Overcoming the lack of kinetic information in biochemical reactions networks

Niccolò Totis
University of Turin, Computer Science Department
Corso Svizzera 185
Turin, Italy
ntotis@di.unito.it

Laura Follia
University of Turin, Molecular Biotechnology and Health Sciences Department
Via Santena 5
Turin, Italy
laura.follia@unito.it

Chiara Riganti
University of Turin, Oncology Department
Via Santena 5/bis
Turin, Italy
chiara.riganti@unito.it

Francesco Novelli
University of Turin, Molecular Biotechnology and Health Sciences Department
Via Santena 5
Turin, Italy
franco.novelli@unito.it

Francesca Cordero
University of Turin, Computer Science Department
Corso Svizzera 185
Turin, Italy
fcordero@di.unito.it

Marco Beccuti
University of Turin, Computer Science Department
Corso Svizzera 185
Turin, Italy
beccuti@di.unito.it

ABSTRACT

A main aspect in computational modelling of biological systems is the determination of model structure and model parameters. Due to economical and technical reasons, only part of these details are well characterized, while the rest are unknown. To deal with this difficulty, many reverse engineering and parameter estimation methods have been proposed in the literature, however these methods often need an amount of experimental data not always available.

In this paper we propose an alternative approach, which overcomes model indetermination solving an Optimization Problem (OP) with an objective function that, similarly to Flux Balance Analysis, is derived from an empirical biological knowledge and does not require large amounts of data. The system behaviour is described by a set of Ordinary Differential Equations (ODE). Model indetermination is resolved selecting time-varying coefficients that maximize/minimize the objective function at each ODE integration step. Moreover, to facilitate the modelling phase we provide a graphical formalism, based on Petri Nets, which can be used to derive the corresponding ODEs and OP. Finally, the approach is illustrated on a case study focused on cancer metabolism.

1. INTRODUCTION

The inherent complexity of biological systems makes the study of their dynamics and behaviours a difficult issue. Within this context, computational modelling can help biologists/clinicians to identify which molecules and interactions have a crucial role on the global behaviour of the system. Many modelling approaches have been proposed in the literature. An appropriate choice among them depends on the specific biological question being addressed and on the available data. An extensive review of the most successful modelling approaches used for metabolic, and more in general for any biochemical network, is reported in [4], where the authors divide these approaches into three classes (i.e. *interaction-based*, *constraint-based* and *mechanism-based* approaches) depending on the used level of abstraction. For instance, *interaction-based* approaches do not consider any quantitative aspect, so that they allow only a topological analysis of the system. Instead, *constraint-based* approaches rely on the assumption of steady state and typically exploit Flux Balance Analysis (FBA) and its extensions [18] to study the system at equilibrium. *Mechanism-based* approaches describe interactions with high level of detail and are the most likely candidates to provide a complete understanding of biochemical dynamics.

In order to build a mechanism-based the modeller has to specify all its characteristics, like, for instance, the structure of the system, the mechanism of the events that produce changes in variables and the mathematical expressions used to represent them, with all their parameters defined. This considerable amount of preliminary information requires the modeller to acquire an extensive knowledge of the biological phenomena and to formulate all the assumptions on which the model relies. In this article we will focus on a particular type of biochemical systems, namely metabolic systems, and on modelling these systems with Ordinary Differential Equations (ODEs). Due to economical and technical reasons, the complete information needed to build the set of ODEs is rarely available, causing the modeller to face situations of indetermination. In this study we will consider in particular those metabolic models in which part of the reactions are fully characterized while the rest are not. Two groups of reactions are thus defined: determined reactions, having well characterized structure, mathematical expression and parametrization, and undetermined reactions, described just by their stoichiometry.

Issues of indetermination have been already deeply studied in literature and solutions have been proposed at different levels, depending on the type of missing information. Tech-

niques of reverse engineering (RE) and Parameter Estimation (PE), for instance, overcome this indetermination by means of available experimental data and optimization algorithms [4, 15, 8].
In our previous work [22] we proposed an alternative approach to deal with situations in which insufficient experimental data hamper the application of PE and RE methods. To overcome the lack of information concerning undetermined reactions we chose to exploit an empirical biological knowledge that helps us formulate some modelling assumptions and define an optimization problem (OP). Since the direct definition of the ODEs and of the OP may be difficult and may require advanced modelling skills, we facilitate this process providing a high level graphical formalism based on Petri Nets (PNs).

In this article, we reviewed the approach presented in [22]. In detail we propose two new solution algorithms to speedup the solution process. Moreover, the performance of these new algorithms, in terms of accuracy and computational cost, are compared with the implementation proposed in [22] through a new set of experiments. The paper is organized as follows. Section 2 gives a general outlook on other methods currently used for similar purposes, while Section 3 provides the necessary background on PNs and OPs. In Section 4 our approach is introduced: a formal definition of our PN extension and the procedure to automatically translate a model described through such a high level formalism into the underlying ODE system and OP are provided. In Section 6 we conclude presenting some future works.

2. STATE OF THE ART

As mentioned in the introduction, situations of inadequate biochemical characterization are traditionally managed with **Reverse Engineering** (RE) or **Parameter Estimation** (PE) computational techniques. When the topology of the network is not known, several techniques of RE can be used. Besides topology, the parametrization of the model represents at present the main area of lacking information. The parameters required for a kinetic model include both the kinetic parameters and the initial conditions. Many parameters have been experimentally calculated with biochemical assays, and have been reported in literature, but many others are still unavailable. As some authors have pointed out, if we consider the parametrization of the mathematical expressions of enzyme activity: "for the foreseeable future, full availability of *true* rate equations for all enzymes is certainly an illusion" [3]. To cope with this situation, many PE techniques have proposed different approaches that aim to automatically infer model parameters from experimental data. In general, the inputs of PE methods include network topology, initial conditions and discrete-time time series of some biochemical species present in the model. When parameters and network topology are unknown, PE has to be embedded in the solution of a Reverse Engineering problem [4]. Instead, if parameters and initial conditions are unavailable, a joint state-kinetic constants estimation (see,e.g., [13]) has to be performed. PE performs a global optimization of a fitness function which evaluates the distance between the simulated solution and the experimental data points. Several alternative optimization algorithms can be found in literature, each adopting a specific approach to escape from local minima and to limit the computational complexity. The difficulty of PE lies in the availability of experimental data, which in many real cases are very scarce. In this paper we present an approach that can be used in situations where both part of the network structure and part of the parametrization are unknown, representing an alternative to a PE embedded in a RE process. Differently from these techniques, our method tries to escape a strict dependency from experimental data using an OP that relies on biological empirical knowledge. This OP exploits an objective function that is conceived to represent a specific biological behaviour or phenotype, as it will be discussed in Section 5.

The use of an objective function with a biological meaning has already been incorporated in **Flux Balance Analysis** and **Cybernetic Modelling** (CM). In most FBA studies the objective function describes an objective that we believe is physiologically relevant for the cell, like the maximisation of biomass. This is used to predict a physiological behaviour of the cell. The idea that a system, like a cell, seeks a biological objective is justified acknowledging the apparent goal-directedness of nature. Other computational approaches, like cybernetic modelling [25], have taken advantage of this concept, which has been formally defined with the word teleonomy. A teleonomic view of nature indeed considers that the genetic programs of all living organisms have been shaped by evolution in a way that highly performing phenotypes have been selected [25]. Both FBA and cybernetic modelling have proved that this perspective can be used in different ways to simulate real biological behaviours. However, referring to the modelling categories defined in the introduction, FBA pertains to the group of *constraint-based* models, which are able to offer only a representation of the system at steady state. In FBA, the assumption is that the period in which the concentrations of biochemical species fluctuate can be neglected, while primary relevance is attributed to the condition in which the system is at equilibrium, i.e. when concentrations do not change in time any more. A further improvement of FBA is represented by **dynamic Flux Balance Analysis** (dFBA), proposed by Mahadevan and coworkers [20]. Here the extracellular space is modelled dynamically while inside the cell the steady-state assumption holds. In dFBA an ODE system is built to describe all extracellular events while the distribution of intracellular fluxes is calculated via a static optimization algorithm (SOA) or a dynamic optimization algorithm (DOA). For reasons of similarity with our approach we will focus specifically on the SOA algorithm, in which a biological objective function, like cellular growth rate or biomass, is maximized at each integration step of the ODEs. It is worth highlighting that dFBA always considers the intracellular system at steady state, so it neglects all the dynamic fluctuations of internal metabolites. In the field of mechanism based models, the teleonomic perspective was adopted by cybernetic models [25], which have been presented in a long series of progressive refinements. Besides teleonomy, the CM approach also assumes that (i) the topology of the network is completely known, that (ii)metabolic enzymes are regulated at the gene expression level or via allosteric mechanisms, that (iii) some economic principles govern the way the cell implements these regulations, meaning that the cell, as if it were a rational investor, enhances those reactions that assure the highest yield in terms of the objective function. Finally CM assumes that (iv) accurate mathematical expressions with a complete list of kinetic parameters are

available. These assumptions limit the good predictions of cybernetic models only to situations in which all parameters are available or where sufficient experimental data allow PE methods to be employed.

3. BACKGROUND

In this section we first introduce a brief overview of the (Stochastic) PN (SPN) formalism highlighting how it is possible to derive from an SPN model an ODE system that can be used to study the system's behaviour. Subsequently, we will re-call the basis of the OPs and their solution techniques which will be used in the sequel of this paper.

Stochastic Petri Nets. PN and their extensions are graphical modelling formalisms which are becoming quite popular to build models of biological systems. The reason of this appeal is their capabilities of representing in a simple and intuitive manner many important features of these systems and thus of constructing models that are easy to understand also by non-mathematicians and non-computer scientists. PNs are bipartite directed graphs with two types of nodes called *places* and *transitions*. The *places*, graphically represented as circles, correspond to the system variables (e.g. enzymes and compounds), while the transitions, graphically represented as rectangles, encode the events (e.g. interactions among biochemical entities) which cause the system evolution. Arcs connecting places to transitions (and vice versa) express the relations between states and event occurrences. Places can contain tokens, graphically represented as black dots, that in the context of systems biology, often describe the number of molecules of the corresponding entities. An example of a PN model is shown in Fig. 2 which describes glycolysis in human red blood cells.

The state of a PN, called *marking*, is defined as the number of tokens in each place of the net. An example of marking for the PN in Fig. 2 is showed in the third column of the Table in Fig. 3 **.** The system evolution is given by the occurrence of enabled transitions, where a transition is enabled if each input place contains a number of tokens greater or equal than a given threshold defined by the multiplicity of the corresponding input arc. A transition occurrence, called *firing*, removes a fixed number of tokens from its input places and adds a fixed number of tokens to its output places. The multiplicities of the input/output arcs determine the number of tokens involved by transition firings. The set of all the markings, that a net can reach through transition firings from an initial marking, is called the *Reachability Set* (RS). Instead, the behaviour of the net is encoded by means of the *Reachability Graph* (RG), a directed graph whose nodes are the markings of the RS and whose arcs are tagged with the labels of the transitions that cause the corresponding marking changes.

Temporal specifications must be introduced to model and study the temporal dynamics of a PN. Several timed extensions have been proposed in the literature [19]. In this paper we focus on Stochastic PN (SPN) [16], in which exponentially distributed random delays (interpreted as durations of certain activities) are associated with transition firings. Thanks to this assumption the temporal behaviour of the system can be modelled with a random process governed by the so-called Chapman-Kolmogorov differential equations [5]. These equations correspond to the Master Chemical Equations [6] that are used to describe the behaviour of biological systems, thus making this formalism quite attractive for these types of applications. Specifically, the underlying stochastic process corresponds to a Continuous Time Markov Chain (CTMC) that can be represented as a graph isomorphic to the RG of the net. Formally an SPN can be defined as follows:

DEFINITION 1. *A stochastic Petri net system is a tuple*

$$\mathcal{N} = (P, T, I, O, \mathbf{m}_0, \lambda)$$

where:

- $P = \{p_i\}_{1 \le i \le n_p}$ *is a finite and non empty set of* places *of cardinality* n_p*;*
- $T = \{t_i\}_{1 \le i \le n_t}$ *is a finite, non empty set of* transitions *with cardinality* n_t *and such that* $P \cap T = \emptyset$*. All these transitions are Timed transitions which fire with a random delay characterized by a negative exponential probability distribution;*
- $I, O : P \times T \to \mathbb{N}$ *are the* input, output *functions that define the arcs of the net and that specify their multiplicities;*
- $\mathbf{m_0} : P \to \mathbb{N}$ *is a multiset on* P *representing the* initial marking*; the notation* $m_0(p_i)$ *specifies the initial marking of the place* p_i*;*
- $\lambda : T \to \mathbb{R}$ *gives the* firing intensities *of the transitions.*

The values assumed by the functions I and O can be collected in $n_p \times n_t$ matrices (which we still call I and O) and whose entries are $I(p_i, t_j)$ and $O(p_i, t_j)$, respectively. By $I(t)$ we denote the column of I corresponding to transition t (the same holds for O). The set of input places of transition t (i.e. the preset of t), denoted ${}^\bullet t$, and the set of output places of t (i.e. postset of t), denoted $t^\bullet$, are defined as follows: ${}^\bullet t = \{p \in P \mid I(p,t) \neq 0\}$, and $t^\bullet = \{p \in P \mid O(p,t) \neq 0\}$.

From SPN to ODE. It often happens that, in case of very complex models, the underlying CTMC can not be derived or/and solved due to the well-known state space explosion problem. To cope with this difficulty, whenever the stochasticity of the modelled system can be neglected (e.g. due to huge number of molecules), the so-called deterministic approach can be exploited, assuming that the behaviour of entities contained in a place of the net is described with an Ordinary Differential Equation (ODE) and that the whole model is specified with a system of ODEs, one for each place of the net. In the literature, different laws (e.g. Michaelis-Menten, Hill-equation, etc.) have been proposed to encode each reaction of the biological system into an ODE. Here we focus on the Mass Action (MA) law [23][1] in which the

[1]Observe that this choice does not affect the generality of our approach that can be applied independently of the assumed law.

ODEs describing the model have the following form:

$$\frac{dx_{p_i}(\nu)}{d\nu} = \sum_{j:O(p_i,t_j)\neq 0} O(p_i,t_j)\lambda(t_j) \prod_{h:I(p_h,t_j)\neq 0} x_{p_h}(\nu)^{I(p_h,t_j)} - \sum_{j:I(p_i,t_j)\neq 0} I(p_i,t_j)\lambda(t_j) \prod_{h:I(p_h,t_j)\neq 0} x_{p_h}(\nu)^{I(p_h,t_j)} \quad (1)$$

where $x_{p_i}(\nu)$ represents the amount of the entity in place p_i at time ν assuming that $x_{p_i}(0)$ is defined through the initial marking of the net so that $x_{p_i}(0) = m_0(p_i)$.

For instance, considering the PN model in Fig.2 the behaviour of place *GLC* is described by the following ODE equation assuming the MA law:

$$\frac{dx_{GLC}(\nu)}{d\nu} = +\lambda(K_{F_1}) \cdot x_{HK} \cdot x_{GLC} \cdot x_{ATP} - \lambda(K_{R_1}) \cdot x_{HK} \cdot x_{G6P} \cdot x_{ADP} \quad (2)$$

Optimization problem. In Mathematics, Computer Science, and Operations Research, optimization or mathematical programming consists of minimizing (or maximizing) a function by systematically choosing the values of its variables from a set of feasible possibilities properly exploiting analytical or numerical methods. In Systems Biology optimization is not a new concept since it has been already proposed to reconstruct gene regulatory networks, transcriptional regulatory networks, protein interaction networks, conditional specific sub-networks, and active pathways [11], and to perform FBA. Formally an optimization problem with inequality constrains can be defined as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathcal{F}_{opt}(\mathbf{x}) \\ \text{subject to} \quad & \mathcal{G}_i(\mathbf{x}) \geq b_i, \quad 1 \leq i \leq l \\ & \mathcal{L}_i(\mathbf{x}) \leq c_j, \quad 1 \leq j \leq m \end{aligned}$$

where the vector $\mathbf{x} = (y_1, \ldots, y_n)$ is the *variable vector*, the function $\mathcal{F}_{opt} : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*, the functions $\mathcal{G}_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ and $\mathcal{L}_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ are *inequality constraint functions*, and the constants $b_1, \ldots, b_l, c_1, \ldots, c_m$ are the *bounds* for the constraints. A vector $\mathbf{x}^\bullet$, called *optimal*, is the solution of the OP if, among all vectors that satisfy the constraints, it is that which yields the smallest (largest) value of the optimization function: $\forall \mathbf{z}$ s.t. $\mathcal{G}_1(\mathbf{z}) \geq b_1, \ldots, \mathcal{L}_1(\mathbf{z}) \leq c_m$ we have that $\mathcal{F}_{opt}(\mathbf{z}) \geq \mathcal{F}_{opt}(\mathbf{x}^\bullet)$.

We recall that an OP is called a *linear program* if the objective and constraint functions are linear and *non-linear* otherwise. As shown in the next sections of this paper, we will focus on non-linear programs in which constraints can be non-linear as well. To solve this type of OPs, several algorithms have been proposed in the literature, and the reader can find a complete survey of these methods in [12].

4. OUR APPROACH

Before describing our approach in details, we introduce the biological considerations which motivated our proposal.

First, in our method we assume that both the topology and the kinetic parameters of the network are specified with different level of detail, depending on the biochemical event we are observing. Since in biochemical systems events are generally represented as biochemical reactions, we decide to divide all the reactions into two classes: *determined* reactions and *undetermined* reactions.

We assume that for determined reactions the mechanism of their biochemical interaction has been already properly characterized, so that they can be modelled with mathematical expressions that contain the full lists of kinetic parameters.

Instead, regarding undetermined reactions, we assume that in this part of the network the topology is only partially known, so that while all the reagents, all the products and some modifiers are specified, we do not exclude that other modifiers could be present.

For instance, this assumption can be motivated by many studies that showed how many cells of different organisms own different isoforms of the same enzyme and that every isoform has a specific affinity for a group of molecular regulators [17]. Thus, in a specific cell, the resulting kinetic behaviour of a reaction is a direct consequence of the proportion at which these isoforms are present [17] [2]. In some situations this proportion may hardly be defined, and, even if it is known, the list of modifiers for secondary isoforms is not always clear.

Moreover, for specific conditions, like cancer, cells may present genetic mutations and quantitative alterations or isoform swithces that change the affinity between the enzymes and their regulators, thus affecting the overall enzymatic dynamics. Assuming the molecular interactions of an enzyme are unclear necessarily implies that its activity is not expressible with a fully-parametrized mathematical formulation.

Finally, as anticipated in Section 2, we assume that an objective function with a biological meaning can be profitably used to formulate an OP and to reproduce an actual biological behaviour. From our point of view, this biological objective can be interpreted in different ways. For instance, it can be related to its definition in FBA or CM and formulated similarly. In FBA, according to the teleonomic perspective, the objective function describes an objective that we believe is physiologically relevant for the cell, like the maximisation of biomass. This is used to predict a physiological behaviour of the cell. Inspired by other successful applications of FBA, also in our approach the objective function can express a non-physiological goal of the cell, like the maximisation of ATP production, and used to identify some property of the network, or, it may be used to impose an engineering (e.g. maximisation of the production of a particular amminoacid) or therapeutic (e.g. minimisation of some reaction fluxes vital for cancer cells) objective to a metabolic system. More in general, the objective function is intended to represent any relevant biological behaviour that has been experimentally measured or that has to be achieved. Our approach thus tries to investigate if and how this specific biological phenotype can be reproduced in a biochemical system where the topology and the parametrization are just partially known.

In details we propose a method that takes advantage of an iterative process of optimization: at each simulation step, an objective function with the aforementioned biological meaning allows us to estimate the activities of undetermined reactions, and thus to obtain a complete description of system behaviour.

In order to acheive this, we define a set of time-varying parameters for all undetermined transitions. Thus, while the kinetic parameters of determined transitions maintain their fixed values, only the parameters of T_us are allowed to vary. These are defined with the intent that their changing values

recapitulate the lack of information about the structure and the kinetic parameters of undetermined reactions.

To facilitate the construction of the model we propose a new graphical formalism based on PN, which allows to automatically translate the model into its mathematical representation, consisting of the ODEs system and the Optimization Problem (OP).

According to this we decide to present our approach firstly introducing this new graphical formalism, and then providing its automatic translation into a ODE system in which indeterminate transitions are tackled through an OP. For this purpose we use the model of Fig.2 as a "running example" that we comment in the rest of the paper to discuss the features of this new modelling formalism.

SPN with Indetermination. The formal definition of a new PN extension called Stochastic Petri Net with Indetermination (SPNI) is the following:

DEFINITION 2. *A stochastic Petri net with indetermination is a tuple*

$$\mathcal{N} = (P, T, I, O, \mathbf{m}_0, \lambda_n, \Lambda_u, \mathcal{F}^{\mathcal{N}}_{opt})$$

where:

- $T = T_n \cup T_u$ *is a finite, non-empty set of* timed transitions *with* $T_n \cap T_u = \emptyset$. T_n *is the set of* determined transitions, *while* T_u *is the set of* undetermined transitions.
- $\lambda_n : T_n \to \mathbb{R}$ *gives the* firing intensity *of* T_n *transitions.*
- $\Lambda_u : T_u \to \mathbb{R}^2$ *gives the range of variation of the flux of* T_u *transitions.*
- $\mathcal{F}^{\mathcal{N}}_{opt} : T \times P \to \mathbb{R}$ *is an* objective function *whose terms are represented by place markings and transition firing intensities.*

We use the notation $\Lambda^L_u(t)$ (resp. $\Lambda^U_u(t)$) to denote the lower (resp. upper) bound values of the interval in which the flux of a $t \in T_u$ can vary; $\Lambda_u(t)$ then represents a possible flux value of the (undetermined) transition t compatible with its specified lower and upper bounds.

From SPNI to ODE and OP.
Due to the indetermination associated with the T_u transitions, it is not possible to directly use Eq. 1 to represent the deterministic behavior of an SPNI model. We can however re-write Eq. 1 as follows:

$$\frac{dx_{p_i}(\nu)}{d\nu} = \sum_{j:O(p_i,t_j)\neq 0} O(p_i,t_j)\mathcal{M}_{t_j}(\nu) \prod_{h:I(p_h,t_j)\neq 0} x_{p_h}(\nu)^{I(p_h,t_j)} - \sum_{j:I(p_i,t_j)\neq 0} I(p_i,t_j)\mathcal{M}_{t_j}(\nu) \prod_{h:I(p_h,t_j)\neq 0} x_{p_h}(\nu)^{I(p_h,t_j)} \quad (3)$$

where $\mathcal{M}$ is a function defined in the following way:

$$\mathcal{M}_t(\nu) = \begin{cases} \lambda_n(t) & if \ \ t \in T_n \\ y_t(\nu) & otherwise \end{cases} \quad (4)$$

The parameter $y_t(\nu)$ encodes the indetermination associated with the undetermined transition t at time ν and must be properly estimated to solve the ODE system.

As we already pointed out, the undetermined transitions are part of the model either because their exact specification is not relevant with respect to the goals of the analysis carried on with the SPNI or because they are too difficult to identify in a precise manner. Independently of the context of the modelling experiment, it is usually the case that we want to minimize (or maximize) certain measures defined on the portion of the state of the system that is not directly affected by undetermined transitions. These measures, that may assume complex definitions, become the optimization functions that we use to study these models.

To cope with this problem we thus propose to exploit an optimization process in which the objective function depends on the solution of the ODE systems in Eq.3. In practice, the optimization process solves the ODE system for a specific time interval while, simultaneously, it uses the obtained solution to compute the objective function of the optimization problem. The maximum/minimum value of the objective function allows to identify the values of unknown parameters of undetermined reactions.

Given a SPNI model, the corresponding OP, whose solution will be used to estimate the firing intensity values of the T_us, is derived using the following definition.

DEFINITION 3. *The OP derived by the SPNI is a tuple*

$$\mathcal{O} = (\mathbf{y}_\nu, \mathcal{F}_{opt}, \mathcal{G}, \mathcal{L})$$

where:

- $\mathbf{y}_\nu$ *represents the optimizing values of undetermined transitions at time* ν, *i.e.* $\forall t \in T_u \Rightarrow y_t(\nu) \in \mathbf{y}_\nu$;
- $\mathcal{F}_{opt} = \mathcal{F}^{\mathcal{N}}_{opt}$;
- $\mathcal{G}$ *is defined by*

$$\forall t \in T_u \Rightarrow y_t(\nu) \prod_{h:I(p_h,t)\neq 0} x_{p_h}(\nu)^{I(p_h,t)} \geq \Lambda^L_u(t)$$

- $\mathcal{L}$ *is defined by*

$$\forall t \in T_u \Rightarrow y_t(\nu) \prod_{h:I(p_h,t_j)\neq 0} x_{p_h}(\nu)^{I(p_h,t_j)} \leq \Lambda^U_u(t)$$

For instance, considering the SPNI in Fig. 2, where the gray boxes highlight the transitions affected by indetermination, the vector $\mathbf{y}(\nu)$ has size six and represents the optimal values of the firing rates of transitions T_{uf1}, T_{ur1}, T_{uf3}, T_{ur3}, T_{uf12}, T_{ur12}. An example of objective function could be the maximization of the Lactate (LAC) as described in our case study in Section 5.

Moreover in our example
$\Lambda^L_u(T_{u_{f_1}}) = 1620 \cdot x_{HK} \cdot x_{GLC} \cdot x_{ATP}$
$\Lambda^U_u(T_{u_{f_1}}) = 2.592e+08 \cdot x_{HK} \cdot x_{GLC} \cdot x_{ATP}$,
with limit values chosen as explained in Section 5.

How to compute the model behavior. Let $\mathbf{x}(\nu)$ represent the behaviour of the model at time ν . The numerical integration of Eq. 3 provides the behaviour of the model at time $\nu+\tau$, in terms of the behaviour $\mathbf{x}(\nu)$ computed at time ν and of a set of parameters deriving from the structure of the SPNI (I, and O), the firing intensities of the definite transitions of the net (λ_n) and of the firing intensities of the undetermined transitions estimated at time ν and collectively represented as $y(\nu)$. The values of $\mathbf{x}(\nu+\tau)$ are thus the results of the evaluation of a function whose input

Algorithm 1 Algorithm to solve ODE system with Indetermination

```
1: function SolveODEI(ODEI, G, L, F_opt, y_t, τ, FinalTime)
2:     ν = 0.0;
3:     ODEI.Init(Value);
4:     while (ν ≤ FinalTime) do
5:         print(ν, Value);
6:         Res=SolveOP(y_t, Value, ODEI, ν + τ, G, L, F_opt);
7:         Value=Res.Value;
8:         y_t=Res.y_t;
9:         ν += τ;
10:    end while
11: end function
```

parameters are represented by a tuple $B(\nu) = (B, B_u(\nu))$ where $B = (I, O, \lambda_n)$ and $B_u(\nu) = (\mathbf{x}(\nu), \mathbf{y}(\nu))$ [2]. The integration step s identifies the time points $\nu_i = i * \tau$ where the evaluation of the model behaviour is of interest.

Role of the estimation phase of our method is that of finding a set $\mathbf{y}(\nu)$ that, being compatible with the constraints of the SPNI model (Λ_u), minimizes the objective function at time $\nu + s$. The optimization phase identifies a number K of initial conditions, that we denote with $B_u^{[k]}(\nu), k = 1, ..., K$, consisting of the behaviour of the model computed at time ν and of K random points within the space of firing intensities of the undetermined transitions identified by the constraints Λ. From each of these configurations the method numerically integrates the system of ODEs up to time $\nu + s$ to derive $\mathbf{y}(\nu)$. Letting $B^{[k]}(\nu) = \left(B, B_u^{[k]}(\nu)\right)$, with $B_u^{[k]}(\nu) = (\mathbf{x}(\nu), \mathbf{y}^{[k]}(\nu))$, the solutions obtained from the integration of the ODEs with parameters $B^{[k]}(\nu), k = 0, 1, ..., K$ and up to time $\nu + s$ are compared to identify the choice of $B^{[k]}(\nu)$ which provides the best evaluation of the objective function, thus identifying $\mathbf{y}^{[k]}(\nu + s) = \mathbf{y}(\nu)$. Crucial in this optimization step is that the numerical integration of the ODEs is performed with a method capable of identifying an integration step h small enough to allow a precise solution of the ODEs during these "tentative" evaluations that are used to select the firing intensities of T_us.

In general, this whole method is repeated for each time point ν_i starting from $\nu_0 = 0$. However, solving the OP for each value of ν_i can be excessively costly and we can thus reduce this computational effort by identifying a time interval ρ that is a multiple of τ and that determines the time points where the optimization is requested. By doing so, if we set $\rho = m \cdot \tau$, we assume that for $m - 1$ intermediate evaluation steps the values of $\mathbf{\Lambda}_u(\nu)$ (i.e. $\mathbf{y}(\nu)$) remain constant and an approximation is introduced.

Having discussed how to derive from an SPNI model (i) an ODE system with indetermination (see Eq.3), and (ii) an OP (see Def.3), we can devise an algorithm which combines them to derive the model behaviour.

The pseudo-code of this algorithm is shown in Alg. 1. It takes as input the ODE system with indetermination (i.e. *ODEI*), the OP (i.e. described by functions $\mathcal{G}$, $\mathcal{L}$ and $\mathcal{F}_{opt}$), the initial guess for the rate of undetermined transition (i.e. y_t), the step size used for the optimization schema (i.e. τ), and the final time (i.e. *FinalTime*). The output of the algorithm is represented by the values generated for each system entity at different time points (i.e. ν_i). In details, the method *Init()* at line 3 initializes the vector *Value* encoding the initial values assumed for all the entities of the model. Then, the code from line 7 to line 13 is repeated until the time horizon is not reached. In each iteration the function *print()* is called to print the current values of the system entities. Subsequently, the function *SolveOP()* solves the optimization and returns the new values of the system entities and of the rates of T_us (i.e. $Res.Value$ and $Res.y_t$ respectively). It takes as input an initial guess for the rate of T_us (i.e. y_t), the current values of the entities (i.e. *Value*), the final time in which the objective function will be evaluated (i.e. $\nu + \tau$), the ODE system (i.e. *ODE*), and a set of functions encoding the OP (i.e. $\mathcal{G}, \mathcal{L}$ and $\mathcal{F}_{opt}$). The functions $\mathcal{G}$ and $\mathcal{L}$ are used by the optimization solver to test if a new vector $\mathbf{y}$, randomly generated according to the parameter constrains, is a feasible solution. Indeed the functions $\mathcal{G}$ and $\mathcal{L}$ verify if $\mathbf{y}$ satisfies the inequality constraints. The function $\mathcal{F}_{opt}$ is instead called by the optimization solver to compute the value of the objective function associated with a feasible vector $\mathbf{y}$.

This function, takes as input the vector $\mathbf{y}$, the current values of the entities (i.e. *Value*), the ODE system (i.e. *ODE*), and the final time in which the objective function must be evaluated (i.e. $\nu + \tau$). First it computes the quantities values at $\nu + \tau$ assuming the missing rates to be equal to $\mathbf{y}$. Then, the computed values are used to evaluate the objective function, whose derived value is returned. When the optimization step is terminated the vector *Value* is updated with the new computed values.

Moreover, in Alg. 2 we report an extension of the previous pseudo-code in which the optimization solver could be executed less frequently, so not at each time step. Indeed, we propose to exploit a heuristic function to decide when the optimization phase must be performed. Hence, when the optimization solver is not called the previous value y_t are considered during the solution of ODE system (i.e. method *SolveODE()*). In Section 5 an example of such a heuristic function is discussed and some experimental results are presented.

Algorithm 2 Algorithm to solve ODE system with Indetermination integrated with the heuristic function

```
1: function SolveODEI(ODEI, G, L, F_opt, y_t, τ, FinalTime)
2:     ν = 0.0;
3:     ODEI.Init(Value);
4:     while (ν ≤ FinalTime) do
5:         print(ν, Value);
6:         if Heurist(Value,time) then
7:             Res=SolveOP(y_t, Value, ODEI, ν + τ, G, L, F_opt);
8:             Value=Res.Value;
9:             y_t=Res.y_t;
10:        else
11:            Value=ODE.SolveODE(Value, ν + τ, y_u);
12:        end if
13:        ν += τ;
14:    end while
15: end function
```

5. EXPERIMENTAL RESULTS AND DISCUSSION

Our implementation. To perform our experiments a prototype implementation of the proposed method integrated

[2] In the sequel of the paper we will indifferently use $\mathbf{y}_t(\nu)$ or $\Lambda_u(t, \nu)$ to represent the undetermined parameters of our models as provided by the optimization problem at time ν.

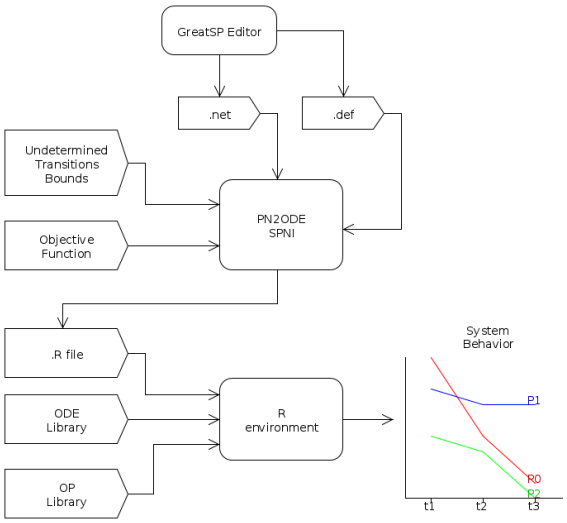

Figure 1: PN2ODE translation process.

in the *GreatSPN* framework [1] was developed. In detail, we extended the *GreatSPN* tool PN2ODE providing the automatic translation from SPNI to the ODEs system and OP system. A scheme representing this process is reported in Fig. 1. The generated ODEs system and OP are encoded in R language and saved into a file that is processed through the *R environment* to obtain the system behaviour. The R package *deSolve* [10] is used to solve ODEs integration, while package *GenSA* [21] is used to solve the OP.
The translation tool takes as inputs:

- the SPNI model, drawn with the GreatSPN GUI and saved into two files with extension .net and .def;
- One text file listing the undetermined bounded transitions;
- One text file containing the objective function.

The translation is performed with the following pipeline:

- The program extrapolates from .net and .def files all the PN information, such as places, transitions, arcs, initial marking and firing rates. Unkown transition rates are marked as NA.
- The program verifies the correctness of the file containing undetermined bounded transitions. For each undetermined transition T_u two values $\Lambda_u^L(t)$ and $\Lambda_u^U(t)$ that bound its flux are required. Optionally, the starting point, from which the OP solver starts searching the optimal solution, can be specified. If no starting point is provided, then a default value is computed as half of the sum of $\Lambda_u^L(t)$ and $\Lambda_u^H(t)$.
- The objective function, stored in the .txt file, is processed through a lex and yacc parsing tool. It can be a generic expression whose terms are the places and the transitions of the net.
- The whole translation process is executed from the command line as follows: PN2ODE SPNI_file_name -M -P -T ./transitions_file -F ./obj_fun_file_name , where -M enables Mass Action policy, -P enables export format in R with the optimizer, while -T and -F are respectively used to specify the two text files containing the undetermined bounded transitions and the objective funtion.

Case study. The proposed approach is used to investigate the metabolic behaviour of cancer cells to illustrate its practical applicability. The model represents the glycolytic pathway in a generic human cell. It is inspired by the model presented in [9], which describes glycolysis in human red blood cells. Glycolysis is the most important and best studied intracellular metabolic pathway. In every cell of the human body, it leads to the consumption of Glucose (GLC) and a progressive production of Pyruvate (PYR) and energy, in the form of Adenosine Triphosphate (ATP). Then, in physiological conditions, in the presence of oxygen, PYR is metabolised by other pathways to generate the majority of the energy consumed by the cell. In absence of oxygen, PYR is converted to LAC without further energetic yields. The model is characterized by seventeen metabolic reactions, the related equations are reported in the first column of the Table in Fig. 3, and it can be graphically described by the SPN model in Fig. 2 where place names are chosen to recall the corresponding biological compounds. The first and the last transitions are included to reproduce the inflow of GLC and the outflow of LAC in and from the cell. All the other transitions describe forward and reverse reactions, catalized by specific metabolic enzymes.

Differently form normal cells, cancer cells exhibit an enhancement of glycolysis and production of LAC even in the presence of oxygen, a phenomenon known as Warburg Effect [7]. This phenomenon represents the central focus of our experiments. It has been recently shown that metabolic alterations seen in cancer cells are promoted by specific mixtures of isoforms of their metabolic enzymes. In particular, it seems that isoforms of **Hexokinase** (HK), **Phosphofructokinase** (PFK) and **Pyruvate Kinase** (PK) may play an eminent role [14]. Despite these discoveries, it is still complicated to characterize the *in vivo* kinetics of these isoforms. Conditioned by these constraints, we chose to set the reactions involving HK, PFK and PK as undetermined transitions, i.e deficient of a complete list of regulators and of a specific mathematical expression containing its kinetic parameters. Our approach is used here as an attempt to acquire a deeper understanding of cancer metabolic dynamics. The idea is to use an objective function that encodes the Warburg Effect, considering every type of cancer at every possible tumour stage. We decided to formalize it as the maximization of LAC production at every integration step. Thus, the optimization process searches the values of the firing intensities of all T_us that allow to maximize LAC. The fluxes of undetermined transitions $T_{u_{f_1}}$, $T_{u_{r_1}}$, $T_{u_{f_3}}$, $T_{u_{r_3}}$, $T_{u_{f_{12}}}$ and $T_{u_{r_{12}}}$ were allowed to vary in a wide range that agrees with the available biological knowledge. Specifically the boundary conditions were set as follows:

$$\Lambda_u^L(T_{u_{f_1}}) = 1620 \cdot x_{HK} \cdot x_{GLC} \cdot x_{ATP}$$
$$\Lambda_u^U(T_{u_{f_1}}) = 2.592e+08 \cdot x_{HK} \cdot x_{GLC} \cdot x_{ATP}$$
$$\Lambda_u^L(T_{u_{r_1}}) = 12.24 \cdot x_{HK} \cdot x_{G6P} \cdot x_{ADP}$$
$$\Lambda_u^U(T_{u_{r_1}}) = 1.9584e+06 \cdot x_{HK} \cdot x_{G6P} \cdot x_{ADP}$$

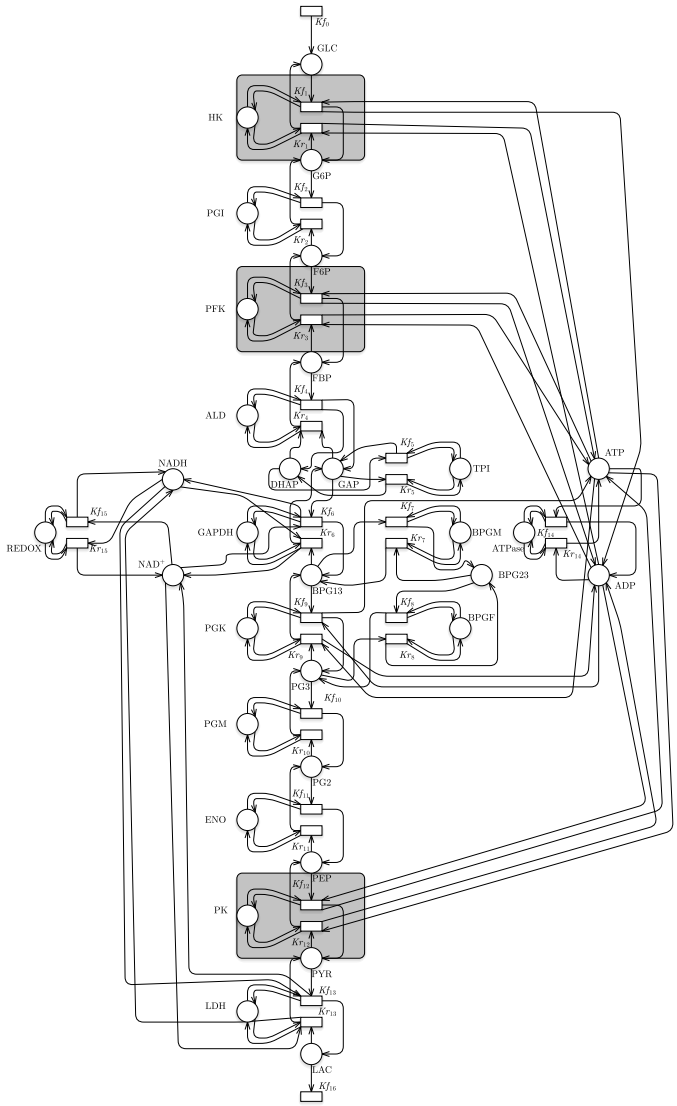

Figure 2: Case study: Glycolysis in *Homo Sapiens*.

$$\Lambda_u^L(T_{u_{f_3}}) = 2.5e+06 \cdot x_{PFK} \cdot x_{F6P} \cdot x_{ATP}$$
$$\Lambda_u^U(T_{u_{f_3}}) = 4e+11 \cdot x_{PFK} \cdot x_{F6P} \cdot x_{ATP}$$
$$\Lambda_u^L(T_{u_{r_3}}) = 211.864 \cdot x_{PFK} \cdot x_{FBP} \cdot x_{ADP}$$
$$\Lambda_u^U(T_{u_{r_3}}) = 3.38983e+07 \cdot x_{PFK} \cdot x_{FBP} \cdot x_{ADP}$$
$$\Lambda_u^L(T_{u_{f_{12}}}) = 1.29234 \cdot x_{PEP} \cdot x_{PK} \cdot x_{ADP}$$
$$\Lambda_u^U(T_{u_{f_{12}}}) = 206774 \cdot x_{PEP} \cdot x_{PK} \cdot x_{ADP}$$
$$\Lambda_u^L(T_{u_{r_{12}}}) = 0.0058511 \cdot x_{PK} \cdot x_{PYR} \cdot x_{ATP}$$
$$\Lambda_u^U(T_{u_{r_{12}}}) = 0.620322 \cdot x_{PK} \cdot x_{PYR} \cdot x_{ATP}$$

Fig. 4 (left) shows the evolution of LAC over time. The black line represents the results of the model of a normal cell, where all parameters are well characterized. The blue dashed line show the time evolution of LAC when uncertainty is applied. It can be noticed that this objective function is able to drive the system to accelerate LAC production. Even if the difference might not seem large enough to represent the Warburg Effect, we point out that these diagrams show the behaviour of our model for a very short time interval. Fig. 4 (right) shows the rapid accumulation of Fructose 6-Phosphate (F6P) in the normal cell model compared to a more balanced production-consumption dynamics in the cancer model. F6P, a high glycolitic intermediate, increases as a direct consequence of GLC degradation and is later processed by PFK. It is then significant to see which parameters the optimization solver independently chooses to tune in order to maximize its objective function. While parameter values of HK and PK did not vary markedly if compared to the normal cell model (data not shown), the highest difference regarded PFK. Many articles as [17] have demonstrated that PFK kinetics is highly non-linear and depends on many allosteric interactions. Our results seem to reinforce the Mulukutla's thesis [17] that the regulation of PFK activity has a crucial impact on the glycolitic flux and may be relevant to explain metabolic alterations in cancer.

With an additional set of experiments we studied if it was possible to reduce the computational costs of our approach while maintaining a good accuracy of the solution. With this intent, we progressively decreased the frequency at which the optimizator was invoked and then explored the performance of our algorithm. When the optimization process is not repeated at every integration step the time-varying parameters associated with T_us are then transformed into piecewise constant parameters. This can be motivated with the assumption that in the time interval between two consecutive optimization processes some fixed parameter values can well approximate the real behaviour of all T_us. The time interval that separates different optimization processes, in alternative called optimization step, was set to four different values: 1e-7, 5e-7, 1e-6, 5e-6 h. As reported in Table 5 the resulting computational times are compared. As expected, reducing the number of optimization processes allowed to significantly diminish the overall computational efforts of the algorithm. We then studied how these changes affected the dynamics of the system. Intriguingly, we found that some places, like LAC, as shown in Fig 5 (left), displayed little or no changes, while for other places, like PEP, the changes were much more relevant, as shown in Fig. 5 (right). We can then observe that for the more sensible places the reduction of computational time comes at the cost of the precision of the solution, which nevertheless maintains the capability to provide some qualitative information about the behaviour of the model.

Reactions	Rate Equations	Initial Marking
$\emptyset \xrightarrow{Kf0} GLC$	$Kf_0 = 6.48E+06$	$GLC_0 = 0$
$HK + GLC + ATP \underset{Kr1}{\overset{Kf1}{\rightleftarrows}} HK + G6P + ADP$	$Kf_1 = 6.48E+05, Kr_1 = 4.9E+03$	$HK_0 = 24$
$PGI + G6P + ATP \underset{Kr2}{\overset{Kf2}{\rightleftarrows}} PGI + F6P$	$Kf_2 = 1.15E+03, Kr_2 = 2,68E+03$	$G6P_0 = 0, PGI_0 = 218$
$PFK + F6P + ATP \underset{Kr3}{\overset{Kf3}{\rightleftarrows}} PFK + FBP + ADP$	$Kf_3 = 1E+09, Kr_3 = 8.47E+04$	$F6P_0 = 0, PFK_0 = 28$
$ALD + FBP \underset{Kr4}{\overset{Kf4}{\rightleftarrows}} ALD + DHAP + GAP$	$Kf_4 = 1.46E+02, Kr_4 = 1.18E+00$	$FBP_0 = 0, ALD_0 = 3,7E+03, DHAP_0 = 0$
$TPI + GAP \underset{Kr5}{\overset{Kf5}{\rightleftarrows}} TPI + DHAP$	$Kf_5 = 7.93E+00, Kr_5 = 4.53E+06$	$TPI_0 = 1.14E+04, GAP_0 = 0$
$GAPDH + GAP + NAD^+ \underset{Kr6}{\overset{Kf6}{\rightleftarrows}} GAPDH + BPG13 + NADH$	$Kf_6 = 1.42E+05, Kr_6 = 5.28E+06$	$GAPDH_0 = 7.6E+02, NAD^+_0 = 1E+03, NADH_0 = 0$
$BPGM + BPG13 \underset{Kr7}{\overset{Kf7}{\rightleftarrows}} BPGM + BPG23$	$Kf_7 = 1E+08, Kr_7 = 1E+05$	$BPGM_0 = 4.10E+02, BPG13_0 = 0$
$BPGF + BPG23 \underset{Kr8}{\overset{Kf8}{\rightleftarrows}} BPGF + PG3$	$Kf_8 = 6.84E+02, Kr_8 = 1E-09$	$BPGF_0 = 4.10E+02, BPG23_0 = 0$
$PGK + BPG13 + ADP \underset{Kr9}{\overset{Kf9}{\rightleftarrows}} PGK + PG3 + ATP$	$Kf_9 = 2.61E+04, Kr_9 = 1.45E+01$	$PGK_0 = 2.74E+02$
$PGM + PG3 \underset{Kr10}{\overset{Kf10}{\rightleftarrows}} PGM + PG2$	$Kf_{10} = 5.38E+01, Kr_{10} = 7.92E+00$	$PGM_0 = 4.10E+02, PG3_0 = 0$
$ENO + PG2 \underset{Kr11}{\overset{Kf11}{\rightleftarrows}} ENO + PEP$	$Kf_{11} = 5.82E+02, Kr_{11} = 3.44E+02$	$ENO_0 = 2.20E+02, PG2_0 = 0$
$PK + PEP + ADP \underset{Kr12}{\overset{Kf12}{\rightleftarrows}} PK + PYR + ATP$	$Kf_{12} = 5.17E+02, Kr_{12} = 5.17E-01$	$PEP_0 = 0, PK_0 = 6.9E+01,$
$LDH + PYR + NADH \underset{Kr13}{\overset{Kf13}{\rightleftarrows}} LDH + LAC + NAD^+$	$Kf_{13} = 1.04E+03, Kr_{13} = 2.34E+00$	$PYR_0 = 0, LDH_0 = 3.12E+02$
$ATPase + ATP \underset{Kr14}{\overset{Kf14}{\rightleftarrows}} ATPase + ADP$	$Kf_{14} = 9.74E-01, Kr_{14} = 9.74E+00$	$ATPase_0 = 1E+02, ATP_0 = 7E+02,$
$REDOX + NAD^+ \underset{Kr15}{\overset{Kf15}{\rightleftarrows}} REDOX + NADH$	$Kf_{15} = 9.74E-01, Kr_{15} = 9.74E-04$	$ADP_0 = 5E+02, REDOX_0 = 97$
$\emptyset \xrightarrow{Kf16} LAC$	$Kf_{16} = 1$	$LAC_0 = 0$

Figure 3: Table: Reactions, Equations and Initial marking of glycolysis in *Homo Sapiens*.

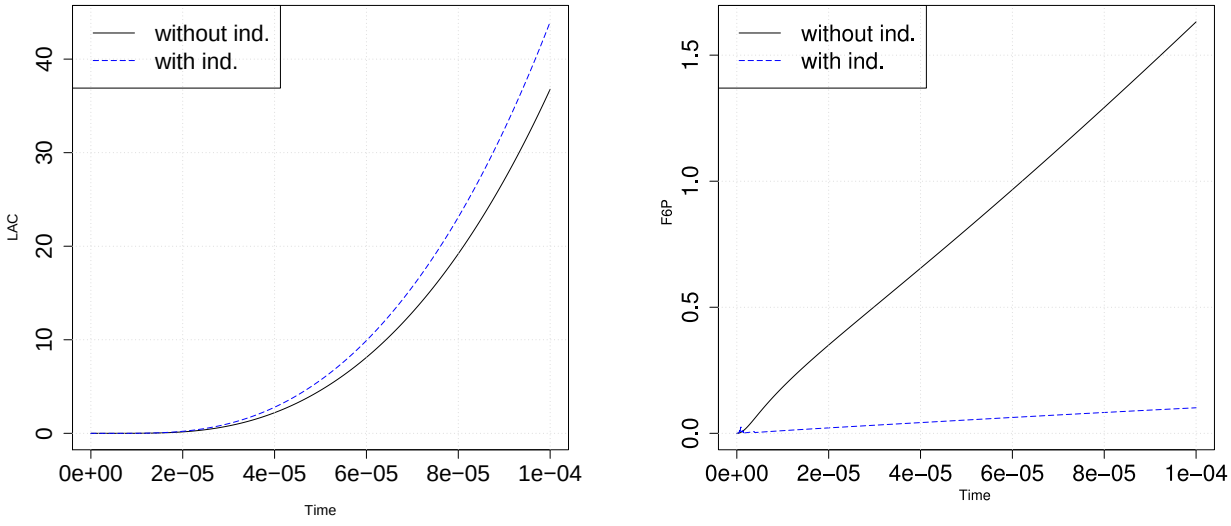

Figure 4: Behaviour of place *LAC* (left) shows that the model representing the metabolism of a cancer cell is able to reproduce a higher production of LAC over time. Significative differences affect the dynamic behaviour of place *F6P* (right), which is less likely to accumulate in the cancer model

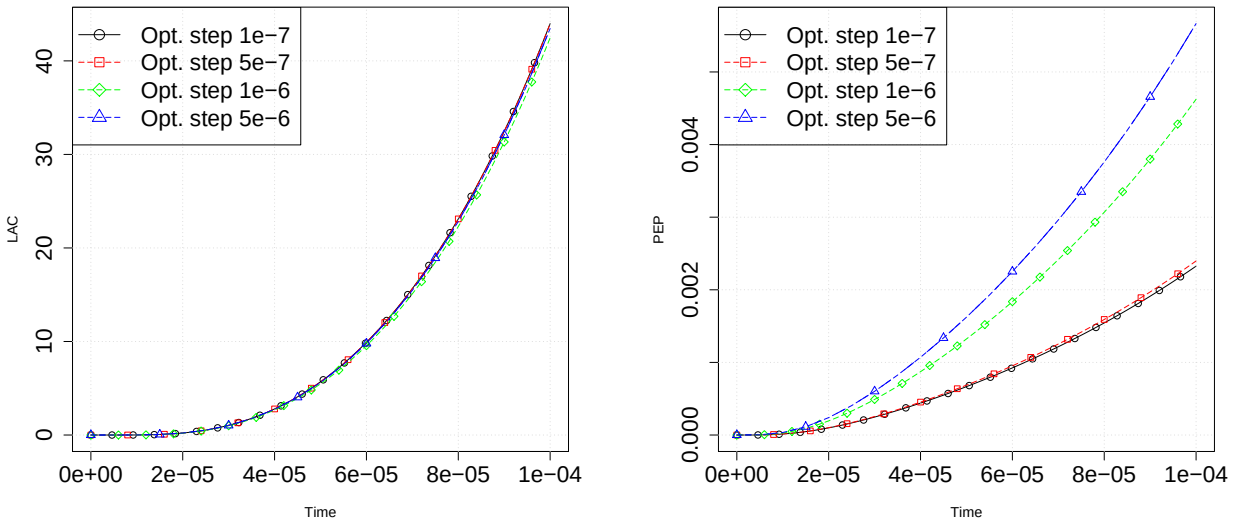

Figure 5: The figures show how, when we try to increase the time inbetween two optimization processes for some places, as for instance LAC (left), the behaviours are almost not affected, while for other, like PEP (right), the difference is non-trivial

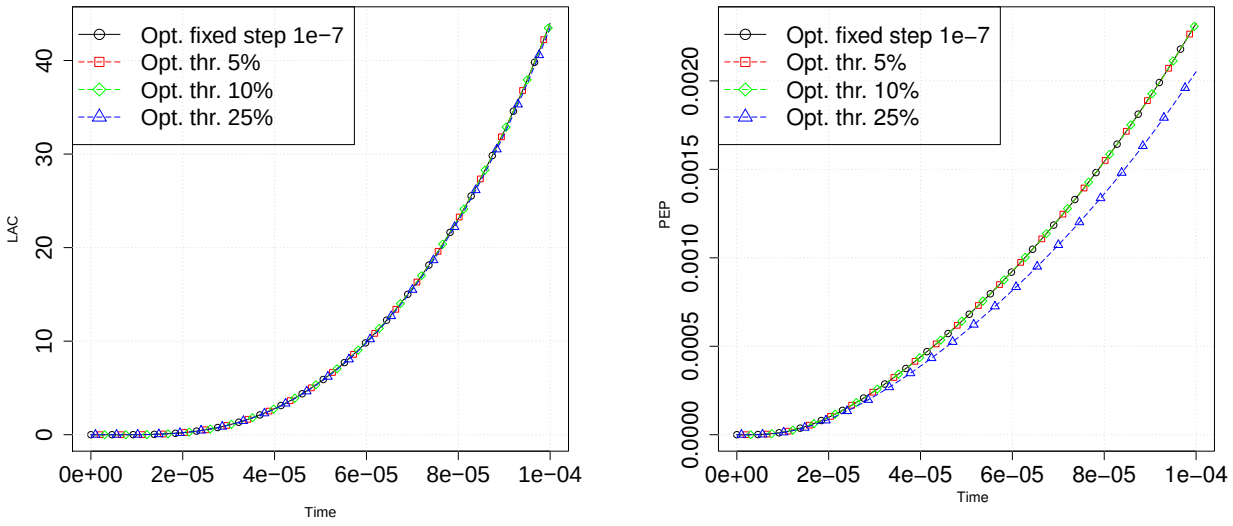

Figure 6: Behaviours of places LAC (left), PEP (right). The figures show that, when the algorithm with the heuristics is used and threshold values of 5 or 10% are chosen, the solution of the model is accurate, while the computation time is considerably reduced

ODE	Opt int 1e-7	Opt int 5e-7
0.732235s.	2441.878sec.	569.7389sec.
	Opt int 1e-6	**Opt int 5e-6**
	254.9282sec.	119.2946sec.

Table 1: Execution times for different optimization time intervals. All executions were performed on an INTEL i7 64bit 2.60GHz processor

In a further set of experiments we tested if the time step of the optimization processes could be adaptively tuned along the solution of the system. The rationale behind this choice is to invoke the optimization process just if it produces significant effects. In order to do it, we added in our algorithm an heuristics, presented in Section 4, that considers the relative change in the markings of input places of all T_us. When the relative change in one of these places crosses a user-defined threshold, the optimization process is executed. We studied how three different thresholds values (i.e. 5%, 10% and 25% of change) impacted both the computational time and the accuracy of the solution. Figure 6 displays the differences in the dynamic behaviours of PEP and LAC, when the different threshold values are considered. The execution times in the three cases are reported in Table 5, compared to the performance of the algorithm in the absence of the implemented heuristics. Data clearly show that, for instance when a 5% threshold value is used, no negative effects reflect on model solutions, while the computational cost is reduced of a factor greater that 10. Regarding the accuracy of the solution, also greater threshold values are satisfactory, with the advantage that they allow considerably higher time savings.

Opt int 1e-7	Thr=5%	Thr=10%	Thr=25%
2441.878sec.	195.0151sec.	135.4613sec.	82.67076sec.

Table 2: Execution times for different threshold values of relative change in place markings. All executions were performed on an INTEL i7 64bit 2.60GHz processor

Discussion. It is important to observe that our approach, moving forward from the strict teleonomic perspective adopted by dFBA and CM, allows to define objective functions that may mimic some experimentally observed behaviour. It is worth underlying that teleonomy has shown great utility to help dissect the metabolism of microorganisms. In the case of multicellular and more complex organisms like humans, however, the process of selection that justifies the teleonomic view is much more complex, and a clear goal-directedness of intracellular metabolism is not equally reasonable to consider. However, the LAC maximisation function, which we adopted in our case study of cancer glycolysis, besides reflecting the fact that a high production of LAC in cancer cells is a renown experimental finding, may also have a teleonomic value. In fact, it can be hypothesized that the process of evolution of cellular phenotypes observed in cancer, i.e. in the selection of aggressive cancer clones during its progression (current focus of extensive studies [24]), tends to select a goal-directed cellular phenotype characterized by LAC maximization.
Moreover our proposed approach, differently from dFBA and CM, does not assume the steady state of the intracellular metabolites, as in dFBA, and it does not require a complete knowledge of all kinetic parameters, as in CM. Finally, it is useful to highlight again that the lack of experimental data, which is very frequent in cancer cell scenarios, does not affect our approach as it does for techniques of RE and PE.

6. CONCLUSIONS AND FUTURE DIRECTIONS

In this work we proposed a new approach to deal with biochemical models with lacking kinetic information. As discussed in Section 5 it may be useful for different applications, although additional experiments will provide a further validation and will show its actual relevance.
For the future we intend to apply our approach to study different networks, either in microorganisms, in single human cells or in population of cells.

As for all OPs, the performances are highly dependent on the objective function, on the constraints that define the space of feasible solutions, and on the algorithm used to explore it. Depending on the specific model, objective functions with different biological meaning, as described in Section 4, will be tested. We will also investigate which available biological data can be effectively used to adapt the model to the specific situation under study. Either data obtained with transcriptomics, proteomics and metabolomics, as well as reaction fluxes and thermodynamic information are all possible candidates. These data will provide a fundamental support to define the initial conditions of the model, to inspire the choice of the objective function and to set the constraints for the optimization. Finally, depending on OP characteristics, the performances of different optimization algorithms will be tested.
In addition, for those cases in which stochastic behaviours are relevant, we plan to expand our approach with systems of stochastic differential equations.

7. ADDITIONAL AUTHORS

Gianfranco Balbo (University of Turin, Computer Science Department, Corso Svizzera 185 Turin Italy, email: *balbo@di.unito.it*)

8. REFERENCES

[1] J. Babar, M. Beccuti, S. Donatelli, and A. S. Miner. Greatspn enhanced with decision diagram data structures. In *Proceedings of 31st Int. Conf. of Applications and Theory of Petri Nets*, pages 308–317. IEEE Computer Society, June 2010.

[2] B. D. Bennett, E. H. Kimball, M. Gao, R. Osterhout, S. J. V. Dien, and J. D. Rabinowit. Absolute metabolite concentrations and implied enzyme active site occupancy in escherichia coli. *Nature Chemical Biology*, 5(8):593–599, 2009.

[3] S. Bulik, S. Grimbs, C. Huthmacher, J. Selbig, and H. G. Holzhutter. Kinetic hybrid models composed of mechanistic and simplified enzymatic rate laws - a promising method for speeding up the kinetic modelling of complex metabolic networks. *FEBS Journal*, 276(2):410–424, 2009.

[4] P. Cazzaniga, C. Daminani, D. Besozzi, R. Colombo, M. Nobile, D. Gaglio, D. Pescini, S. Molinari,

G. Mauri, L. Alberghina, and M. Vanoni. Computational strategies for a system-level understanding of metabolism. *Metabolites*, 4:1034–1087, 2014.

[5] W. Feller. *An Introduction to Probability Theory*, volume 1. Wiley, New York, NY, 1968.

[6] D. T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications*, 188(1-3):404–425, 1992.

[7] M. V. Heiden. Targeting cancer metabolism: a therapeutic window opens. *Nature Reviews. Drug discovery*, 10(9):671–684, 2011.

[8] D. M. Hendrickx, M. M. W. B. Hendriks, P. H. C. Eilers, A. K. Smilde, and H. C. J. Hoefsloot. Reverse engineering of metabolic networks, a critical assessment. *Molecular BioSystems*, 7:511–520, 2011.

[9] N. Jamshidi and B. O. Palsson. Mass action stoichiometric simulation models: Incorporating kinetics and regulatoin into stoichiometric models. *Biophysical Journal*, 98(2):175–185, 2010.

[10] S. a. Karline, c. Thomas, Petzoldt [aut, S. a. R. Woodrow, and odepack authors [cph]. Solvers for Initial Value Problems of Differential Equations (ODE, DAE, DDE). https://cran.r-project.org/web/packages/deSolve/deSolve.pdf, 2016.

[11] R.-S. W. L. Chen and X.-S. Zhang. *Biomolecular Networks: Methods and Applications in Systems Biology*. Wiley, New York, NY, 2009.

[12] K. Lange. *Optimization*. Springer, New York, NY, second edition, 2013.

[13] X. Liu and M. Niranjan. State and parameter estimation of the heat shock response system using kalman and particle filters. *Bioinformatics*, 28(11):1501–1507, 2012.

[14] U. E. Martinez-Outschoorn, M. Peiris-Pagès, R. G. Pestell, F. Sotgia, and M. P. Lisanti. Cancer metabolism: a therapeutic perspective. *Nature Reviews Clinical Oncology*, May 2016.

[15] C. G. Moles, P. Mendes, and J. R. Banga. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, 13(11):2467–2474, 2003.

[16] M. K. Molloy. Performance analysis using stochastic petri nets. *IEEE Transactions on Computers*, 31(9):913–917, 1982.

[17] B. C. Mulukutla, A. Yongky, P. Daoutidis, and W.-S. Hu. Bistability in glycolysis pathway as a physiological switch in energy metabolism. *PLoS ONE*, 9(6):1–12, June 2014.

[18] J. D. Orth, I. Thiele, and B. Ø. Palsson. What is flux balance analysis? *Nature biotechnology*, 28:245–248, March 2010.

[19] L. Popova-Zeugmann. *Time and Petri nets*. Springer, Heidelberg, GE, 2013.

[20] M. Radhakrishnan, J. S. Edwards, and F. J. Doyle. Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophysical Journal 83.3 (2002)*, 83(3):1331–1340, 2002.

[21] G. Sylvain, X. Yang, S. Brian, H. Julia, and P. SA. R Functions for Generalized Simulated Annealing. https://cran.r-project.org/web/packages/GenSA/GenSA.pdf, 2016.

[22] N. Totis, M. Beccuti, F. Cordero, L. Follia, C. Riganti, F. Novelli, and G. Balbo. Dealing with indetermination in biochemical networks. *In Proceedings of ACM Int. Workshop. of Italian Group on Quantitative Methods in Informatics (InfQ2016), Taormina, Italy, October 25 2016.*

[23] E. O. Voit. *Computational analysis of biochemical systems : a practical guide for biochemists and molecular biologists*. Cambridge University Press, Cambridge, New York, 2000.

[24] L. R. Yates and P. J. Campbell. Evolution of the cancer genome. *Nat Rev Genet*, 13:795–806, 2012.

[25] J. D. Young. Learning from the steersman: A natural history of cybernetic models. *Industrial & Engineering Chemistry Research*, 54(42):10162–10169, 2015.