}$$

If these are the possible paths for the expansion that expands $\mathbf{X}_K$ at every iteration then $\mathfrak{C}_{N_1,N_{10}}(min)$ will be defined by the only path that has the lowest $P^{(t)}(\mathbf{X}_K')$. This can be generalised as follows:

$$\mathfrak{C}_{N_i,N_i'}(min) \propto \frac{1}{P^{(t)}(\mathbf{X}_K)}, \tag{48}$$

which means that in order to have a minimum cost of the expansion for the optimal domain $\mathbf{X}_K$ at least one path should have states with high probabilities for $\mathbf{X}_K$ and; therefore, it is beneficial to follow the path with $\mathfrak{C}_{N_i,N_i'}(min)$, which leaks the minimum probabilities of the system.

For large biochemical models there exists infinite cases when the node is unreachable from the initial or another node, and such cases are ignored when $\mathfrak{C}_{N_i,N_i'}(min) = \{Path\text{: } 0\}$

because some probabilities are always dropped in the approximation. Therefore, $\mathfrak{C}_{N_i,N_i'}(min)$ as defined by the lowest $P^{(t)}(\mathbf{X}_K')$ is strictly limited to,

$$P^{(t)}(\mathbf{X}_K) > \mathfrak{C}_{N_i,N_i'}(min) > 0, \tag{49}$$

Upon expanding the root node N_1, we expand the child nodes carrying new states, and then the child-child nodes are explored. The walk between nodes $N_i \xrightarrow{V_\mu(\mathbf{X}_K(t))} N_{i+1}$ is defined by dictionary $Dict$ and this represents the occurrence of R_M reactions through M elementary channels. For Fig (a) of Figure 19, the typical form of dictionary is given below:

$$\begin{aligned} D = ([1 \longrightarrow 2,7], [2 \longrightarrow 1,3,5], [3 \longrightarrow 4,5], [4 \longrightarrow 6], [5 \longrightarrow 6], [6 \longrightarrow 8], \\ [7 \longrightarrow 1,8,9], [8 \longrightarrow 10], \ [9 \longrightarrow Nil], [10 \longrightarrow 9]), \end{aligned} \tag{50}$$

and is indexed with the propensities, $[a_{i,j}]$, for all the R_M reactions. As the propensities are changing by $\Delta a_{i,j}$, we will consider the recent values of $a_{i,j}$ in every iteration of ISP that corresponds to the reactions involved. To make the $\mathfrak{C}_{N_i,N_i'}(min)$ feasible for any type of biochemical system (*stiff, non-stiff*) for capturing probable states, it is important to consider the cost of expansion for small t_{step} (time step). This may be because there are some cases when $\mathfrak{C}_{N_i,N_i'}(min)$ to reach two or more different child nodes are equal or very close to each other. In addition, we intend to expand the state-space in the direction of carrying states with high probability mass. To achieve this, we need some provision made to treat or convert our *uninformed search* to an *informed search* infrastructure at run-time to have intuitive knowledge beyond our reach. Figure 20 shows the limits of our visibility in the state-space.

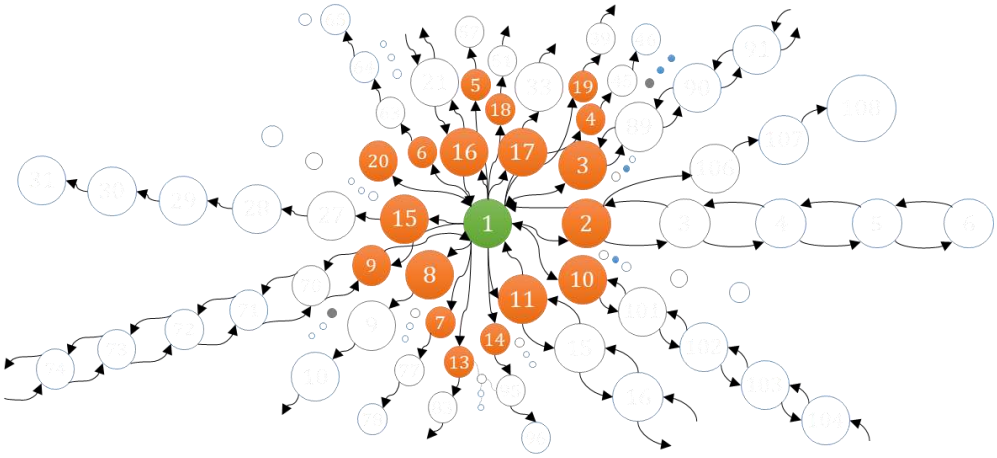

Figure 20. Limits of our visibility in the state-space before expansion. Visualised by a Markov chain graph, where ● is the initial node and ● nodes are directly reachable nodes from the initial node when exactly one $\boldsymbol{R_M}$ occurs. When a further $\boldsymbol{R_M}$ occur, the system jumps to other ○ nodes.

Consequently, it is important to track the reactions having high propensity function values. In such, cases it is difficult to determine the direction of the expansion; therefore, in following section (a), we first develop the post successor function on Bayes' theorem [30,31] to prioritise the expansion direction based only on those reactions that can be triggered at a particular time point and then, in sections (b) and (c), we will layout the direction strategy with the depth and bounds of the expansion.

(a) Bayesian Likelihood Node Projection Function

Bayesian methods [31,45] are based on the principle of linking prior with the posterior probability through Bayes' theorem [30,31]. For an event, the posterior probability is the improved form of the prior probability, via the likelihood of finding factual support for a valid fundamental hypothesis. Therefore, we will employ the standards of Bayes' theorem to develop a function targeted to advocate the quality of the expansion based on R_M reactions active in the network at any particular moment. For a concise definition for the purpose of fundamentals, refer to SI 6.

To improve the quality of expansion through a projection function, one may find useful to remove the set of states having low probabilities before calculation of Eq. (3); however, this will compromise the accuracy as the step error will increase at every t. Moreover, it will greatly affect the solution, as defined at t_f (at which solution is required), for large dimension systems having large state-spaces, as the step error will be much higher due to dropping of probabilities without solving Eq. (3). In large systems, any species may change its behaviour after a certain number of firing of reactions triggering inactive reactions in the network that will affect the probabilities of the states. If the change in behaviour increases the probabilities of certain states, then removing them in an earlier stage is not suitable.

Through the *Bayesian Likelihood Node Projection* $(BLNP)$ function, we seek to predict the posterior probability based on the parent state's probability and calculate the likelihood of the occurrence of reactions that will take the system from the present state to the future state. Through $BLNP$ we can capture a sense of knowledge about the situation of the system that will help us to make better predictions about the future state and still keep good accuracy of the solution with optimal domain.

In such a situation it is important to decide the direction of the expansion when choosing the future state of the system as any reaction can occur and take the system to any new state. To understand this situation more closely on a node level, we assume a Markov chain graph as shown in Figure 21 of this system having almost the same number of species count. In Figure 22, the expansion is at intermediate position as the initial state $state(N_0) = X_0$ is already expanded and now the expansion of $state(N_2) = X_2$ is to be undertaken. To calculate the likelihood of the occurrence of reactions R_1, R_2, R_5, we consider the propensities $a_{i,j}$ as a parameter and $\Delta a_{i,j}$ depends on the kinetic parameter of the reaction. To assign weight to our belief, we deduce a function that will calculate the probability of occurrence of reactions and prioritise the expansion in order from reactions resulting in states with high probabilities to reaction giving states with low probabilities. It is important to note that none of the probabilities will be removed before the calculation of Eq. (5). With this function the likelihood of occurrence of R_M can be computed.

We consider each node as a junction of the prior reactions $\{R'_1 \dots \dots R'_M\}$ with propensities $\{a'_{1,N} \dots \dots a'_{N',N}\}$ having prior likelihood values $\{b'_{1,N} \dots \dots b'_{N',N}\}$ and future reactions $\{R_1 \dots \dots R_M\}$ with propensities $\{a_{1,N'} \dots \dots a_{N,N'}\}$ having likelihood values $\{b_{1,N'} \dots \dots b_{N,N'}\}$, as given in Figure 21.

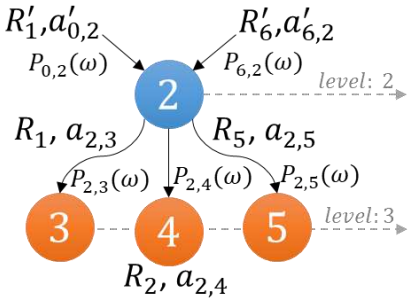

Figure 21. Current $state(N_2) = X_2$, and future $states(N_{3,4,5}) = (X_{3,4,5}, d_{l=3})$ with corresponding reactions R_1, R_2, R_5 and assumed propensities $a_{2,3} = 38, a_{2,4} = 39, a_{2,5} = 40$, respectively, at any time t, given $b_{0,2} = 0.4871, b_{6,2} = 0.5128$.

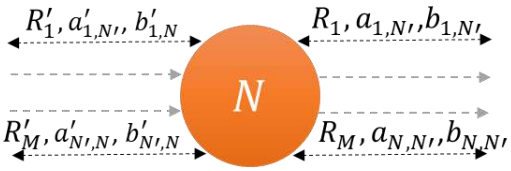

Figure 22. Node N as a junction of forward and backward reactions R_M, where $a'_{1,N'}, \ldots, a'_{N,N'}$ are propensities of the prior reactions' and $b'_{1,N}, \ldots, b'_{N',N}$ are the likelihood of the prior reactions.

To calculate the likelihood of the reactions, it is important to have prior information about the occurrence of reactions. If the expansion is to be done at initial node $state(N_0) = X_0$ (at level 1) then the prior likelihood value $b'_{N,N'}$ is considered as the initial probability or as ≈1. Once the initial node is explored, we can calculate the likelihood of the reactions inductively. To calculate the probabilities $b_{1,N'}, \ldots, b_{N,N'}$ of the occurrence of $R_1, \ldots, R_M$, we first calculate the weighted probabilities $P_{N,1}(\omega), \ldots\ldots, P_{N,N'}(\omega)$ of a system leaving any state by:

$$\frac{a_\mu(X - v_\mu)}{\sum_{\mu=1}^{M} a_\mu(X - v_\mu)} = \frac{\text{Propensity of } R_M \text{ reaction leaving state } X_i \text{ at } \mathfrak{d}_l}{\text{Sum of propensities of all the reactions leaving state } X_i \text{ at } \mathfrak{d}_l} \tag{51}$$

and multiply with the prior probability $b'_{1,N} \ldots\ldots b'_{N',N}$ of the system. This will calculate the likelihood inductively in exactly R_M that is responsible to shift the system to present $state(N_i) = X_i$ at t, leading to a function,

$$b(N_{N1,..NM} | b'_{1,N....N',N}) = \frac{a_\mu(X - v_\mu)}{\sum_{\mu=1}^{M} a_\mu(X - v_\mu)} * b'_{N',N}(X - v_\mu)' \tag{52}$$

where,

$$P_{N,1....N,N'}(\omega) = \frac{a_\mu(X - v_\mu)}{\sum_{\mu=1}^{M} a_\mu(X - v_\mu)}, \tag{53}$$

$$b(N_{N1,..NM} | b'_{1,N....N',N}) = P_{N,1....N,N'}(\omega) * b'_{N',N}(X - v_\mu)'. \tag{54}$$

Once $b(N_{N1,..NM} | b'_{1,N....N',N})$ is calculated for all the adjacent nodes, the values are arranged in descending order. Every value is bound to one reaction and represents the likelihood of the

occurrence of that reaction that takes the system from the present node to the child nodes. Based on likelihood values (highest to lowest) the corresponding reactions are considered one by one and labelled as *true* events for expansion. For example, if system has R_1, R_2, R_3 reactions that bound to $BLNP$ likelihood values in order from highest to lowest, respectively, then it is considered as three events of the system that takes the system to new state. When R_1 is considered for expansion, R_2 and R_3 are labelled as *false* events and R_1 as the *true* event. When second highest $BLNP$ likelihood value is considered, which is for R_2, then it is labelled as a *true* event and the others, R_1, R_3 are labelled as false. Similarly, the last and lowest $BLNP$ likelihood value is for R_3, which is labelled as a *true* event and the others as *false* events. All states are added in the domain in order from the 1st *true* event to the 3rd true event. The Eq. (54) of probabilities $b(N_{N1,..NM}|b'_{1,N....N',N})$ is what we call a $BLNP$ function.

In Figure 27, Markov chain tree, for selection present at level 2 (assuming that the initial node is already expanded), we calculate the weighted probability of a system leaving $state(N_2) = X_2$ by:

$$P_{2,3}(\omega) = \frac{a_\mu(X - v_\mu)}{\sum_{\mu=1}^{3} a_\mu(X - v_\mu)} = 0.3247$$

similarly, $P_{2,4}(\omega) = 0.3333$ and $P_{2,5}(\omega) = 0.3418$.

At level 2, the conditional probability of the occurrence of reaction R_1 given the probability of occurrence of reaction R_1 at level 1 is given by:

$$b\big(N_{2,3}|b'_{0,2}\big) = \frac{a_\mu(X - v_\mu)}{\sum_{\mu=1}^{3} a_\mu(X - v_\mu)} * b'_{0,2}\big(x - v_\mu\big)',$$

and, similarly, the occurrence of reaction R_1 at level 2 given the probability of occurrence of reaction R_6 at level 1, is given by:

$$b\big(N_{2,3}|b'_{6,2}\big) = \frac{a_\mu(X - v_\mu)}{\sum_{\mu=1}^{3} a_\mu(X - v_\mu)} * b'_{6,2}\big(x - v_\mu\big)'.$$

If at level 1, $state(N_1) = X_1$ and at level 2, $state(N_2) = X_2$ are explored through R_1 then we say that this is a *true* event and temporarily consider other events *false* events with respect to the other reactions. Such a condition holds *true* for the other two cases, when, at level 1, $state(N_1) = X_1$ is explored through R_1 followed by an exploration of $state(N_2) = X_2$ through either by R_2 or R_5. Given $b'_{0,2}(X - v_\mu)'$ and $b'_{6,2}(X - v_\mu)'$, we will calculate the likelihood of all the R_M events, as given in Table 6. The likelihood values of future reactions cannot be equal as they are based on the probabilities of occurrence of prior reactions.

Table 6. Events with the likelihood of the future reactions. Where, *True* events define the expansion of nodes.

$\boldsymbol{b_{N,N'}}$	$\boldsymbol{N_{0,2}}$	$\boldsymbol{N_{6,2}}$	$\boldsymbol{b_{N,N'}}$(Value)	$\boldsymbol{R_{next}}$
$b(N_{2,3}\|b'_{0,2})$	True	False	0.1581	$R_{1,1}$
$b(N_{2,3}\|b'_{6,2})$	False	True	0.1665	$R_{6,1}$
$b(N_{2,4}\|b'_{0,2})$	True	False	0.1623	$R_{1,2}$
$b(N_{2,4}\|b'_{6,2})$	False	True	0.1709	$R_{6,2}$
$b(N_{2,5}\|b'_{0,2})$	True	False	0.1664	$R_{1,5}$

$b(N_{2,5}\|b'_{6,2})$	False	True	0.1752	$R_{6,5}$

From Figure 27 and Table 6, we can infer, based on the prior reactions for R_M, where $M = \{1,6\}$ as such that:

Case 1 (R_1): At level 2, if the prior reaction is R_1 and holds a *True* event for $N_0 \longrightarrow N_2$ then:

$$b(N_{2,5}|b'_{0,2}) > b(N_{2,4}|b'_{0,2}) > b(N_{2,3}|b'_{0,2})$$

as per $b_{N,N}$, and likelihood of occurrence of reactions will be in the order $R_5 > R_2 > R_1$.

Case 2 (R_6): At level 2, if the prior reaction is R_6 and holds a *True* event for $N_6 \longrightarrow N_2$ then

$$b(N_{2,5}|b'_{6,2}) > b(N_{2,4}|b'_{6,2}) > b(N_{2,3}|b'_{6,2})$$

as per $b_{N,N}$, and likelihood of occurrence of reactions will be in the order $R_5 > R_2 > R_1$.

There will be M number of cases (equal to elementary chemical reaction channels) if there are R'_M prior reactions in the system that bring the system to the current node and the value of the likelihood will change based on $b'_{N',N}(X - v_\mu)'$. The $BLNP$ function cannot be used standalone for expansion because it only assign weightage to direction for expansion. Therefore, in **Error! Reference source not found.**, we will derive the condition for our expansion strategies to work with the Markov chain graph state-space and define the criteria for the formation of bounds (domain formed at anytime t) with time. The $BLNP$ function with expansion strategies will choose the probable states in large biochemical systems where it is important to capture the moments at time t that define the behaviour of a system. $BLNP$ will be useful in identifying the most active reactions in the system while guiding the expansion towards the set of states with high probability mass.

(b) Latitudinal Search Strategy

We delve deeper into the first subroutine of ISP called the *Intelligent State Projection Latitudinal Search* ($ISP\ LAS$). Figure 23 manifest the infrastructure of the LAS strategy, showing G_{mc}, the $queue$ and the domain. The $queue$ data structure of LAS is based on $FIFO$ (First In, First Out) method. It assumes that oldest state added to the $queue$ is considered first. In particular, we will define and exploit the direction of expansion step-by-step based on intuitive knowledge (as discussed in section (a)) gained from probability of future reaction events and follow the conditions (as discussed in *Results* section). Furthermore, we show step-by-step how the nodes are explored, and states updated in the domain in I_{tr} iterations.

The states at level $\mathfrak{d}_l$ are expanded only after all the states at level $\mathfrak{d}_l - 1$ have been expanded, i.e. the search is undertaken *level-by-level* and depth $\mathfrak{d}_l$ increases in every I_{tr} iteration. In the case of networks with *reversible* reactions, the ISP condition will prevent LAS from returning to the state it came from and also prevent transitions containing cycles resulting in DAG with no repetition of any state whatsoever; however, the changes in propensities $a_{i,j}$ are validated. Verifying the explored states in $\mathbf{X}_K$ in iterations ensures that the algorithm completes and a deadlock in the state transition cycles cannot occur.

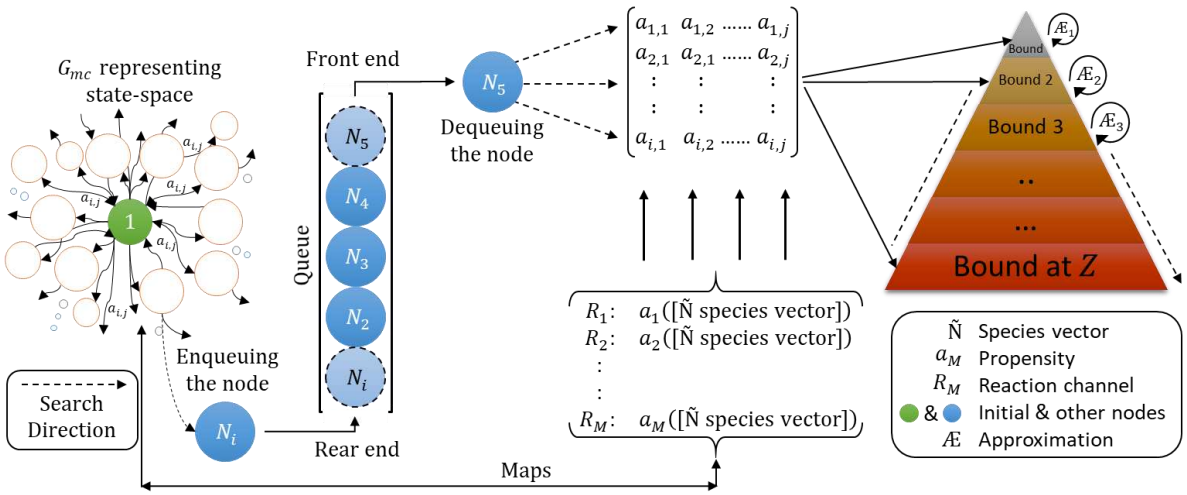

Figure 23. Infrastructure of the *Latitudinal Search* strategy, showing G_{mc}, the $queue$ and the domain.

The *time complexity* of LAS depends on the average transitioning factor T and depth $\bar{\mathrm{d}}_l$ and is given by (see SI 7 for detailed discussion),

$$1 + \mathrm{T}^1 + \mathrm{T}^2 + \cdots + \mathrm{T}^{\bar{\mathrm{d}}_l} + (\mathrm{T}^{\bar{\mathrm{d}}_l+1} - \mathrm{T}) = O(\mathrm{T}^{\bar{\mathrm{d}}_l+1}), \tag{55}$$

where,

$$\mathrm{T} = \frac{\text{Total no. of walk between different nodes}}{\text{Total no. of nodes explored}} \tag{56}$$

For the nodes at the deepest level $\bar{\mathrm{d}}_l$, all walks are valid except for the very last node which stores the end state of the system. Therefore, once the end state is found, based on Eq. (20), LAS will zip $\mathbf{X}'_K$ further leaking the highest probabilities to $\mathbf{X}_K$ for the solution of Eq. (3) which includes the end state of the system. As no state is ever repeated in the domain, *space complexity* will decrease when the set of states $\mathbf{X}'_K$ is bunked at t'seconds in iterations if Eqs. (19) and (20) are satisfied. In Eq. (13), $P^{(t)}(\mathbf{X}'_K)$ is computed according to Eq. (5) (exponential form of the CME), where τ_m is the tolerance and I is the identity matrix. Due to this stepping bunking of $\mathbf{X}'_K$ from $\mathbf{X}_K$, the time complexity $O(\mathrm{T}^{d+1})$ reduces to $O(min(\mathrm{T}^{\bar{\mathrm{d}}_l+1}, \mathrm{T}|\mathbf{X}_J|))$, where $|\mathbf{X}_J|$ is the size of the state-space [13]. In contrast, the expansion of new nodes carrying similar states tends to increase $O(min(\mathrm{T}^{\bar{\mathrm{d}}_l+1}, \mathrm{T}|\mathbf{X}_J|))$; however, repetitive states are ignored.

If the input τ_m is too small, the default value of $sqrt(eps)$ is automatically used by the algorithm. Where, $sqrt$ is the square root and eps is the default value of the epsilon on machine. The expansion of child nodes containing $state(N_i) = X_i$ stops if the condition of Eq. (32) is not satisfied. If criterion of *slow and fast* reaction is considered, then condition of Eq. (31) or (32) is used depending on number of $R_{M(sr)}$ and $R_{M(fs)}$. The Table 7 shows the steps of the LAS method with embedded $BLNP$ function from step 4a to 5b.

Table 7. Steps of ISP Latitudinal Search (LAS) Algorithm.

Step 0:	**Inputs:** Initial node $\boldsymbol{N_0}$, $\boldsymbol{a}_\mu$, $\boldsymbol{v}_\mu$, tol $\boldsymbol{\tau_m}$, $\boldsymbol{t_f}$, $\boldsymbol{t_{step}}$ **Initialise:** $\boldsymbol{Bound_{lower}} = \mathbf{X}_K$, $\boldsymbol{b'_{N',N}}(\boldsymbol{X} - \boldsymbol{v}_\mu)' = \boldsymbol{P^{(t)}}(\boldsymbol{X_0})$, $\boldsymbol{A} = [\,]$
Step 1:	Start from parent node $N_i = (X_0, \bar{\mathrm{d}}_l) \longleftarrow$ Current State of the system at t_d,
Step 2:	Flag the current node as explored, update A and add the state X_i in the domain so that; *if* $1 - I^T exp(t.A_j).P^{(t)}(X_0) \geq \tau_m(leak)$ holds true go to *Step 3*; *else* stop the algorithm

Step 3:	Sort $exp(t.A_j).P^{(t)}(X_0)$ and shift the set of states in $\mathbf{X}'_K$ at t' having smallest probabilities, if $P^{(t)}(\mathbf{X}_K) \geq \tau_m(leak) > P^{(t)}(\mathbf{X}'_K)$ and at t_d update $\mathbf{X}_K \longleftarrow \mathbf{X}_K - \mathbf{X}'_K$
Step 4a:	Extend the graph dictionary $Dict$ by $v_\mu(X_i(t))$ by 1 level to check all the nodes $\mathbf{n}_J = \left(\mathbf{X}_J, \mathfrak{a}_l, \mathfrak{C}_{N_i,N'_i}(min)\right)$ adjacent to N_i: $Bound_{upper} \leftarrow R_M(Bound_{lower})$ reachable by exactly R_M reactions (from fast to slow) having $\mathfrak{C}_{N_i,N'_i}(min)$. If $\mathbf{n}_K = \left(\mathbf{X}_K, \mathfrak{a}_l, \mathfrak{C}_{N_i,N'_i}(min)\right)$ be the set of adjacent nodes such that $\mathbf{n}_K \in \mathbf{n}_J$ then go to next *Step,*
Step 4b:	Compute the $BLNP$ function for $\mathbf{n}_K \in Bound_{upper}$: $b\left(N_{N1,..NM}\middle\vert b'_{1,N....N',N}\right) = P_{N,1....N,N'}(\omega) * b'_{N',N}(X - v_i)'$
Step 5a:	If $\mathbf{n}_K = \left(\mathbf{X}_K, \mathfrak{a}_l, \mathfrak{C}_{N_i,N'_i}(min)\right) \in domain$, then *update* the values of the set of states $\mathbf{X}_K$ present in $domain$ and take $domain \longleftarrow domain_{previous} \cup domain$ and go back to *Step 1*; *else* If $\mathbf{n}_K = \left(\mathbf{X}_K, \mathfrak{a}_l, \mathfrak{C}_{N_i,N'_i}(min)\right) \notin domain$, then add it to the $queue$ in order, according to reachability and go to the next *Step,*
Step 5b:	*sort* $b\left(N_{N1,..NM}\middle\vert b'_{N1,..NM}\right)$ in descending order and update $queue \longleftarrow (queue\ ; b\left(N_{N1,..NM}\middle\vert b'_{1,N....N',N}\right))$
Step 6:	Pull out the nodes $\mathbf{n}_K = \left(\mathbf{X}_K, \mathfrak{a}_l, \mathfrak{C}_{N_i,N'_i}(min)\right)$ from the $queue$ in order and add the set of states $\mathbf{X}_K$ in the domain as $domain \longleftarrow domain + \mathbf{X}_K$ and take $domain_{previous} \cup domain$, then go back to *Step 1,*
	Output: $domain$ with probable states

LAS will be optimal if the transitions between all the states are uniform, i.e., all the R_M reactions have the same propensity values; however, in real biochemical models this condition is unusual. To see the step-by-step demonstration of the $ISP\ LAS$ algorithm on toy model, refer to SI 2. In following section, we will discuss the second variant of the ISP and apply the method on toy model to see how it differs from LAS.

(c) Longitudinal-Latitudinal Search Strategy

In this section, we delve deeper into the second sub-routine of ISP called the *Intelligent State Projection Longitudinal Latitudinal Search* ($ISP\ LOLAS$). Figure 24 manifest the infrastructure of the $LOLAS$ strategy, showing the G_{mc}, $stack$ and the domain. The $stack$ data structure of $LOLAS$ is based on $LIFO$ (Last In, First Out) method. It assumes that newest state added to the $stack$ is considered first. In particular, we define the bound limit and exploit the direction of the expansion step-by-step based on intuitive knowledge (as discussed in section (a)) gained from the probability of future reaction events and follow the conditions (as discussed in *Results* section). Furthermore, we will show step-by-step how nodes carrying states are explored in bidirectional way and how these states were updated in the domain in I_{tr} iterations.

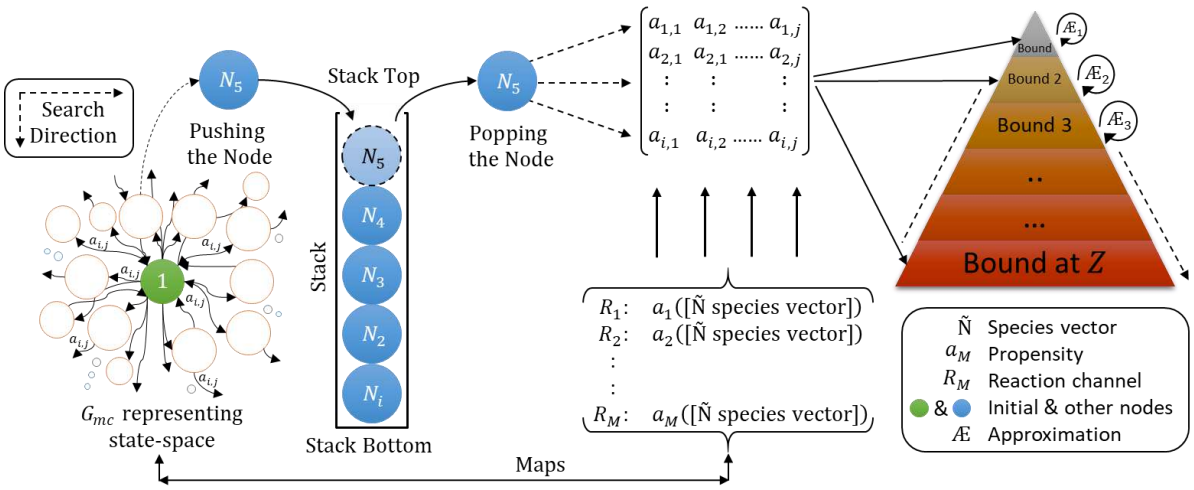

Figure 24. Infrastructure of the *Longitudinal Latitudinal Search* strategy, showing the G_{mc}, the $stack$ and the domain.

The states at level $\mathfrak{d}_l$ are expanded only after the neighbouring states at level $\mathfrak{d}_l - 1$ have been expanded for R_M, i.e., the search is undertaken *level-by-level* and depth $\mathfrak{d}_l$ increases in the same I_{tr} iteration up to a certain $\mathfrak{b}_{limit}$ (bound limit). The limit of the expansion is set by $\mathfrak{b}_{limit}$, $\mathfrak{d}_{step}$ (depth step) but not by the depth $\mathfrak{d}_l$, as in LAS. The $LOLAS$ search updates the dictionary $Dict$ of G_{mc} by the stoichiometric vector function, $v_\mu(X(t))$ on state at level $\mathfrak{d}_l$ to explore the child nodes carrying states on levels $\mathfrak{d}_l + 1$, $\mathfrak{d}_l + 2$ ..... $\mathfrak{d}_l + l$, where $l = \{1,2 \ldots . \infty\}$ and then *retracts* to level $\mathfrak{d}_l$ at which new state exploration decisions can be made. In case of networks with *reversible* reactions, the ISP conditions will prevent the $LOLAS$ to return to the state it came from and prevent transitions containing cycles resulting in DAG with no repetition of any state whatsoever; however, the change in propensities $a_{i,j}$ is validated. Verifying the explored states in $\mathbf{X}_K$ in iterations ensures that the algorithm completes and deadlocks in the state transition cycles cannot occur.

In the absence of $\mathfrak{b}_{limit}$, the algorithm will not retract and will persist to explore longitudinally by tracking only one R_M reaction. In addition, the algorithm will not terminate with an optimal order domain carrying a maximum probability mass, leading to an increase in the approximation error. Instead, it will terminate when carrying only those set of states as a result of tracking for only a few R_M, creating an insufficient domain for approximation. Therefore, by default, the value of $\mathfrak{b}_{limit} \geq 1$ is kept for large systems and can be increased depending upon the dimension of the model and the availability of the random access memory (RAM) of the testing environment. The worst-case *time complexity* of $LOLAS$ depends on the average transitioning factor $\mathfrak{T}$ and depth $\mathfrak{d}_l$ is given by (see SI 7 for a detailed discussion):

$$(\mathfrak{T}+1)\mathfrak{T}^0 + (\mathfrak{T})\mathfrak{T}^1 + (\mathfrak{T}-1)\mathfrak{T}^2 + \cdots \ldots + 3\mathfrak{T}^{\mathfrak{d}_l-2} + 2\mathfrak{T}^{\mathfrak{d}_l-1} + 1\mathfrak{T}^{\mathfrak{d}_l} = O(\mathfrak{T}^{\mathfrak{d}_l}). \quad (57)$$

$LOLAS$ only stores the transition path to the end state besides the neighbours of each relevant node in the exploration and then discards the node from the domain (explored) once all descendants are updated with the relevant propensities in the projection, ready for the approximation. $LOLAS$ first considers the R_1 reaction and the corresponding stoichiometric vector v_1 of the system, to explore all the neighbouring states up to bound limit $\mathfrak{b}_{limit}$, and then considers R_2, R_3,.....,R_M for same $\mathfrak{b}_{limit}$ and the corresponding v_2, v_3,..... , v_M to explore the states. For $count(\mathfrak{b}_{limit})$, $LOLAS$ retracts to R_1 reaction and explores the new

neighbouring states longitudinally and then reconsiders R_2, R_3,….. , R_M to explore the other states in a similar fashion. Provided with this pattern of tracking the reactions, the $BLNP$ function alters this trend and guides this tracking by considering reactions in different order based on their propensities and number of probable states of the system.

If the system is ending in a set of state $\mathbf{X}_K$ carried by $\mathbf{n}_K$ at t_f, then $LOLAS$ will explore the states efficiently as long as $count(\mathfrak{G}_{limit}) \leq \mathfrak{G}_{limit}$, otherwise $count(\mathfrak{G}_{limit})$ is reset for further expansion. Choosing the appropriate $\mathfrak{G}_{limit}$ and $\mathfrak{a}_{step}$ depends on the type of biochemical reaction network and the computing configuration. Starting with depth $1 \longrightarrow \mathfrak{G}_{limit}$, $LOLAS$ explores all the states until they return $null$, and then it resets the $count(\mathfrak{G}_{limit})$ and *retracts* to explore again. In most cases, fewer states are positioned at the lower level and increase at a higher level when the number of active R_M reactions increases, so retracting gives the freedom to track all the reactions simultaneously. Due to the nature of the $LOLAS$ expansion, this finds more states at any time t compared to LAS and also at the deepest level of the graph. The states at depth $\mathfrak{a}_l$ are explored once, the states at depth $\mathfrak{a}_l - 1$ are explored twice, states at depth $\mathfrak{a}_l - 2$ are explored three times and so on, until it has explored all the states of the system. If the input τ_m is too small, the default value of $sqrt(eps)$ is automatically used by the algorithm. Where, $sqrt$ is the square root and eps is the default value of the epsilon on machine. The expansion of child nodes containing $state(N_i) = X_i$ stops if the condition of Eq. (32) is not satisfied. If criterion of *slow and fast* reaction is considered, then condition of Eq. (31) or (32) is used depending on number of $R_{M(sr)}$ and $R_{M(fs)}$. Table 8 shows the steps of the $LOLAS$ method with an embedded $BLNP$ function from step 4a to 5b.

Table 8. Steps of ISP Longitudinal Latitudinal Search ($LOLAS$) Algorithm.

Step 0:	**Inputs:** Initial node $\boldsymbol{N_0}$, $\mathfrak{a}_{\boldsymbol{step}}$, $\mathfrak{G}_{\boldsymbol{limit}}$, $\boldsymbol{a}_{\mu}$, $\boldsymbol{v}_{\mu}$, tol $\boldsymbol{\tau_m}$, $\boldsymbol{t_f}$, $\boldsymbol{t_{step}}$ **Initialise:** $\boldsymbol{Bound_{lower}} = \mathbf{X}_K$, $\boldsymbol{b'_{N',N}}(\boldsymbol{X} - \boldsymbol{v}_{\mu})' = \boldsymbol{P^{(t)}}(\boldsymbol{X_0})$, $\boldsymbol{A} = [\,]$
Step 1:	Initialise $count(\mathfrak{G}_{limit})$ and start from parent node $N_i = (X_0, \mathfrak{a}_l) \longleftarrow$ Current State of the system at t_d,
Step 2:	Flag the current node as explored, update A and add the state X_i in the domain so that; *if* $1 - I^T exp(t.A_j).P^{(t)}(X_0) \geq \tau_m(leak)$ holds true go to *Step 3*; *else* stop the algorithm.
Step 3:	Sort $exp(t.A_j).P^{(t)}(X_0)$ and shift the set of states in $\mathbf{X}'_K$ at t' having smallest probabilities, if $P^{(t)}(\mathbf{X}_K) \geq \tau_m(leak) > P^{(t)}(\mathbf{X}'_K)$ and at t_d update $\mathbf{X}_K \longleftarrow \mathbf{X}_K - \mathbf{X}'_K$
Step 4a:	For $\mathfrak{a}_{\boldsymbol{step}}$, extend the graph dictionary $Dict$ by $v_{\mu}(X_i(t))$ for $count(\mathfrak{G}_{limit})$ to check all the nodes $\mathbf{n}_J = \left(\mathbf{X}_J, \mathfrak{a}_l, \mathfrak{C}_{N_i,N'_i}(min)\right)$ adjacent to N_i: $Bound_{upper} \leftarrow R_M(Bound_{lower})$ reachable by exactly R_M reactions (from fast to slow) having $\mathfrak{C}_{N_i,N'_i}(min)$. If $\mathbf{n}_K = \left(\mathbf{X}_K, \mathfrak{a}_l, \mathfrak{C}_{N_i,N'_i}(min)\right)$ be the set of adjacent nodes such that $\mathbf{n}_K \in \mathbf{n}_J$, then go to the next *Step,*
Step 4b:	Compute the $BLNP$ function for $\mathbf{n}_K \in Bound_{upper}$: $b(N_{N1,..NM}\|b'_{1,N....N',N}) = P_{N,1....N,N'}(\omega) * b'_{N',N}(X - v_i)'$
Step 5a:	If $\mathbf{n}_K = \left(\mathbf{X}_K, \mathfrak{a}_l, \mathfrak{C}_{N_i,N'_i}(min)\right) \in domain$, then *update* the values of the set of states $\mathbf{X}_K$ present in $domain$ and take $domain \longleftarrow domain_{previous} \cup domain$ and go back to *Step 1*; *else* If $\mathbf{n}_K = \left(\mathbf{X}_K, \mathfrak{a}_l, \mathfrak{C}_{N_i,N'_i}(min)\right) \notin domain$, then add it to the $stack$ in order, according to reachability and go to next *Step,*
Step 5b:	*sort* $b(N_{N1,..NM}\|b'_{N1,..NM})$ in descending order and update $stack \longleftarrow (stack\ ;$ $b(N_{N1,..NM}\|b'_{1,N....N',N}))$

Step 6:	Pop of the top nodes $\mathbf{n}_K = \left(\mathbf{X}_K, \mathfrak{d}_l, \mathfrak{C}_{N_i,N_i'}(min)\right)$ from the $stack$ and add the set of states $\mathbf{X}_K$ in the domain as $domain \longleftarrow domain + \mathbf{X}_K$ and take $domain_{previous} \cup domain$, and go to next *Step,*
Step 7:	If $count(\mathfrak{b}_{limit}) = \mathfrak{b}_{limit}$ creates $Bound_{upper} = \{domain\}\ up\ to\ \mathfrak{b}_{limit}$ then label $Bound_{lower} \leftarrow Bound_{upper}$ and go back to *Step* 1; else *if* $count(\mathfrak{b}_{limit}) < \mathfrak{b}_{limit}$ creates $\{domain\}\ up\ to\ count(\mathfrak{b}_{limit})$ then go to next *Step*,
Step 8:	$count(\mathfrak{b}_{limit}) \leftarrow count(\mathfrak{b}_{limit}) + 1$ and go to *Step 4a*
	Output: $domain$ with probable states

Refer SI 3 for the step-by-step demonstration of the $ISP\ LOLAS$ algorithm, where we assume the same toy model system as discussed in SI 3.

List of abbreviations and notations

$\boldsymbol{a_{i,j}}$ or $\boldsymbol{a}_{\mu}$	Propensity of chemical reaction
$\Delta\boldsymbol{a_{i,j}}$	Change in propensity
$\boldsymbol{a'_{1,N'},\ldots,a'_{N,N'}}$	Propensities of the prior reactions
$\boldsymbol{a_{fr}}$	Probability of a jump process from state X_{i-1} to X_i per unit time
$\boldsymbol{a_{rv}}$	Probability of a jump process from state X_i to X_{i-1} per unit time
$\boldsymbol{A}$ or $\boldsymbol{A_{i,j}}$	Defines the transition between *i, j* and its weightage
$\mathfrak{b}_{\boldsymbol{limit}}$	Exploration bound limit in $LOLAS$
$\boldsymbol{b'_{N',N}(X - v_{\mu})'}$	Prior Bayesian likelihood values $\{b'_{1,N}, \ldots \ldots b'_{N',N}\}$
$\boldsymbol{b(N_{N1,\ldots NM}|b'_{1,N\ldots.N',N})}$	Represents Bayesian likelihood value given prior $b'_{N',N}$
$\boldsymbol{Bound_{lower}}$ or $\boldsymbol{Bound_L}$	Define the set of states $\{X_{1,2\ldots\ldots S}, b_{1,2,3,\ldots.limit}\}$ at $\mathfrak{b}_{limit}$ already present in the domain for current iteration
$\boldsymbol{Bound_{upper}}$ or $\boldsymbol{Bound_U}$	Define the set of states $\{X_{1,2\ldots\ldots S}, b_{1,2,3,\ldots.limit}\}$ at $\mathfrak{b}_{limit}$ added in the domain at the end of current iteration
$\boldsymbol{c, c_1, c_2}$	Constants
$\mathfrak{C}_{\boldsymbol{N_i,N_i'}}$	Total transition/walk cost from node N_i to N_i'
$\boldsymbol{Dict}$	Dictionary of the model having transition records
$\mathfrak{d}_{\boldsymbol{l}}$	Exploration depth limit in LAS
$\mathfrak{d}_{\boldsymbol{step}}$	Exploration depth step in ISP
$\boldsymbol{dim}$	Dimension of sub-matrix in Sliding Windows Method
$\boldsymbol{domain}$	Defines the set of states of domain in current iteration that forms $Bound_{upper}$
$\boldsymbol{domain_{previous}}$	Defines the set of states of domain in previous iteration that forms $Bound_{lower}$
$\boldsymbol{D_j}$	Diagonal matrix whose diagonal entries are one
$\boldsymbol{e}$	Markov chain tree edge representing walk from N_i to N_i'
$\boldsymbol{e_1}$	First unit basis vector in Krylov Subspace Method
$\boldsymbol{e_{rror}}$	Represents error value in calculation

$\boldsymbol{exp}()$	Exponential function
$\boldsymbol{eps}$	Epsilon
$'\mathrm{E}_{\boldsymbol{y}}$	Denote the sequence of events $'\mathrm{E}_1$, $'\mathrm{E}_2$….,
$\boldsymbol{f(y)}$	Represent the positive real value function of y
$\boldsymbol{G_{mc}}$	Represents graph associated with Markov chain tree
$\bar{\boldsymbol{H}}_{\boldsymbol{dim}}$	Upper matrix (Hessenberg Matrix)
$\boldsymbol{I^T}$	Identity matrix $I = diag(1,1, \ldots..1)^T$
$\boldsymbol{I_{tr}}$	Denote the iterations in ISP
$\boldsymbol{k_M}$	Kinetic parameter of the chemical reaction where $M = \{1,2 \ldots. \infty\}$
$\boldsymbol{l}$	Used as subscript for Length of depth, for example d_l
$\mathbf{n_J}$	Set of node as $\{N_1, N_2, \ldots \ldots. N_{S^{\tilde{\mathrm{N}}}}\}$
$\mathbf{n_K}$	Set of nodes carrying set of $\mathbf{X}_K$ at any iteration
$\mathbf{n}'_{\boldsymbol{K}}$	Set of nodes carrying set of $\mathbf{X}'_K$ at any iteration
$\boldsymbol{N_0}$	Root node carrying initial state X_0
$\boldsymbol{N_i}$ or $\boldsymbol{N_i}$	Any node
$\tilde{\mathrm{N}}$ or $\{\boldsymbol{S_1}, \ldots \ldots. \boldsymbol{S}_{\tilde{\mathrm{N}}}\}$	Number of different species
$\boldsymbol{num_1}, \boldsymbol{num_2}$	Random number generated by uniform random number generator (URN)
$\boldsymbol{P^{(t_0)}(X_0)}$	Initial probability at $t = 0$
$\boldsymbol{P^{(t)}(\mathbf{X}_K)}$	Probability of set of states at time t
$\boldsymbol{P_{N,N'}(\omega)}$	Weighted probability of transition from N_i to N'_i
$\boldsymbol{R_M}$	M elementary chemical reaction channels $\{R_1, R_2 \ldots \ldots. R_M\}$
$\boldsymbol{R'_M}$	Prior M elementary chemical reaction channels $\{R'_1, R'_2 \ldots \ldots. R'_M\}$
$\boldsymbol{R_{M(fs)}}$	M elementary chemical reaction channels of fast reactions
$\boldsymbol{R_{M(sr)}}$	M elementary chemical reaction channels of fast reactions
$\boldsymbol{\mathrm{R}_{tract}}$	Number of retractions in $LOLAS$
$\boldsymbol{S}^{\tilde{\mathrm{N}}}$	Approximate number of states
$\hat{\mathrm{S}}$	Number of stages in expansion $\{1,2, \ldots \ldots.\}$
SI	Supporting information
$\boldsymbol{S_{uc}}$	Implicit successor or operator
$\boldsymbol{sqrt}$	Square root
$\boldsymbol{t_0}$	Time at which initial conditions of system are defined
$\boldsymbol{t'}$	Time at which $\mathbf{X}'_K$ is dropped from the domain
$\boldsymbol{t}$	Any random time in seconds
$\boldsymbol{t_d}$	Time at which $\mathbf{X}_K$ is updated in the iteration
$\boldsymbol{t_f}$	Final time at which solution is required
T	Transitioning factor

$\boldsymbol{U_{X_i,X_{i'}}}$	Set of all arborescences
$\boldsymbol{\lvert U \rvert}$	Define the cardinality of any set
$\boldsymbol{v}$	Krylov Sub-space method - A column vector of a length equal to the number of different species present in the system
$\boldsymbol{v_\mu}$ or $\boldsymbol{v_M}$	Stoichiometric vector represents the change in the molecular population of the chemical species by the occurrence of one R_M reaction. It also defines the transition from state X_i to $X_{i'}$ in Markov chain tree
$\boldsymbol{v_\mu(X(t))}$ or $\boldsymbol{v_M(X(t))}$	Stoichiometric vector function, where X is any random state
$\boldsymbol{V_\mu}$ or $\boldsymbol{V_M}$	Matrix of all the Stoichiometric vectors $[v_1; v_2; \ldots\ldots v_\mu]$
$\boldsymbol{W^y}$	Probability that is computed inductively by $W^{(0)} = P^{(0)}$ in uniformisation method
$\boldsymbol{x_1, \ldots\ldots x_{\tilde{N}}}$	Number of counts of different species
$\boldsymbol{X}$ or $\boldsymbol{X_i}$ or $\boldsymbol{X_{i'}}$	Any random state
$\boldsymbol{X_0}$	Initial state or Initial condition
$\mathbf{X_J}$	ordered set of possible states $\{X_1, \ldots\ldots X_{S^{\tilde{N}}}\}$ of the system
$\mathbf{X_K}$	Set of new states or domain at any iteration
$\mathbf{X'_K}$	Set of states dropped from domain at t' at any iteration
$\boldsymbol{y}, \boldsymbol{y_0}$	Positive integers
$\boldsymbol{Y_y}$	Poisson process given that $0 < y \leq M$
$\boldsymbol{Z}$	Number of bounds in ISP
$\boldsymbol{\tau_m}$ or $\boldsymbol{tol_m}$	Tolerance value
$\boldsymbol{\tau_m(leak)}$	$P^{(t)}(\mathbf{X'_K})$ leakage point
Æ	Approximate solution of the CME
$\boldsymbol{\omega}$	Weight or cost of single transition from X_i to $X_{i'}$. It is equivalent to $a_{i,j}$
$\boldsymbol{\mathbb{K}_c}$	Markov chain representing biochemical process
$\mathbb{HK}$	Markov chain tree with $\mathbf{n}_J$
$\boldsymbol{\lambda t}$	Uniformisation rate
$\boldsymbol{\aleph_j}$	Number of nonzero elements in P_j
$\boldsymbol{\varphi}$	Sample space
$\boldsymbol{\Omega}$	Asymptotic lower bound
$\boldsymbol{O}$	Asymptotic upper bound
$\boldsymbol{\Theta}$	Asymptotic tight bound
$\{\mathbf{1}, \mathbf{2} \ldots\ldots \boldsymbol{K}\}$	Indexing of set of states and set of nodes
$\mu = \{\mathbf{1}, \mathbf{2}, \ldots. \boldsymbol{M}\}$	Channels of chemical reaction propensity

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Availability of data and materials

All the results data files generated and analysed during the current study are available in the "ispy/others" repository of https://github.com/rkosarwal; however, codes are not publicly available due to privacy from search engines but are available from the corresponding or first author on reasonable requests.

Competing interests

Not applicable

Funding

RK acknowledges the writing scholarship received by Lincoln University, New Zealand.

Author's contributions

RK and DK developed the research questions and designed the research. RK developed and implemented the algorithm; DK and SS directed the project, RK and DK wrote the manuscript and SS independently critiqued the manuscript, which has been read, improved and approved by all authors.

Acknowledgements

RK acknowledges the project funding and necessary resourcing received from Lincoln University, New Zealand duration of three years.

Author Details

[1,2,3]Centre for Advanced Computational Solutions (C-fACS), Lincoln University, New Zealand, [1,2,3]Complex Systems, Big Data and Informatics Initiative (CSBII), Lincoln University, New Zealand.

References

1. Roberts RM, Cleland TJ, Gray PC, Ambrosiano JJ. Hidden Markov Model for Competitive Binding and Chain Elongation. J Phys Chem B [Internet]. 2004 May;108(20):6228–32. Available from: https://pubs.acs.org/doi/10.1021/jp036941q

2. Kholodenko BN. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. Eur J Biochem [Internet]. 2000 Mar;267(6):1583–8. Available from: http://doi.wiley.com/10.1046/j.1432-1327.2000.01197.x
3. Ozer M, Uzuntarla M, Perc M, Graham LJ. Spike latency and jitter of neuronal membrane patches with stochastic Hodgkin–Huxley channels. J Theor Biol [Internet]. 2009 Nov;261(1):83–92. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0022519309003191
4. Murray JM, Fanning GC, Macpherson JL, Evans LA, Pond SM, Symonds GP. Mathematical modelling of the impact of haematopoietic stem cell-delivered gene therapy for HIV. J Gene Med [Internet]. 2009 Dec;11(12):1077–86. Available from: http://doi.wiley.com/10.1002/jgm.1401
5. Hogervorst E, Bandelow S, Combrinck M, Irani SR, Smith AD. The Validity and Reliability of 6 Sets of Clinical Criteria to Classify Alzheimer's Disease and Vascular Dementia in Cases Confirmed Post-Mortem: Added Value of a Decision Tree Approach. Dement Geriatr Cogn Disord [Internet]. 2003;16(3):170–80. Available from: https://www.karger.com/Article/FullText/71006
6. Schulze J, Sonnenborn U. Yeasts in the Gut. Dtsch Aerzteblatt Online [Internet]. 2009 Dec 21; Available from: https://www.aerzteblatt.de/10.3238/arztebl.2009.0837
7. Zhou Y, Hou Y, Shen J, Huang Y, Martin W, Cheng F. Network-based drug repurposing for novel coronavirus 2019-nCoV/SARS-CoV-2. Cell Discov [Internet]. 2020;6(1). Available from: http://dx.doi.org/10.1038/s41421-020-0153-3
8. Gillespie DT. A rigorous derivation of the chemical master equation. Phys A Stat Mech its Appl. 1992;188(1–3):404–25.
9. Gillespie DT. Exact Stochastic Simulation of Coupled Chemical Reactions. J Phys Chem [Internet]. 1977;81(1):2340–61. Available from: https://www.ncbi.nlm.nih.gov/pubmed/17411109
10. Weber R. Markov Chains. StatslabCamAcUk [Internet]. 2012;28–49. Available from: http://www.statslab.cam.ac.uk/~james/Markov/%5Cnpapers2://publication/uuid/9F19F535-7D18-4854-B40A-5C003D07F126
11. Goutsias J, Jenkinson G. Markovian dynamics on complex reaction networks. Phys Rep [Internet]. 2013;21218(2):199–264. Available from: http://arxiv.org/abs/1205.5524%0Ahttp://dx.doi.org/10.1016/j.physrep.2013.03.004
12. Gillespie DT. Markov Processes - An Introduction for Physical Scientists [Internet]. Elsevier; 1992. 592 p. Available from: https://linkinghub.elsevier.com/retrieve/pii/C2009022215X
13. Burrage K, Hegland M, Macnamara S, Sidje R. A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems. Proc Markov Anniv Meet. 2006;1–18.
14. Jones MT. Estimating Markov Transition Matrices Using Proportions Data: An Application to Credit Risk. IMF Work Pap [Internet]. 2005;05(219):1. Available from: http://elibrary.imf.org/view/IMF001/02089-9781451862386/02089-9781451862386/02089-9781451862386.xml
15. Mouroutsos SG, Sparis PD. Taylor series approach to system identification, analysis and optimal control. J Franklin Inst [Internet]. 1985 Mar 1 [cited 2018 Jul 12];319(3):359–71. Available from: https://www.sciencedirect.com/science/article/pii/0016003285900560
16. Eslahchi MR, Dehghan M. Application of Taylor series in obtaining the orthogonal

operational matrix. Comput Math with Appl. 2011;61(9):2596–604.
17. Wolf V, Goel R, Mateescu M, Henzinger T. Solving the chemical master equation using sliding windows. BMC Syst Biol [Internet]. 2010;4(1):42. Available from: http://www.biomedcentral.com/1752-0509/4/42
18. Sidje RB, Vo HD. Solving the chemical master equation by a fast adaptive finite state projection based on the stochastic simulation algorithm. Math Biosci. 2015;269:10–6.
19. Sunkara V, Hegland M. An optimal finite state projection method. Procedia Comput Sci [Internet]. 2010;1(1):1579–86. Available from: http://dx.doi.org/10.1016/j.procs.2010.04.177
20. Munsky B, Khammash M. The finite state projection algorithm for the solution of the chemical master equation. J Chem Phys. 2006;124(4):1–13.
21. Mikeev L, Sandmann W, Wolf V. Numerical approximation of rare event probabilities in biochemically reacting systems. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2013;8130 LNBI:5–18.
22. MacNamara S, Bersani AM, Burrage K, Sidje RB. Stochastic chemical kinetics and the total quasi-steady-state assumption: Application to the stochastic simulation algorithm and chemical master equation. J Chem Phys. 2008;129(9).
23. Dinh KN, Sidje RB. An application of the Krylov-FSP-SSA method to parameter fitting with maximum likelihood. Phys Biol [Internet]. 2017;14(6):065001. Available from: http://stacks.iop.org/1478-3975/14/i=6/a=065001?key=crossref.1aa76b1779bc3b108b54e7ed0f2d8785
24. Harrison RL, Granja C, Leroy C. Introduction to Monte Carlo Simulation. In 2010. p. 17–21. Available from: http://aip.scitation.org/doi/abs/10.1063/1.3295638
25. Dinh KN, Sidje RB. Understanding the finite state projection and related methods for solving the chemical master equation. Phys Biol. 2016;13(3).
26. Munsky B, Khammash M. A multiple time interval finite state projection algorithm for the solution to the chemical master equation. J Comput Phys. 2007;226(1):818–35.
27. Sunkara V. Analysis and Numerics of the Chemical Master Equation. 2013;1–134. Available from: http://www.math.kit.edu/ianm3/~sunkara/media/thesis_sunkara.pdf
28. Padgett JMA, Ilie S. An adaptive tau-leaping method for stochastic simulations of reaction-diffusion systems. AIP Adv. 2016;6(3).
29. Cao Y, Gillespie DT, Petzold LR. Efficient step size selection for the tau-leaping simulation method. J Chem Phys. 2006;124(4):1–11.
30. Schlecht V. How to predict preferences for new items. Invest Manag Financ Innov. 2014;5(4):7–24.
31. Fahidy TZ. Some Applications of Bayes' Rule in Probability Theory to Electrocatalytic Reaction Engineering. Int J Electrochem [Internet]. 2011;2011(1):1–5. Available from: http://www.hindawi.com/journals/ijelc/2011/404605/
32. Anantharam V, Tsoucas P. A proof of the Markov chain tree theorem. Stat Probab Lett [Internet]. 1989 Jun 1 [cited 2018 May 15];8(2):189–92. Available from: https://www.sciencedirect.com/science/article/pii/0167715289900163
33. Aldous D. The Continuum random tree II: an overview. In: Barlow MT, Bingham NH, editors. Stochastic Analysis [Internet]. Cambridge: Cambridge University Press; 1992. p. 23–70. Available from: https://www.cambridge.org/core/product/identifier/CBO9780511662980A009/type/book_part
34. Diaconis P, Efron B. Markov Chains Indexed by Trees. Ann Stat. 1985;13(3):845–74.

35. Gursoy BB, Kirkland S, Mason O, Sergeev S. On the markov chain tree theorem in the max algebra. Electron J Linear Algebr. 2012;26(12):15–27.
36. Mastny EA, Haseltine EL, Rawlings JB. Two classes of quasi-steady-state model reductions for stochastic kinetics. J Chem Phys [Internet]. 2007 Sep 7;127(9):094106. Available from: http://aip.scitation.org/doi/10.1063/1.2764480
37. Ling H. Investigation of Robustness and Dynamic Behaviour of G1/S Checkpoint/DNA-damage Signal Transduction Pathway based on Mathematical Modelling and a Novel Neural Network Approach. Thesis. PhD Thesis. Lincoln University; 2011.
38. Chijindu EVC. Search in Artificial Intelligence Problem Solving. 2012;5(5):37–42.
39. Barr A, Feigenbaum E. The handbook of artificial intelligence vol. I. Math Soc Sci. 1983;4:320–4.
40. Korf RE. Artificial Intelligence Search Algorithms. Algorithms Theory Comput Handb. 1996;
41. Korf RE. Depth-first iterative-deepening. An optimal admissible tree search. Artif Intell. 1985;27(1):97–109.
42. Rudowsky I. Intelligent agents. Commun Assoc Inf Syst. 2004;14(August):275–90.
43. Lawlor OS. In-memory Data Compression for Sparse Matrices. Proc 3rd Work Irregul Appl Archit Algorithms [Internet]. 2013;(December):6:1--6:6. Available from: http://doi.acm.org/10.1145/2535753.2535758
44. Koza Z, Matyka M, Szkoda S, Mirosław Ł. Compressed Multi-Row Storage Format for Sparse Matrices on Graphics Processing Units. 2012;1–26. Available from: http://arxiv.org/abs/1203.2946%0Ahttp://dx.doi.org/10.1137/120900216
45. Manoukian EB. Modern Concepts and Theorems of Mathematical Statistics [Internet]. New York, NY: Springer New York; 1986. (Springer Series in Statistics). Available from: http://link.springer.com/10.1007/978-1-4612-4856-9

Figures

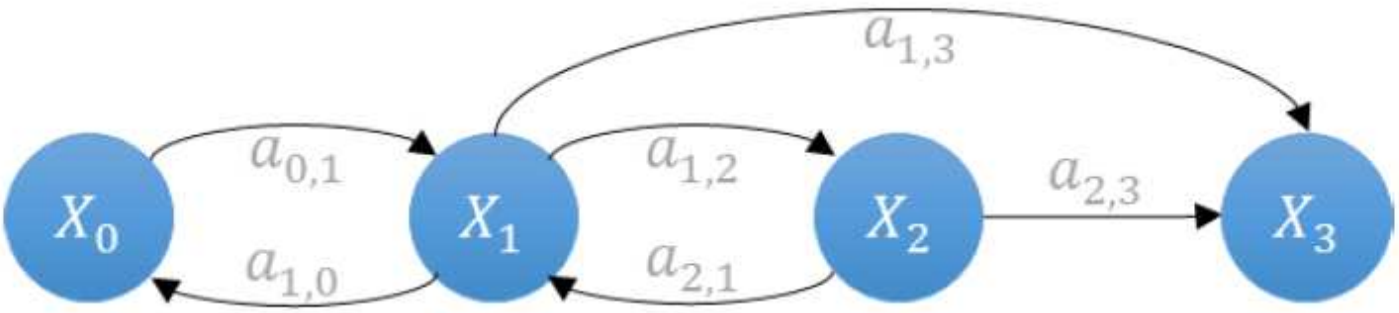

Figure 1

Markov chain graph showing forward and reversible reactions through four different states.

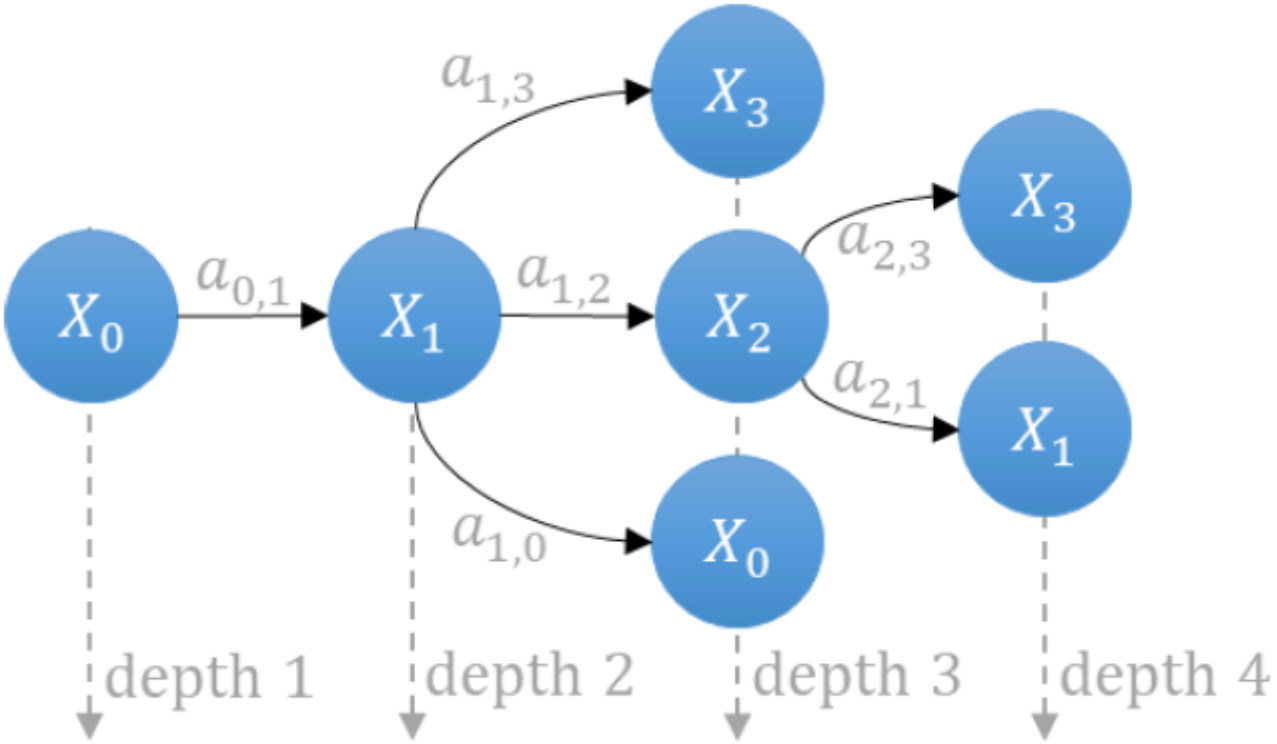

Figure 2

Equivalent tree of Markov chain graph, as shown in Figure 1. It is a special form of graph that has no cycle, no self-loops and depicts the state-space of the system in the form of a tree (□□□).

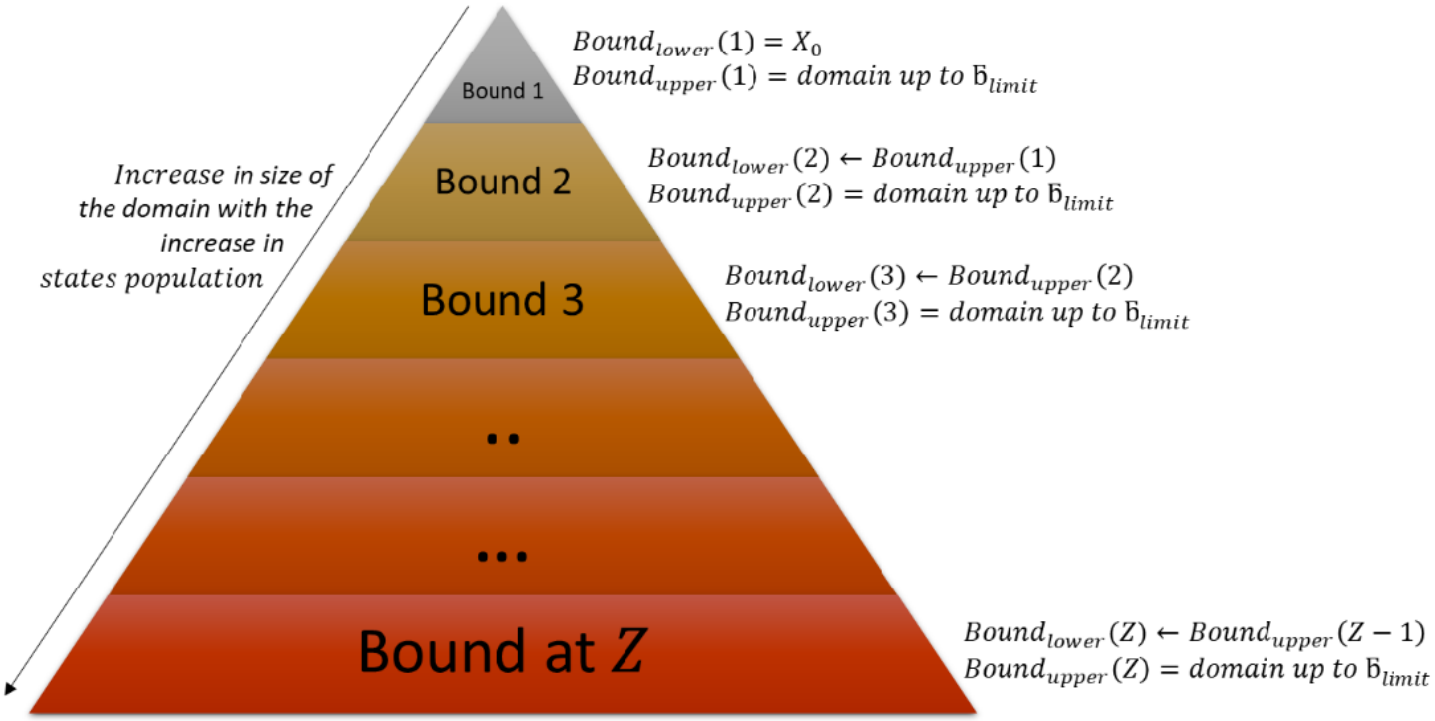

Figure 3

General framework of 2D pyramid domain showing the increase in the size of the domain with the increase in state with an increase in the bounds. □□□□□□□□□□(□) represents the initial condition, whereas □□□□□□□□□□(□) represents the final domain carrying explored set of states of the system.

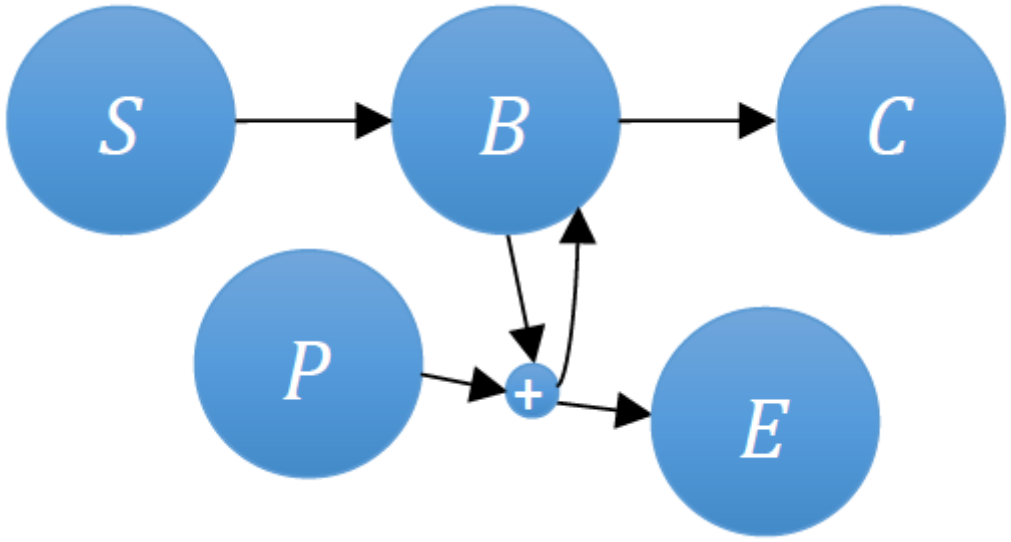

Figure 4

Catalytic reaction network having five Ñ = □ species □, □, □, □, and □ in a network defining reactions, as given in Eq. (33).

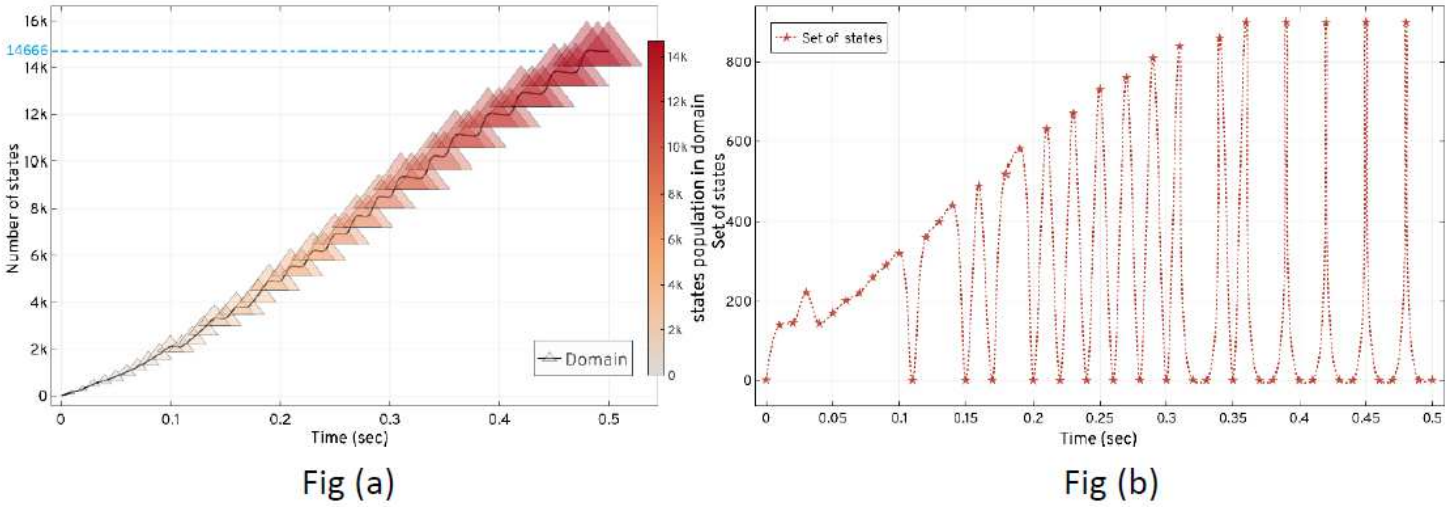

Figure 5

Expansion and updation of the states and set of states explored for the catalytic reaction system based on the □□□ method. Fig (a) depicts that state-space expansion increases the number of additions of new states in the domain. The size and colour of □ shows the increase in the size of the domain with the states population. In Fig (b), □□□ unfolds the state-space pattern to update the states in the domain and expands 14666 probable states in 0.5 sec.

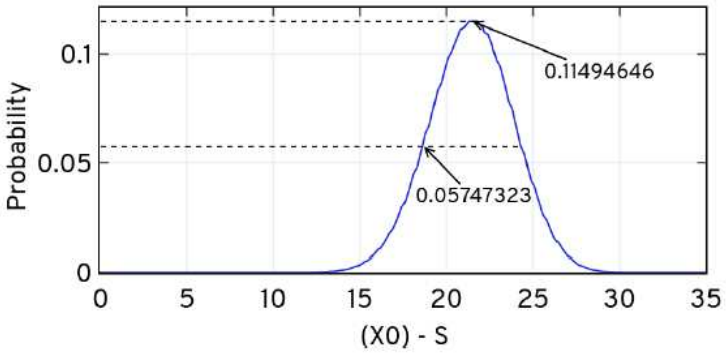

Fig (A). Probability of S over t_f

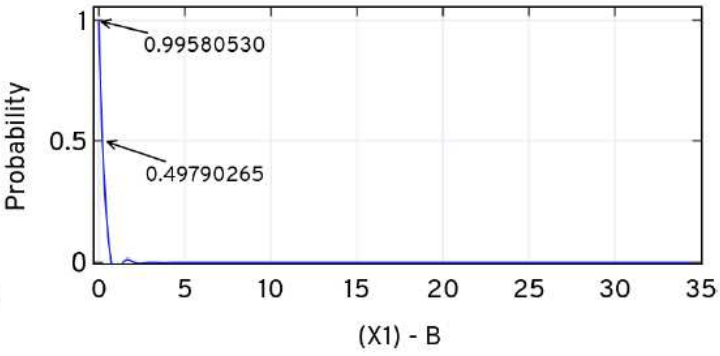

Fig (B). Probability of B over t_f

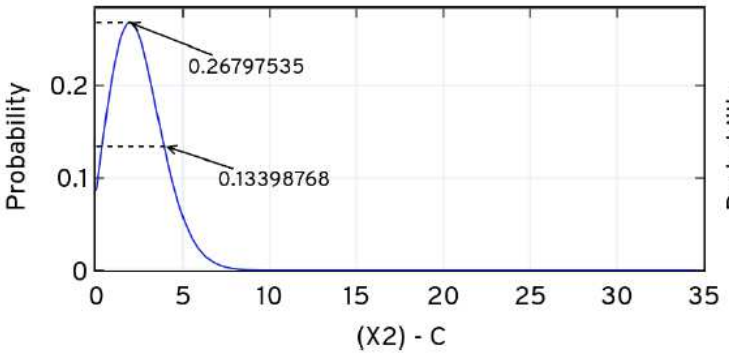

Fig (C). Probability of C over t_f

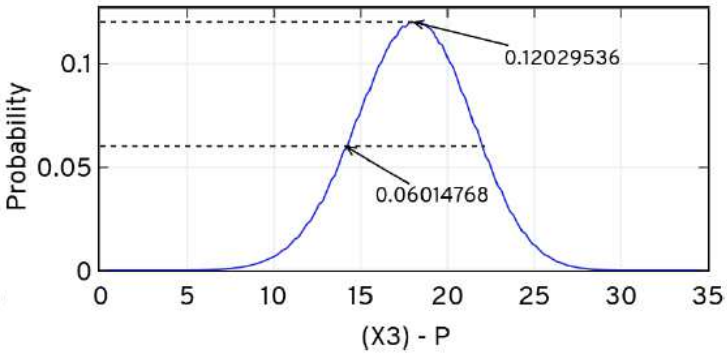

Fig (D). Probability of P over t_f

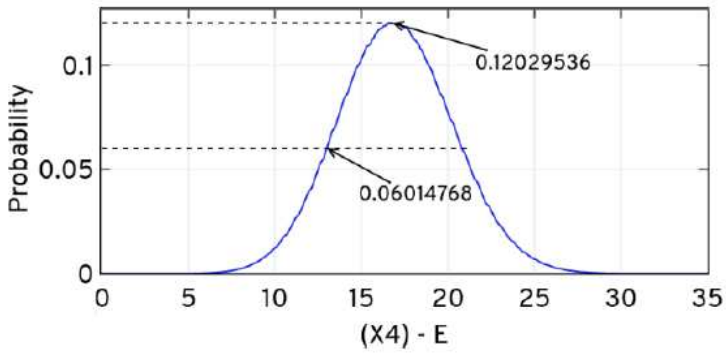

Fig (E). Probability of E over t_f

Figure 6

Conditional probability of the catalytic system evaluated at □□ = □. □ sec, □□□□□ =□. □□ using □□□. Fig (A) is the probability of the species □ over □□, Fig (B) is the probability of the species □ over □□, Fig (C) is the probability of the species □ over □□, Fig (D) is the probability of the species □ over □□, Fig (E) is the probability of the species □ over □□.

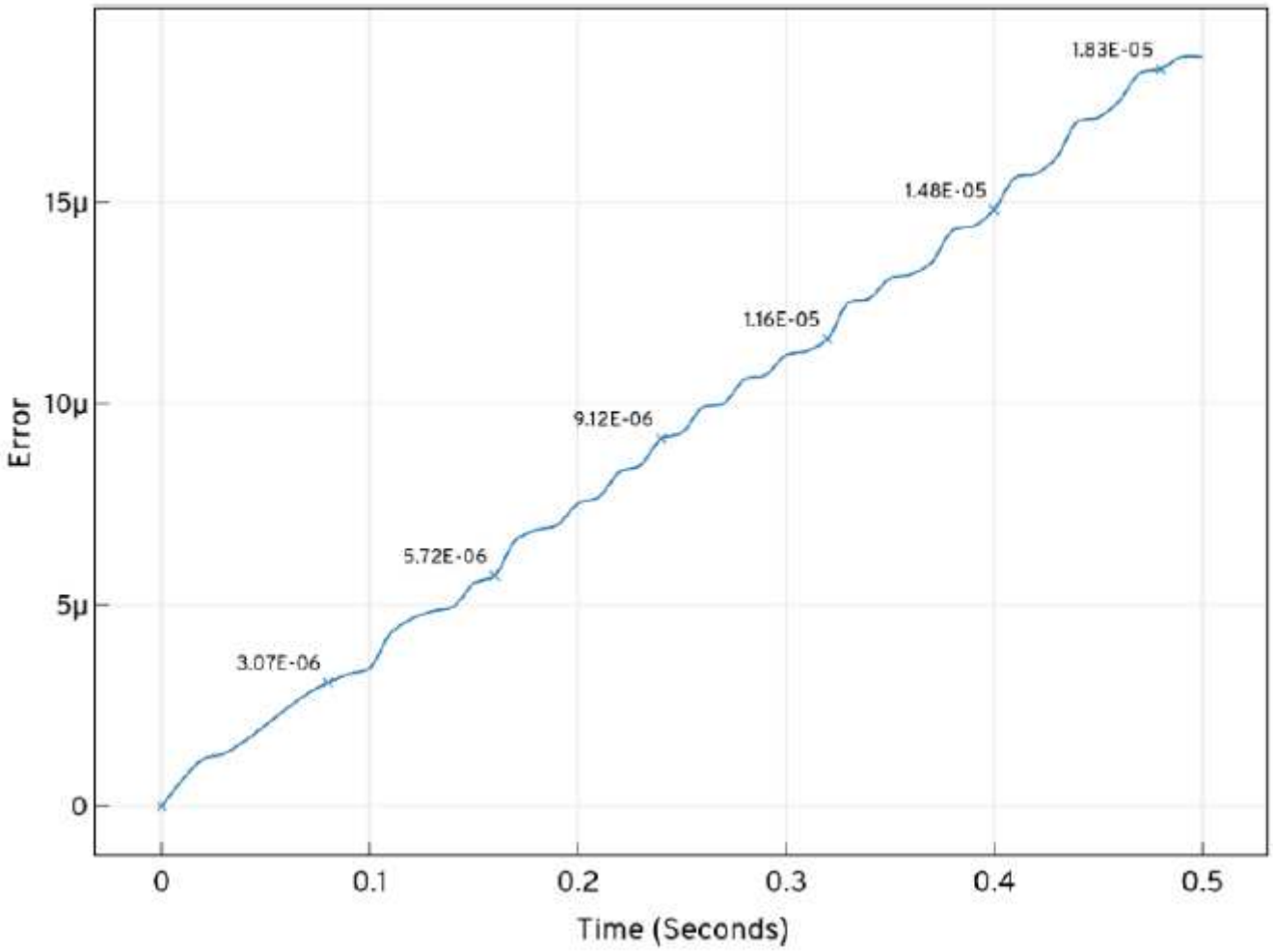

Figure 7

Total probability of states bunked at ' from the domain of catalytic system produced by iteration while expansion and solving the CME.

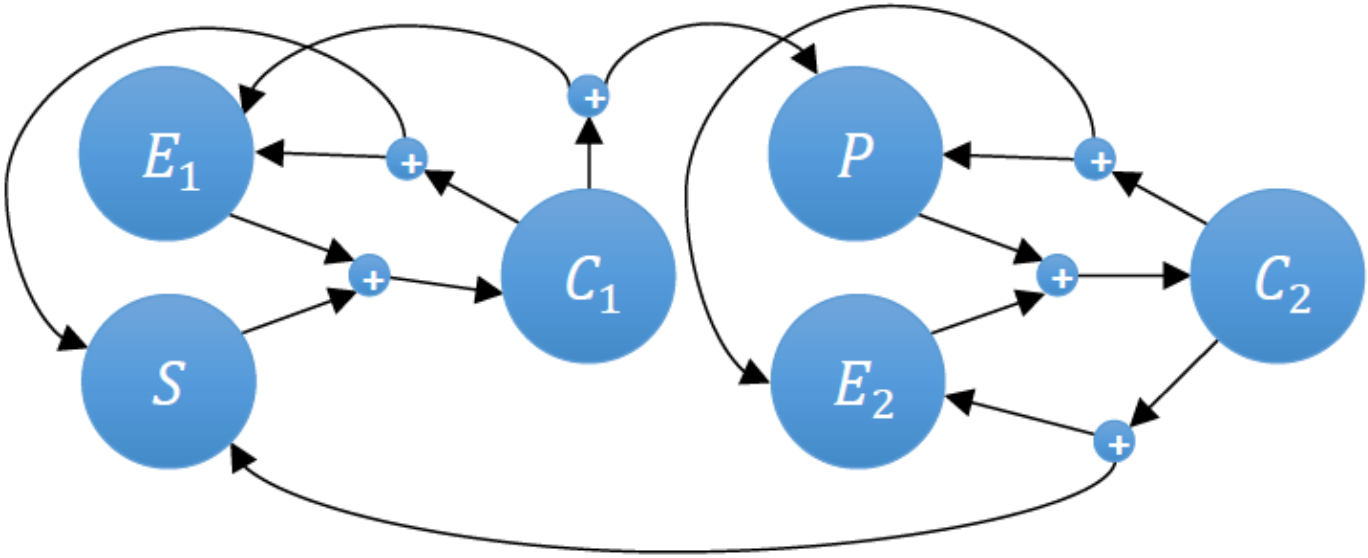

Figure 8

Coupled enzymatic reactions network. Showing six Ñ = □ species, □, □□, □□, □, □□, □□, in a network defining reactions, as given in Eqs. (39) and (40).

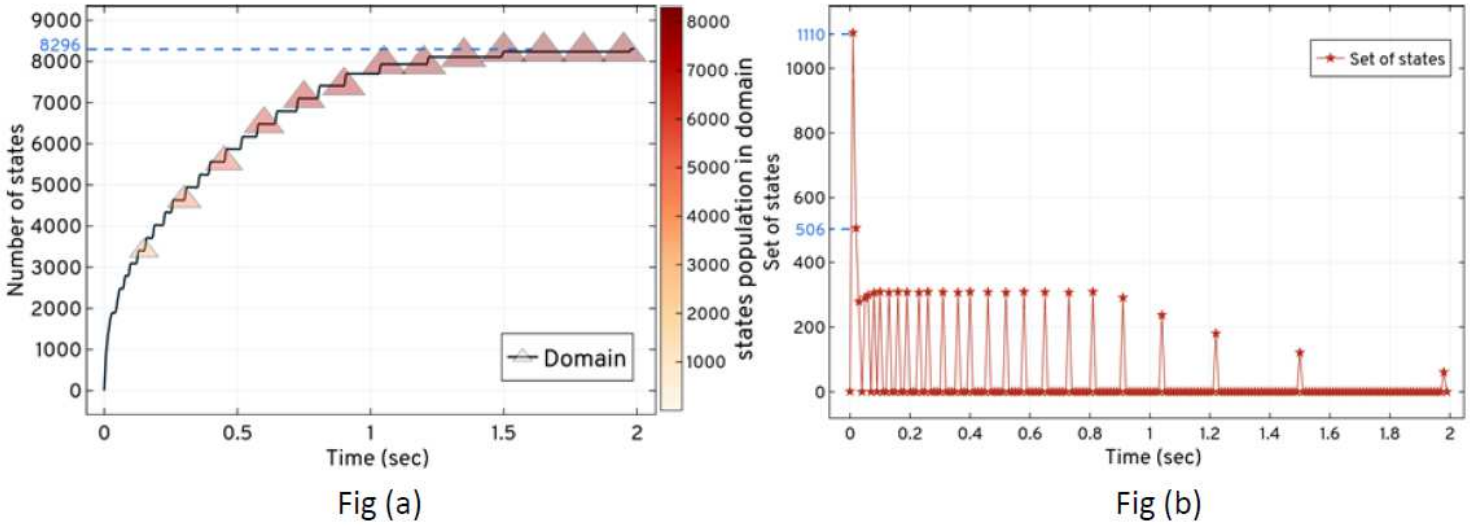

Figure 9

Expansion and updating of the states and set of states explored for the dual enzymatic reaction network based on the □□□ □□□□□ method. Fig (a) depicts state-space expansion increases the number of additions of new states in the domain. The size and colour of □ shows the increase in size of the domain with the states' population. In Fig (b), □□□ □□□□□ unfolds the state-space pattern to update states in the domain and expands 8296 probable states in 2.0 sec. □ shows the time point where new set of states is explored and updated in the domain.

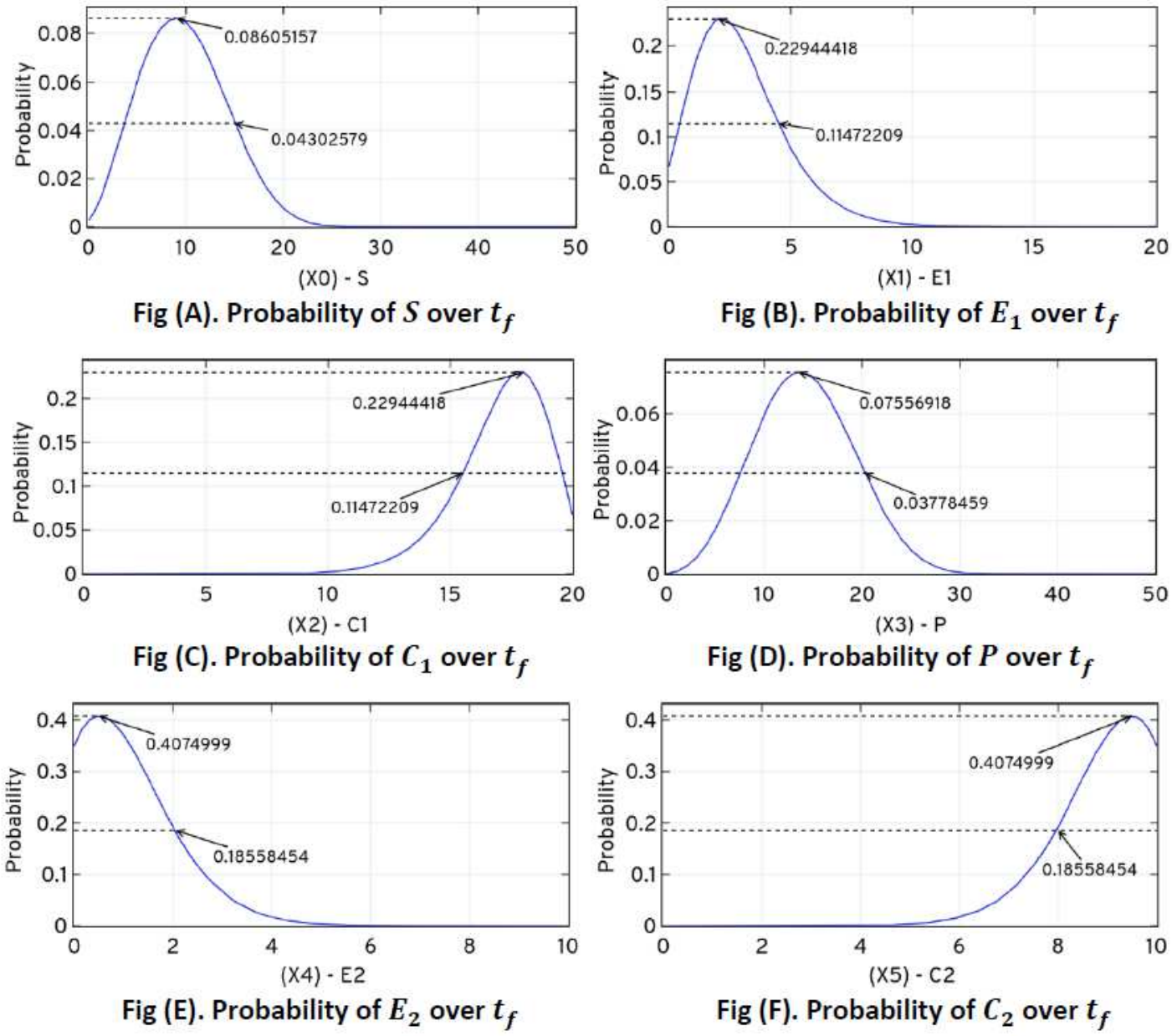

Figure 10

Conditional probability of the dual enzymatic reactions system evaluated at □□ = □. □ sec, □□□□□ = □. □□ using □□□□□. Fig (A) is the probability of the species □ over □□, Fig (B) is the probability of the species □ over □□, Fig (C) is the probability of the species □ over □□, Fig (D) is the probability of the species □ over □□, Fig (E) is the probability of the species □ over □□.

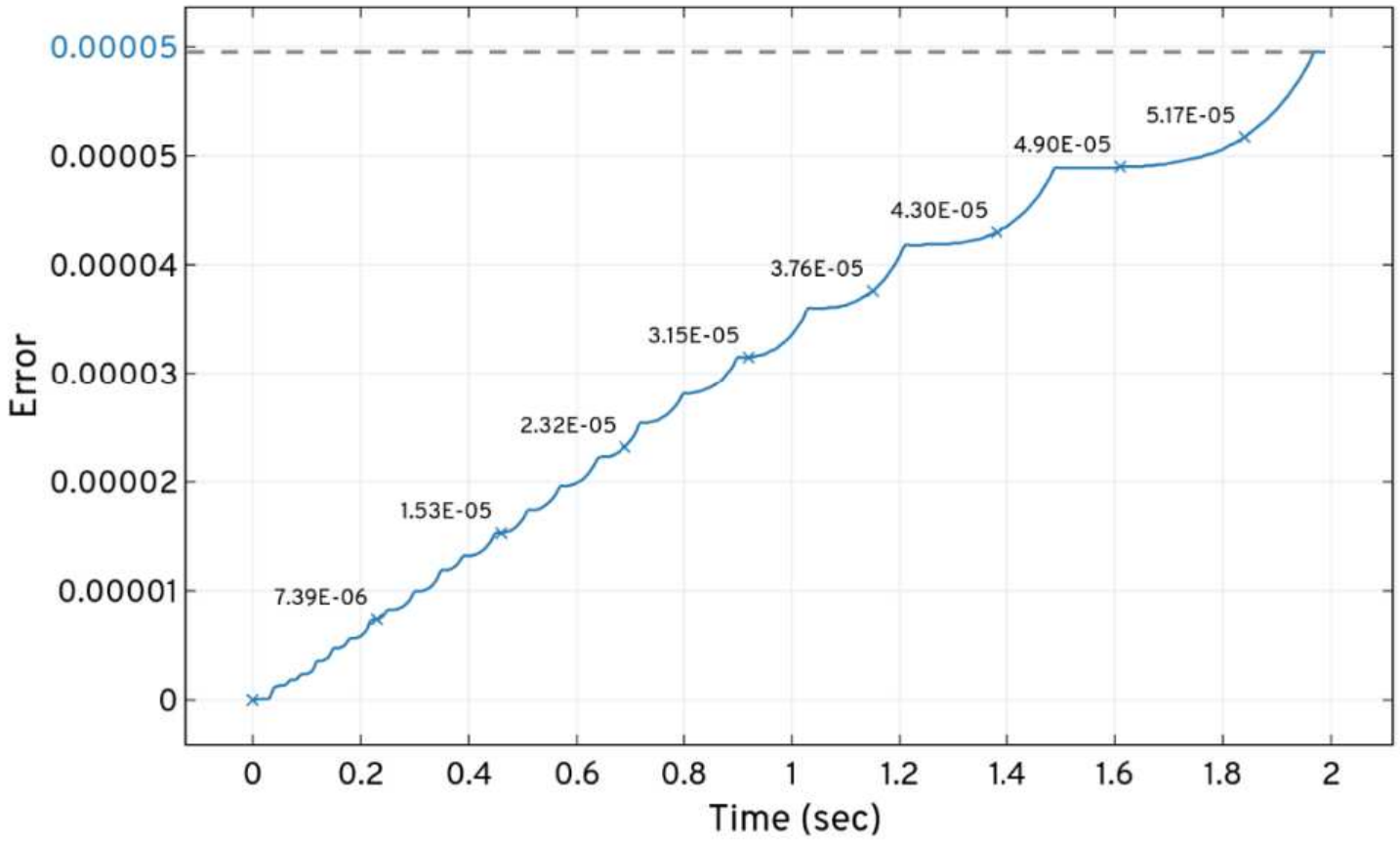

Figure 11

Total probability of states bunked at ' from the domain produced by dual enzymatic reactions system in iteration while expansion and solving the CME.

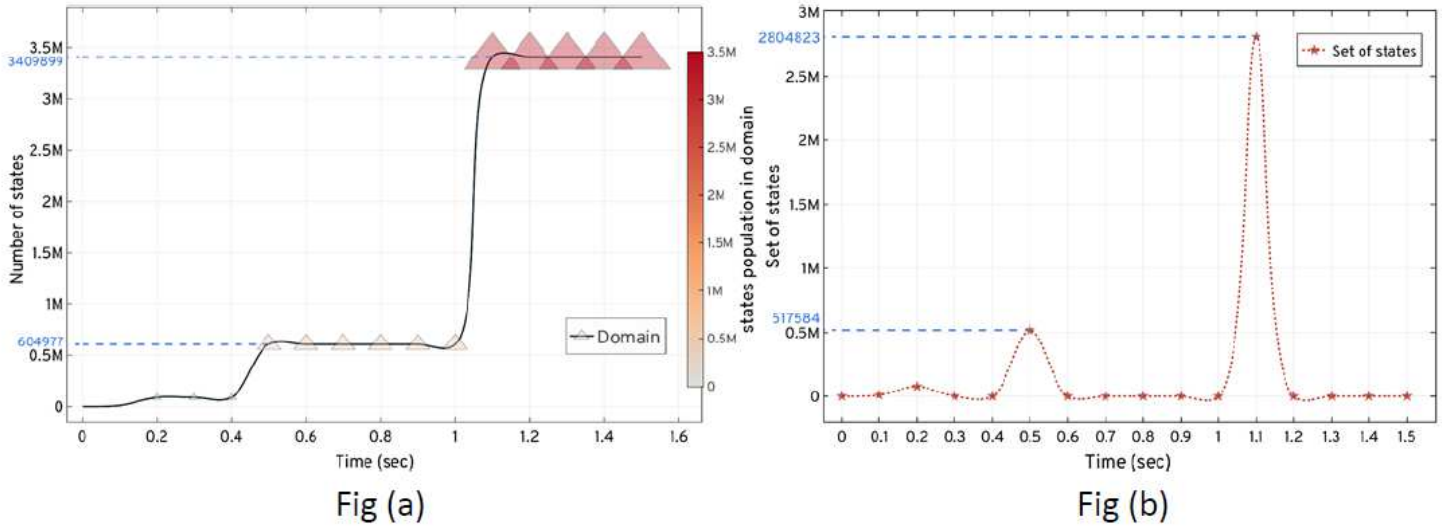

Figure 12

Showing the expansion and updating of the states and set of states explored for the G1/S model based on the method. Fig (a) depicts the state-space expansion increases the number of additions of

new states in the domain. □□□ □□□□□ quickly expands the state-space up to ≈3.5 million states in 1.5 sec. In Fig (b), □□□ □□□□□ unfolds the state- space pattern to update states in the domain and expands 3409899 states up to □□. □ shows the time point where new set of states is explored and updated in the domain.

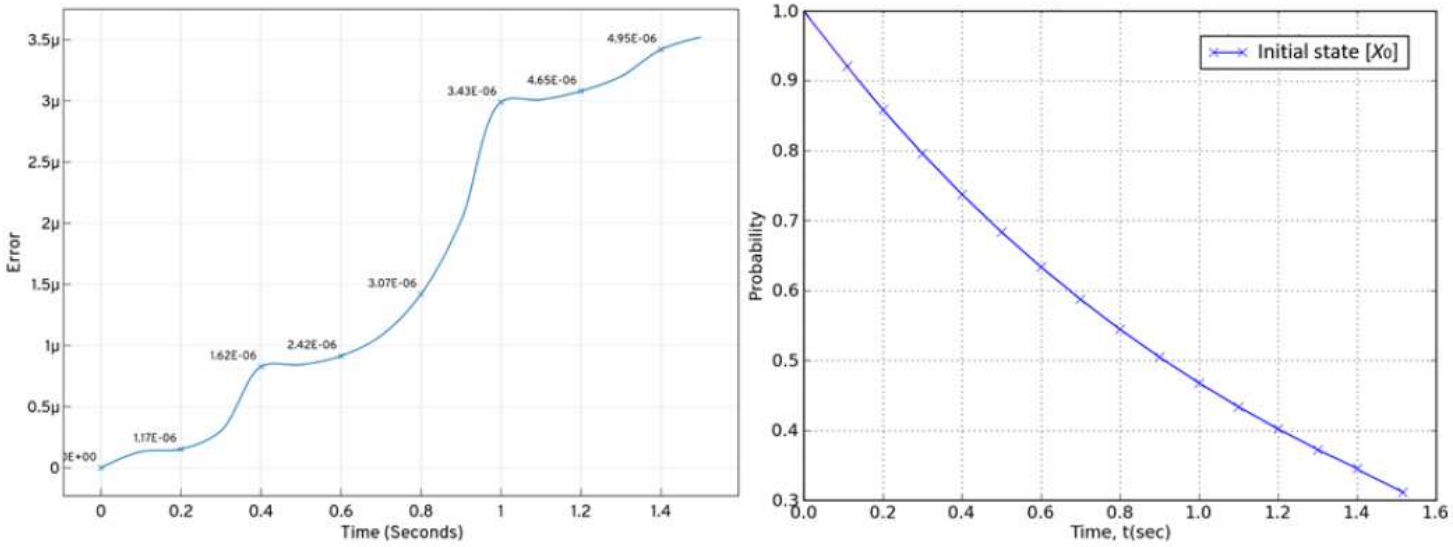

Fig (a) Fig (b)

Figure 13

Response of the □□□ □□□□□ for total probability bunked at □' from the domain and checkpoint for examining the initial state probability over time. Fig (a) shows, how the accuracy of the result is maintained by the □□□ for keeping low errors. Fig (b) shows the decline in probability of the system to remain in the initial state in presence of DNA damage.

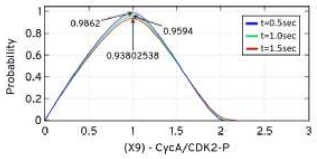

Fig (1). Probability of **CycA/CDK − P** over t_f

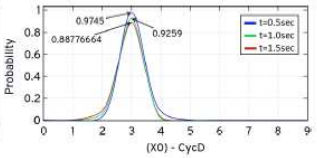

Fig (2). Probability of **CycD** over t_f

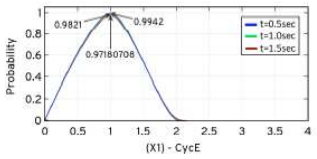

Fig (3). Probability of **CycE** over t_f

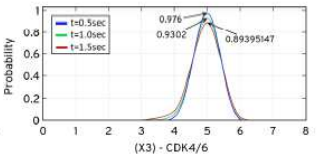

Fig (4). Probability of **CDK4/6** over t_f

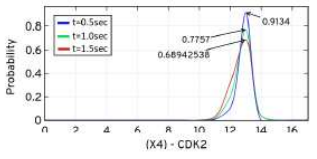

Fig (5). Probability of **CDK2** over t_f

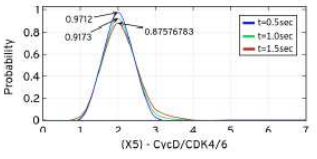

Fig (6). Probability of **CycD/CDK4/6** over t_f

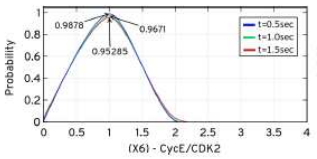

Fig (7). Probability of **CycE/CDK2** over t_f

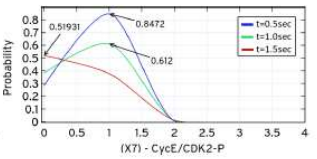

Fig (8). Probability of **CycE/CDK2 − P** over t_f

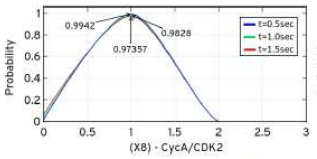

Fig (9). Probability of **CycA/CDK2** over t_f

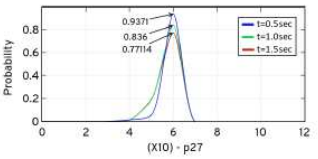

Fig (10). Probability of **p27** over t_f

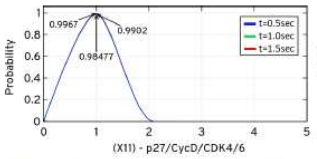

Fig (11). Probability of **p27/CycD/CDK4/6** over t_f

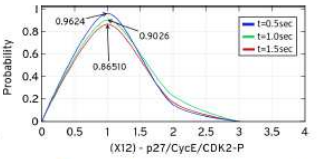

Fig (12). Probability of **p27/CycE/CDK2 − P** over t_f

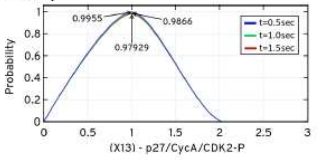

Fig (13). Probability of **p27/CycA/CDK2 − P** over t_f

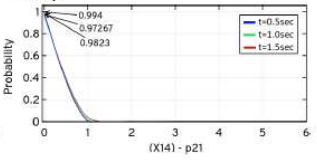

Fig (14). Probability of **p21** over t_f

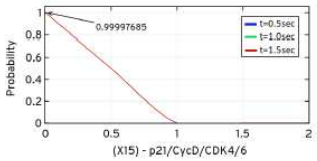

Fig (15). Probability of **p21/CycD/CDK4/6** over t_f

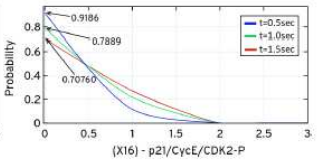

Fig (16). Probability of **p21/CycE/CDK2 − P** over t_f

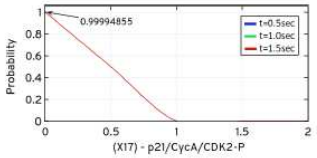

Fig (17). Probability of **p21/CycA/CDK2 − P** over t_f

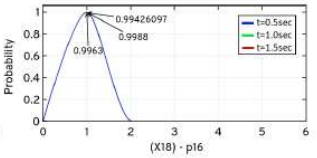

Fig (18). Probability of **p16** over t_f

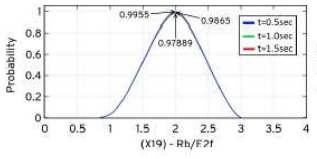

Fig (19). Probability of **Rb/E2f** over t_f

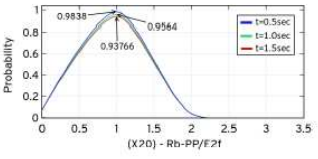

Fig (20). Probability of **Rb − PP/E2f** over t_f

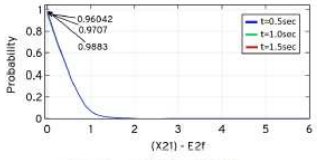

Fig (21). Probability of **E2f** over t_f

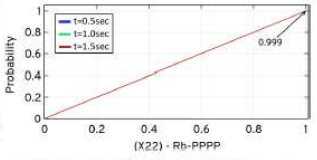

Fig (22). Probability of **Rb − PPPP** over t_f

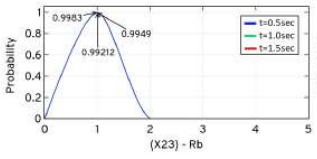

Fig (23). Probability of **Rb** over t_f

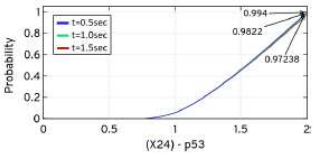

Fig (24). Probability of **p53** over t_f

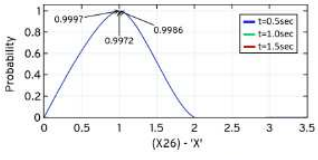

Fig (25). Probability of **'X'** over t_f

Figure 14

Conditional probability of the G1/S model evaluated at □□ = □. □ sec, □□□□□ = □. □ using □□□ □□□□□.

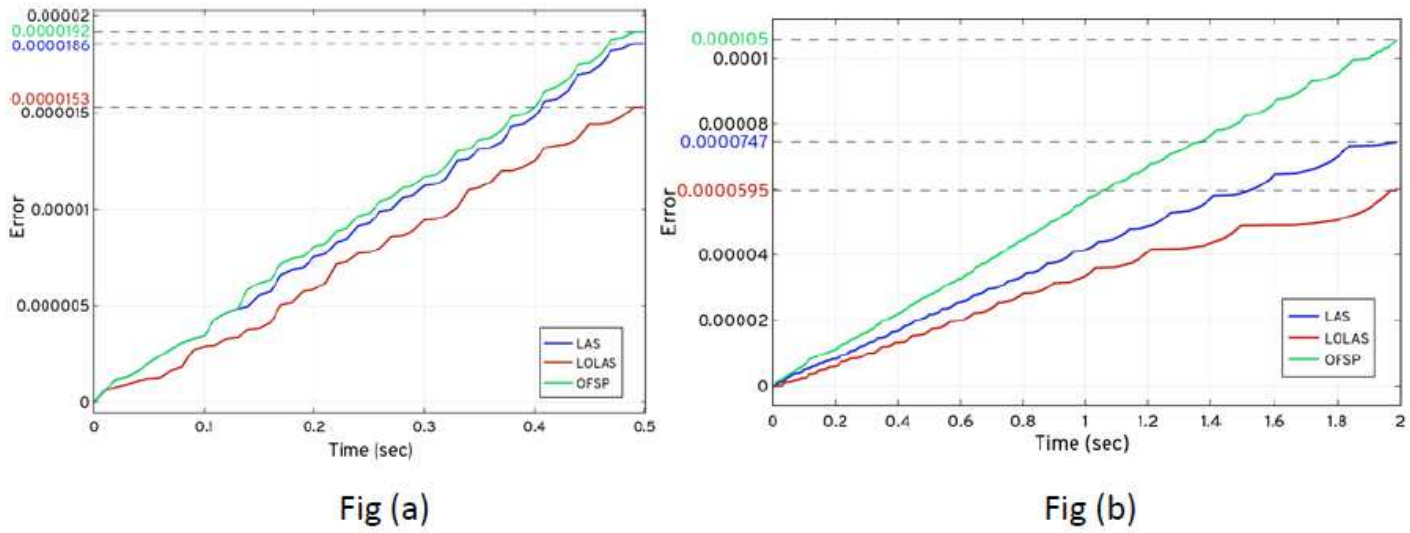

Figure 15

The comparison of [redacted] ([redacted] and [redacted]) with [redacted] based on the solution of the catalytic and dual enzymatic reaction networks.

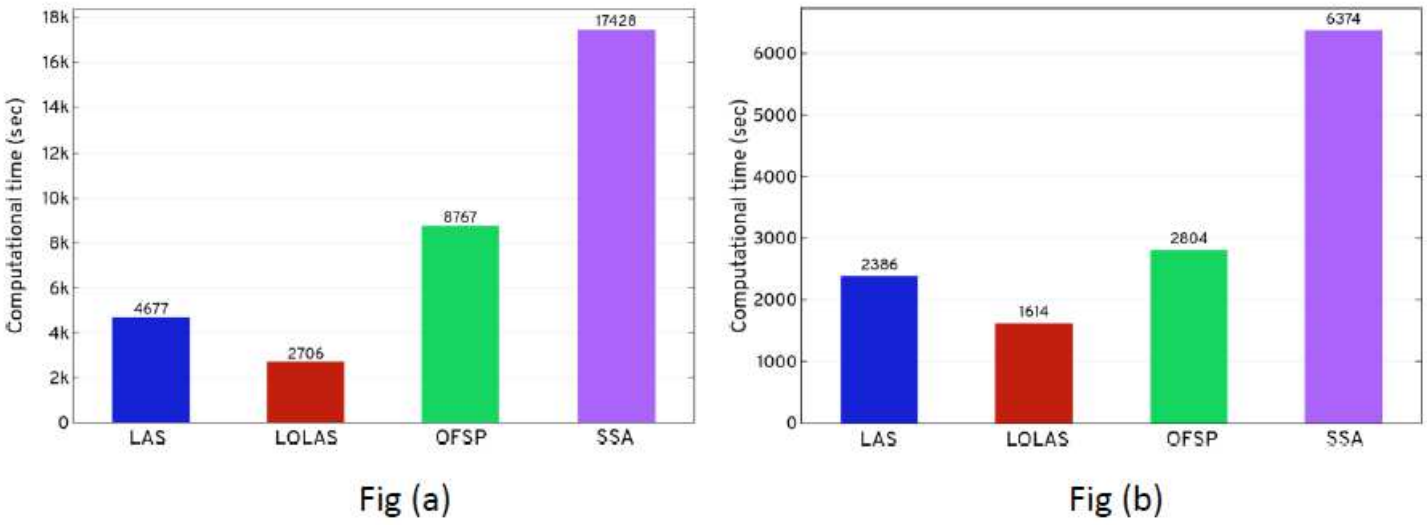

Figure 16

The comparison of [redacted] ([redacted] and [redacted]) with [redacted] and [redacted] by computational time. All methods were applied to the catalytic and dual enzymatic reaction network, respectively, that was previously integrated in Experimental results section.

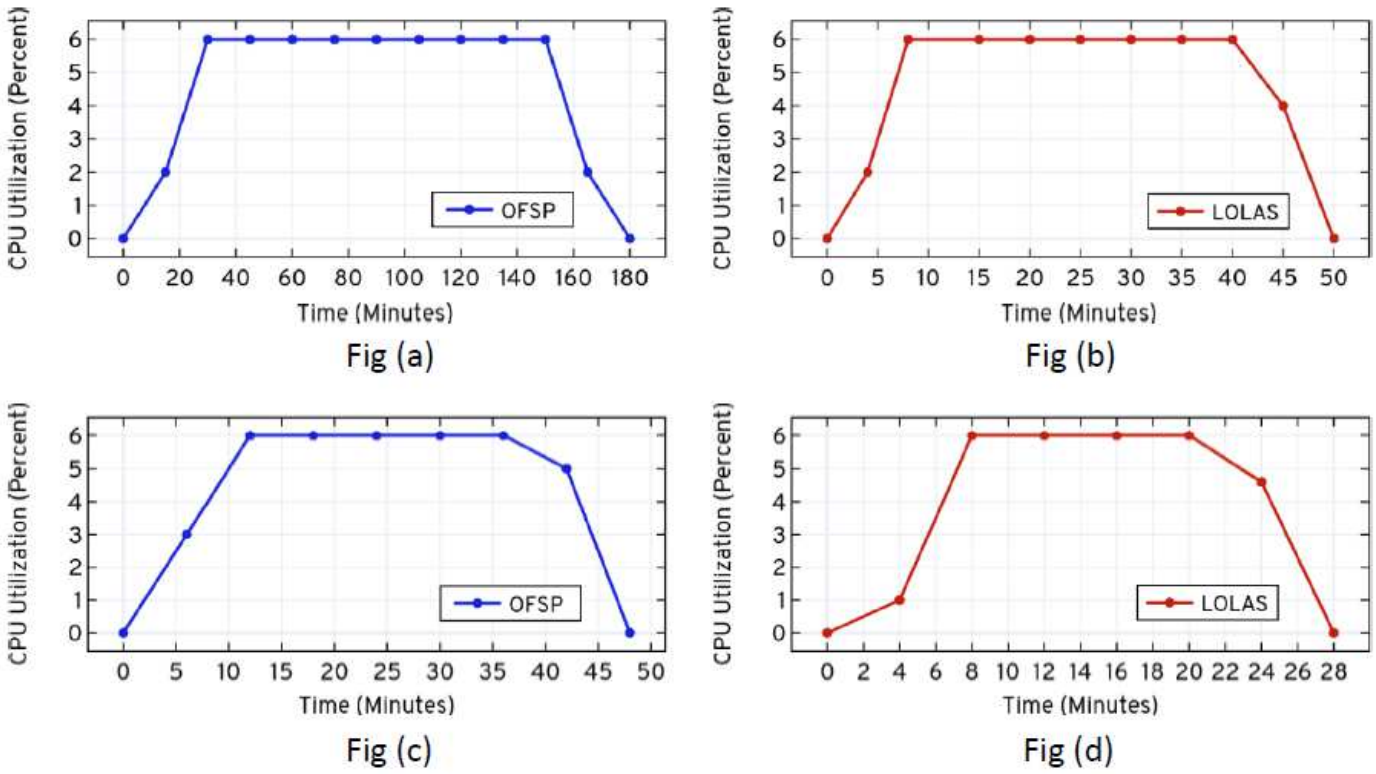

Figure 17

AWS® CPU utilisation percentage, when the catalytic reaction system is solved up to □□ = 0.5 sec and dual enzymatic reaction system solved up to □□ = 2.0 sec using □□□□ and □□□□□. The performance analysis was carried out using CloudWatch® (Statistic: Average, Time Range: Hour, Period: 5 Minutes).

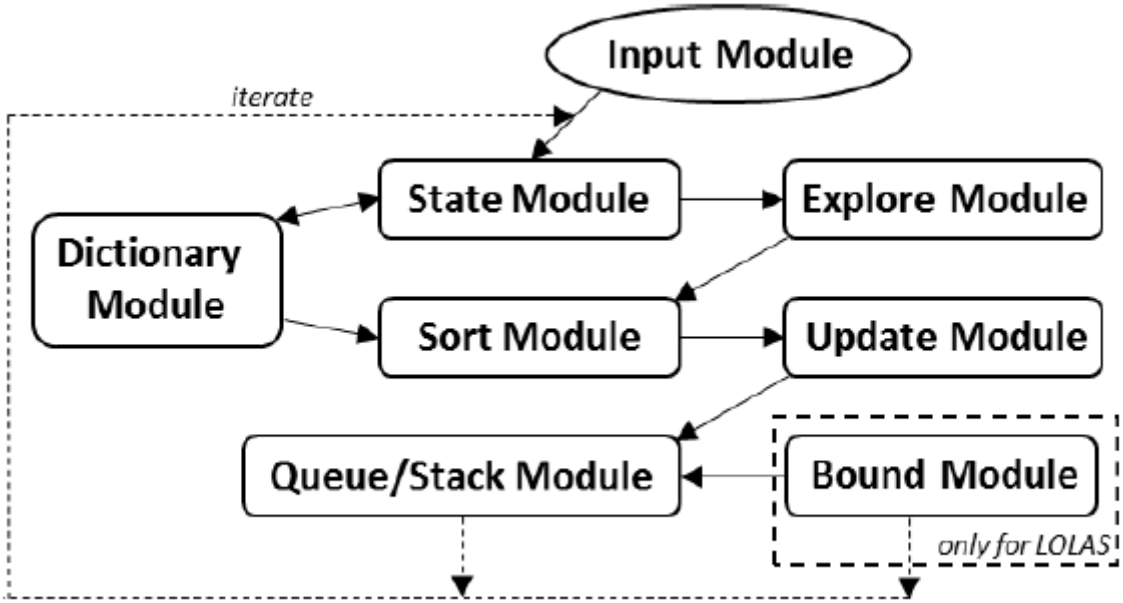

Figure 18

Comprehensive □□□ method flow chart. A description of the modules (steps), sub- modules and the list of components are discussed in SI 5.

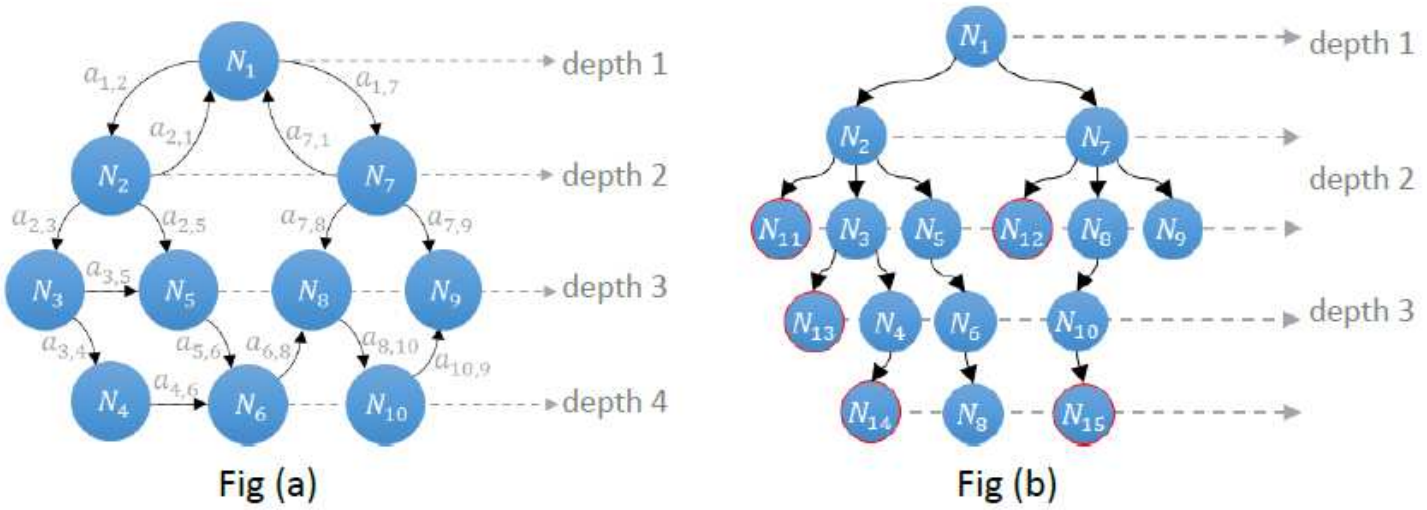

Figure 19

Represents the Markov chain graph and its equivalent tree. Fig (a) depicts Markov chain graph (□□□) with n□ nodes carrying □□ states and the arcs showing transitions between them and all together forming a Markovian process, and Fig (b) depicts equivalent tree ℍ of □□□ as □□□ representing state-space of the system.

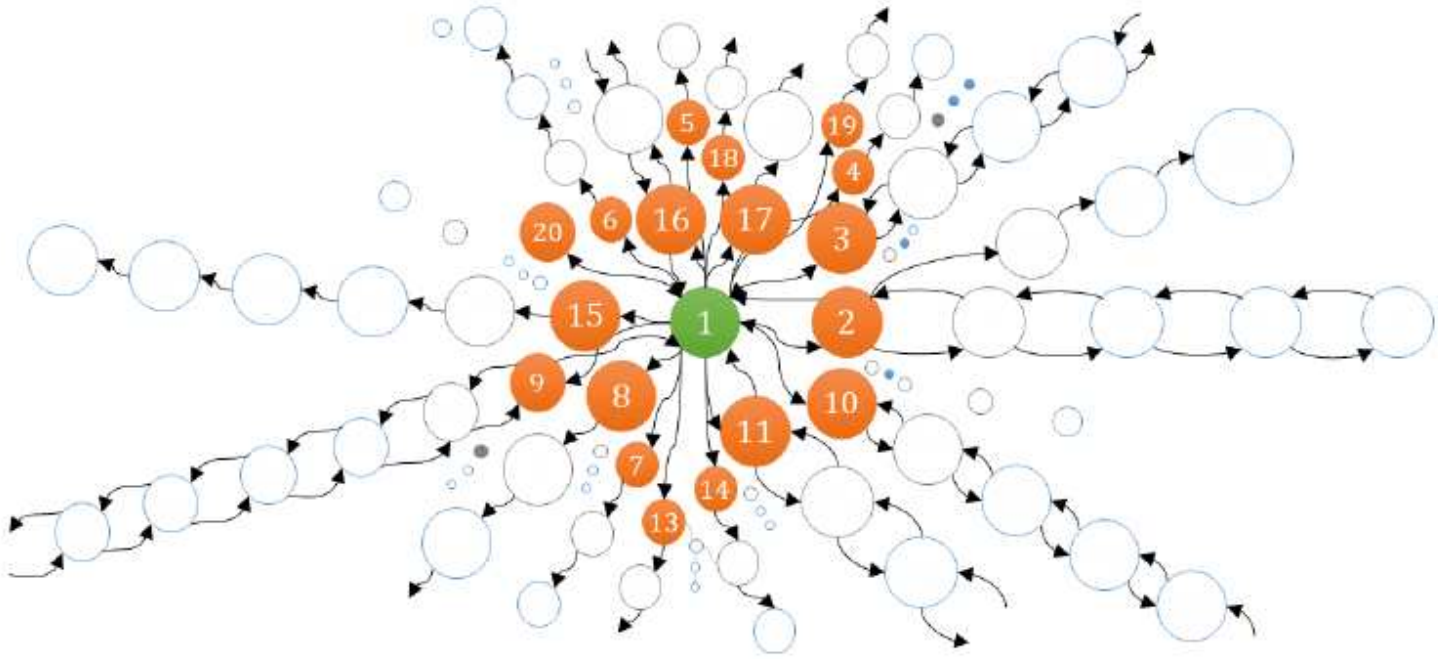

Figure 20

Limits of our visibility in the state-space before expansion. Visualised by a Markov chain graph, where green node is the initial node and orange nodes are directly reachable nodes from the initial node when

exactly one □□ occurs. When a further □□ occur, the system jumps to other empty nodes.

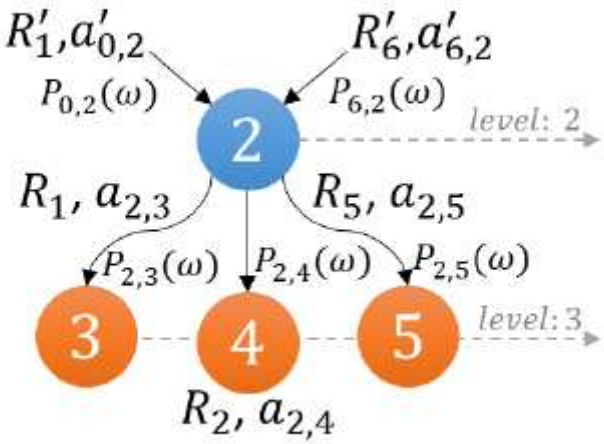

Figure 21

Current □□□□□(□□) = □□, and future □□□□□□(□□,□,□) = (□□,□,□, □□=□) with corresponding reactions □□, □□, □□ and assumed propensities □□,□ = □□, □□,□ = □□, □□,□ = □□, respectively, at any time □, given □□,□ = □. □□□□, □□,□ = □. □□□□.

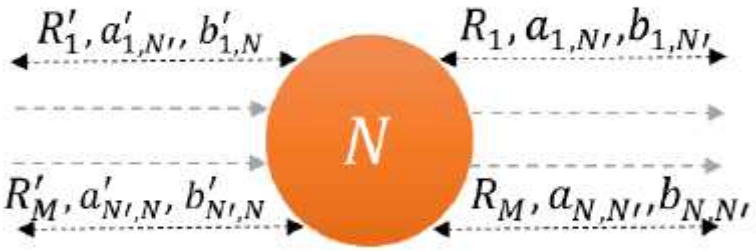

Figure 22

Node □ as a junction of forward and backward reactions □□, where □' ,….,□' are propensities of the prior reactions' and □' ,….,□' are the likelihood of □,□' □,□' □,□ □',□ the prior reactions.

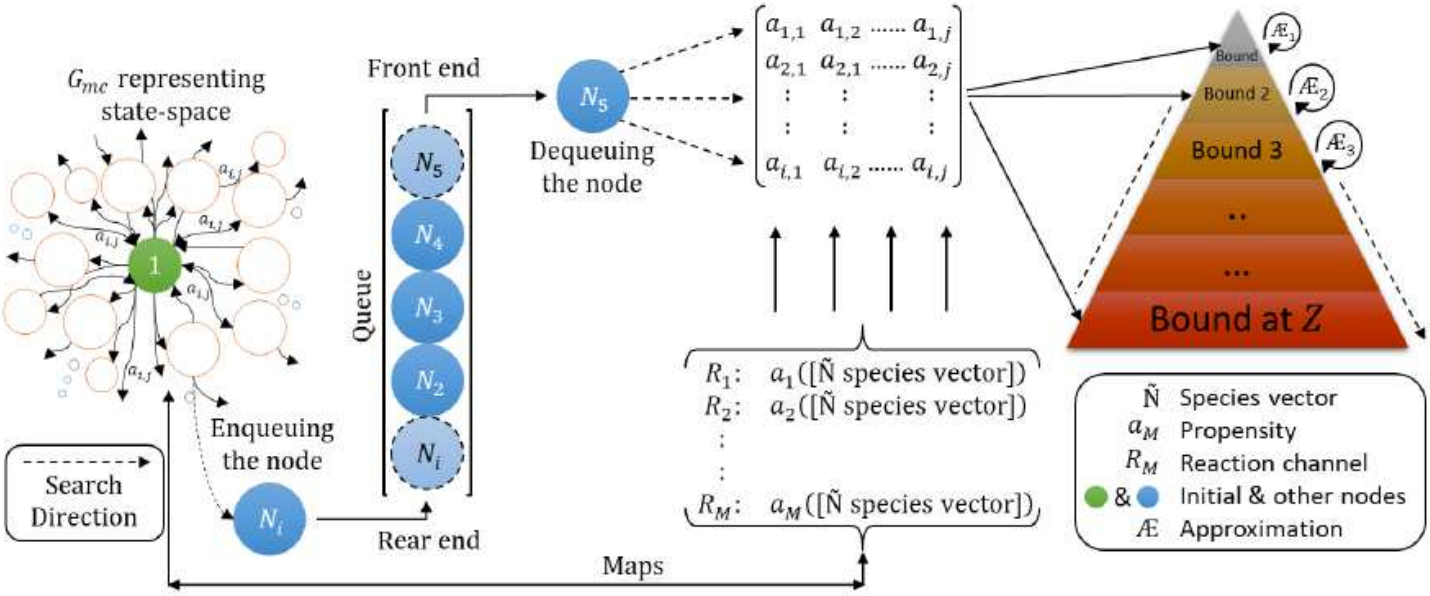

Figure 23

Infrastructure of the Latitudinal Search strategy, showing [illegible], the [illegible] and the domain.

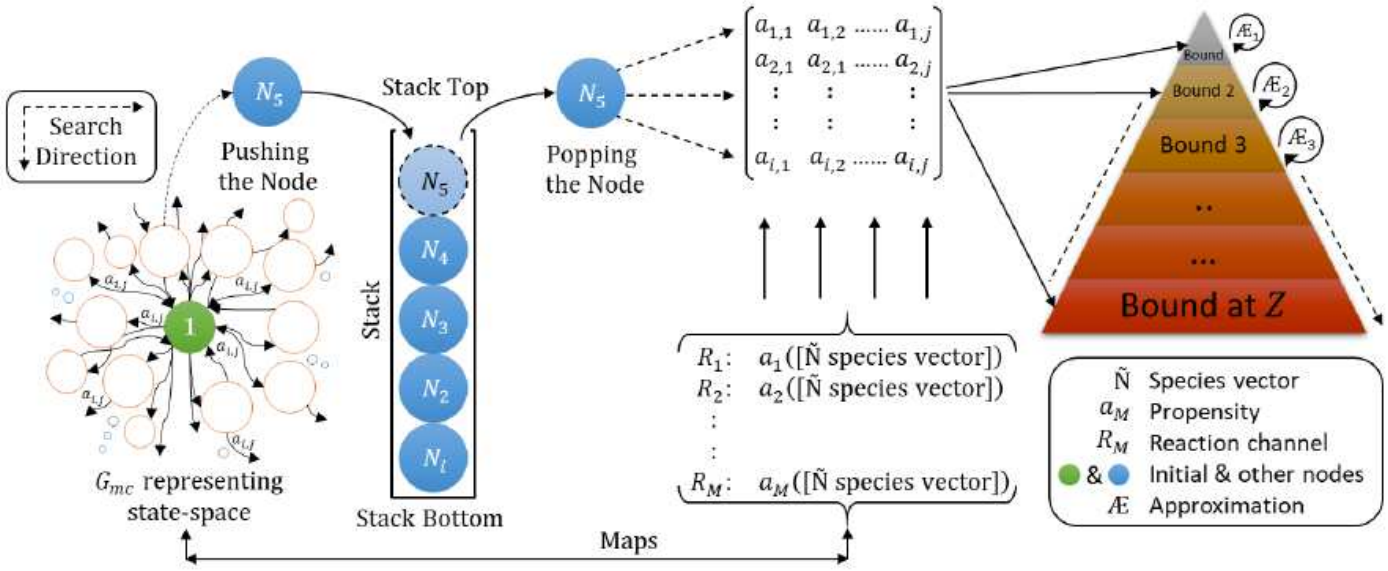

Figure 24

Infrastructure of the Longitudinal Latitudinal Search strategy, showing the [illegible], the [illegible] and the domain.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- BMCSupportingInformationKosarwalKulasiriSamarasinghe.pdf