



Towards Human Cell Simulation

Simone Spolaor^{1,5} , Marco Gribaudo² , Mauro Iacono³ , Tomas Kadavy⁴ ,
Zuzana Komínková Oplatková⁴ , Giancarlo Mauri^{1,5} , Sabri Pillana⁶ ,
Roman Senkerik⁴ , Natalija Stojanovic⁷, Esko Turunen⁸ , Adam Viktorin⁴ ,
Salvatore Vitabile⁹ , Aleš Zamuda¹⁰ , and Marco S. Nobile^{1,5}

¹ Department of Informatics, Systems and Communication,
University of Milano-Bicocca, Milan, Italy
nobile@disco.unimib.it

² Dipartimento di Elettronica, Informazione e Bioingegneria,
Politecnico di Milano, Milan, Italy

³ Dipartimento di Matematica e Fisica,
Università degli Studi della Campania “Luigi Vanvitelli”, Caserta, Italy

⁴ Faculty of Applied Informatics, Tomas Bata University in Zlin,
Zlin, Czech Republic

⁵ SYSBIO.IT Centre for Systems Biology, Milan, Italy

⁶ Department of Computer Science, Linnaeus University, Växjö, Sweden

⁷ Department of Computer Science, University of Niš, Niš, Serbia

⁸ Department of Mathematics, Tampere University of Technology, Tampere, Finland

⁹ Department of Biopathology and Medical Biotechnologies,
University of Palermo, Palermo, Italy

¹⁰ Faculty of Electrical Engineering and Computer Science,
University of Maribor, Maribor, Slovenia

Abstract. The faithful reproduction and accurate prediction of the phenotypes and emergent behaviors of complex cellular systems are among the most challenging goals in Systems Biology. Although mathematical models that describe the interactions among all biochemical processes in a cell are theoretically feasible, their simulation is generally hard because of a variety of reasons. For instance, many quantitative data (e.g., kinetic rates) are usually not available, a problem that hinders the execution of simulation algorithms as long as some parameter estimation methods are used. Though, even with a candidate parameterization, the simulation of mechanistic models could be challenging due to the extreme computational effort required. In this context, model reduction techniques and High-Performance Computing infrastructures could be leveraged to mitigate these issues. In addition, as cellular processes are characterized by multiple scales of temporal and spatial organization, novel hybrid simulators able to harmonize different modeling approaches (e.g., logic-based, constraint-based, continuous deterministic, discrete stochastic, spatial) should be designed. This chapter describes a putative unified approach to tackle these challenging tasks, hopefully paving the way to the definition of large-scale comprehensive models that aim at the comprehension of the cell behavior by means of computational tools.

Keywords: Agent-based simulation · Big data ·
 Biochemical simulation · Computational intelligence ·
 Constraint-based modeling · Fuzzy logic ·
 High-performance computing · Model reduction ·
 Multi-scale modeling · Parameter estimation ·
 Reaction-based modeling · Systems biology

1 Introduction

Cells are inherently complex systems, composed by a wide variety of molecule types, whose functioning is finely regulated by an intricate network of interactions. In order for cells to respond to environmental cues, surviving and reproducing, all of their components have to act together in a orchestrated manner. This wealth of complexity is the main reason for the richness of cellular behaviours that can be found in nature, but is also a major issue in advancing to a complete understanding of these systems.

In the last decades, mathematical modeling and simulation proved to be essential tools to understand and describe how biological functions emerge from the complex network of interactions existing between cellular components [139]. However, even though modeling and simulation proved successful in describing single processes or a limited amount of interacting pathways, extending this approach to define and simulate a whole-cell turned out to be an unfeasible task (besides the notable exception reported in [58], as it will be mentioned below), especially in the case of human cells. As a matter of fact, the definition and simulation of whole-cell models is challenging for several reasons. In particular, the problem is exacerbated by the complex organization of cell systems; the difficulties encountered in integrating different data sources and mathematical formalisms in a single modeling framework; the huge demand of computational power needed to perform the simulation. Although some of these challenges were already discussed and highlighted before (see for example [67]), we hereby provide a brief summary of the main challenges in the definition and simulation of whole-cell models:

- biomolecular systems are composed of a wide variety of heterogeneous components, ranging from small molecules, complex polymers (including proteins, sugars and ribonucleic acids) and protein complexes. All these components are further organized in functionally coherent pathways and organized in specialized compartments (e.g. the organelles in eukaryotic cells), ultimately giving rise to complex (observable) phenotypes;
- cells display a complex spatial and functional hierarchical organization, that results in phenomena occurring at a wide range of spatial and temporal scales [30]. Moreover, this organization often gives rise to complex non-linear dynamics;
- cellular systems are inherently stochastic, that is, the dynamics of cellular processes is characterized by biological noise [39], which is exploited by the cell to obtain specific responses that would be impossible in its absence [37].

Thus, some cellular pathways (e.g., gene expression) must be modeled and simulated as stochastic processes;

- the different nature of the cell components entails that they are measured with different experimental techniques. Some of these components can be measured with a high accuracy and with a high throughput (e.g., genomic or RNA sequencing, mass spectrometry), while others are very difficult or impossible to measure (e.g., kinetic information on the reaction rates). Thus, modelers have to take into account the presence of vast amounts of data, often in qualitative, or semi-quantitative form, together with limited quantitative information;
- the availability of multiple types of data, and the need to model different layers of organization, led to the definition of multiple modelling frameworks [118]. Because of this, models of biochemical systems are usually focused on one of the three main layers in which cellular processes are generally divided, namely: signalling (perceive environmental changes, process information and regulation of behaviour); gene regulation (control of expression levels of gene products); metabolism, i.e., the production and consumption, driven by enzymes, of small molecules essential for the life of cells. Even though attempts to define a single framework were made before [23], the integration of multiple modeling approaches is still challenging. However, a unified modeling framework for these three layers would provide a reliable means to capture their peculiarities [45], as was shown in [58].
- the availability of large amounts of experimental data, combined with the massive complexity of cells components, leads to huge computational requirements, even when considering the simulation of a single cell. Thus, dynamic mechanistic whole-cell models—encompassing all knowledge about biochemical reactions—are basically impossible to simulate on any existing computing architecture. However, we will see that by means of some assumptions about the system, such complexity can be mitigated using hybrid modeling, and model reduction techniques.

Considering all these challenges together, it comes to no surprise that, up to date, the only available example of whole-cell model is the one presented in the pioneering work of Karr *et al.* [58]. In this seminal work, the authors succeeded in simulating a whole-cell of one of the simplest known organisms, the *Mycoplasma genitalium*, adopting for each cellular process a suitable mathematical formalism. In particular, the authors showed the feasibility of predicting different cell phenotypes from a genotype, by relying on computational approaches. To the best of our knowledge, this results was not achieved again for any more complex organism. However, the integration of multiple formalism into a single modeling framework was already explored to smaller extents also in human cell models, for example in [41]. It is out of question that the simulation of whole-cell models will prove to be a challenge for modelers and computer scientists in the coming decades, and this is especially true in the case of human cells. Here, we propose a set of modeling approaches and techniques that would allow us to advance towards the simulation of human whole-cell models.

A dynamic whole-cell model would prove useful to understand how phenotypes emerge from the complex interactions existing between cellular components. Achieving dynamic simulation of a *human cell in silico* would have an even more considerable impact in the fields of molecular and systems biology, bioengineering and medicine [67]. Such a model, once validated, could allow to uncover new and potential unknown processes inside human cells, providing a reliable platform to generate new hypothesis to be tested in laboratory. In this regard, *in silico* tests would guide the experimental design, greatly reducing the costs, both in term of time and resources, of a “wet” laboratory. Moreover, human cell models could be exploited to automatically assess the effects of a vast number of perturbations in physiological or pathological conditions, in order to unveil potentially new drug targets or test known drugs in a high-throughput manner. We envision that human cell models could lead to breakthroughs in many fields of application, including medicine and personalized medicine, pharmacology and drug discovery, biotechnology and synthetic biology.

Regardless of the methodology used to create a whole-cell model, there are some aspects that will always characterize this kind of approach: High-Performance Computing (HPC) is necessary to mitigate the huge computational effort, in particular by distributing the computations over massively parallel machines and co-processors; dynamics modelling requires a proper kinetic parameterization to perform predictive simulations, and such parameters are often difficult—or even impossible—to measure by means of laboratory experiments, leading to a problem of parameter estimation; biological models are often characterized by multiple scales (temporal and spatial) which are not easy to handle; to reduce the huge computational effort due to large-scale models, both model reduction techniques or phenomenological simplifications can be leveraged. All these topics will be introduced and discussed in this paper.

This manuscript is organized as follows: in Sect. 2 we describe how HPC can mitigate the exceptional computational demand required by the simulation of whole-cell models; in Sect. 3 we propose modeling approaches for the definition of whole-cell models, while in Sect. 4 we suggest some techniques that could be employed to tackle the problems mentioned above in order to create a unified modeling approach; finally, in Sect. 5 we give some final remarks and highlight potential future directions.

2 High Performance Computing and Big Data

As it was highlighted in the previous section, High Performance Computing (HPC) architectures and handling of huge amounts of data will be necessary and enabling tools for the simulation of a human cell model. HPC involves the use of many interconnected processing elements to reduce the time to solution of given a problem. Many powerful HPC systems are *heterogeneous*, in the sense that they combine general-purpose CPUs with *accelerators* such as, Graphics Processing Units (GPUs), or Field Programmable Gates Arrays (FPGAs).

There exist several HPC approaches [11, 60, 89] developed to improve the performance of advanced and data intensive modeling and simulation applications.

Parallel computing paradigm may be used on multi-core CPUs, many-core processing units (such as, GPUs [77]), re-configurable hardware platforms (such as, FPGAs), or over distributed infrastructure (such as, cluster, Grid, or Cloud). While multi-core CPUs are suitable for general-purpose tasks, many-core processors (such as the Intel Xeon Phi [24] or GPU [85]) comprise a larger number of lower frequency cores and perform well on scalable applications (such as, DNA sequence analysis [71], biochemical simulation [53,76,81,123] or deep learning [129]).

Widely used parallel programming frameworks [70] for heterogeneous systems include OpenACC [138], OpenCL [115], OpenMP [88], and NVIDIA CUDA [84]. OpenMP is a set of compiler directives, library routines, and environment variables for programming shared-memory parallel computing systems. Furthermore, OpenMP has been extended to support programming of heterogeneous systems that contain CPUs and accelerators. OpenCL supports portable programming of hardware provided by various vendors, while CUDA runs only on NVIDIA hardware. CUDA C/C++ compiler, libraries, and run-time software enable programmers to develop and accelerate data-intensive applications on GPU.

As concerns distributed parallel computing, the available frameworks include the Message Passing Interface (MPI) [48], MapReduce/Hadoop [51] or Apache Spark [112]. MPI is a specification of library routines helpful for users that write portable message-passing programs in C/C++, Fortran or Python. Basic assumption behind MPI is that multiple processes work concurrently using messages to communicate and collaborate with each other. The MapReduce framework, and its open-source implementation Hadoop software stack, hides the details about data distribution, data availability and fault-tolerance, and allows to scale up to thousands of nodes inside cluster or Cloud computing systems. Lastly, Apache Spark [112] is a large-scale parallel computing platform that provides a wide variety of tools for structured data processing, including SQL queries (SparkSQL), streaming applications (Spark Streaming), machine learning (MLlib) and graph operations (GraphX), by means of various programming interfaces in Java, Scala, Python and R.

The data size in Bioinformatics, Computational Biology, and Systems Biology is increasing dramatically in the recent years. The European Bioinformatics Institute (EBI), one of the largest biology-data repositories, had approximately 40 petabytes of data about genes, proteins, and small molecules in 2014, in comparison to 18 petabytes in 2013 [56]. Big data problems in these fields are not only characterized by Velocity, Volume, Value, Variety, and Veracity, but also by incremental and geographically distributed data. While part of these data may be transferred over the Internet, the remaining are not transferable due to their size, cost, privacy, and other ethical issues [69]. Moreover, the computational time required by algorithms designed for the simulation of detailed mechanistic models (see Sect. 3.1) scales poorly when the models are characterized by a huge number of components. Thus, in recent years, research in Bioinformatics,

Computational Biology and Systems Biology started to adopt different HPC approaches to deal with Big Data.

In [86] Hadoop Blast (Basic Local Alignment Search Tool), in short HBLast, a parallelized BLAST algorithm is presented. HBLast exploits the MapReduce programming framework, adopting a hybrid “virtual partitioning” approach that automatically adjusts the database partition size depending on the Hadoop cluster size, as well as the number of input query sequences.

Sadasivam et al. considered in [100] a time efficient approach to multiple sequence alignment, as essential tool in molecular biology. They proposed a novel approach that combines the dynamic programming algorithm with the computational parallelism of Hadoop data grids to improve accuracy and to accelerate of multiple sequence alignment.

Li et al. developed in [65] ClustaWMPI, an accelerated version of ClustalW tool for aligning multiple protein or nucleotide sequences. In ClustaWMPI adopts MPI and runs on distributed workstation clusters as well as on traditional parallel computers.

The work presented in [15] describes a new Molecular Dynamics approach, named Desmond, that achieves unusually high parallel scalability and overall simulation throughput on commodity clusters by using new distributed-memory parallel algorithms. Desmond adopts a novel parallel decomposition method that greatly reduces the requirement for inter-processor communication, a novel message-passing technique that reduces the number of inter-processor messages, and novel highly efficient communication primitives that further reduce communication time.

The estimation of kinetic parameters, mandatory to perform cellular simulations, can be performed using population-based global optimization methods (see Sect. 4.2 for additional information). These algorithms are intrinsically parallel and can be accelerated using GPUs [78,79]. In [124] acceleration of the Differential Evolution algorithm is considered. In this work, a parallel implementation of an enhanced DE using Spark is proposed. Two different platforms have been used for the evaluation, a local cluster and the Microsoft Azure public cloud. The proposal drastically reduces the execution time, by means of including a selected local search and exploiting the available distributed resources. The performance of the proposal has been thoroughly assessed using challenging parameter estimation problems from the domain of computational systems biology. Additionally, it has been also compared with other parallel approaches, a MapReduce implementation and MPI implementation.

Coulier et al. presented in [29] a new framework, named *Orchestral*, for constructing and simulating high-fidelity models of multicellular systems from existing frameworks for single-cell simulation. They combined the many existing frameworks for single-cell resolution reaction-diffusion models with the diverse landscape of models of cell mechanics. They decoupled the simulation of reaction-diffusion kinetics inside the cells from the simulation of molecular cell-cell interactions occurring on the boundaries between cells. *Orchestral* provides a model for simulating the resulting model massively in parallel over a wide range of

distributed computing environments. They proved the flexibility and scalability of the framework by using the popular single-cell simulation software eGFRD to construct and simulate a multicellular model of Notch-Delta signaling over the OpenStack cloud infrastructure.

Finally, HPC is exploited to accelerate the simulation of biochemical models that are defined according to mechanistic formalisms [118] (refer also to Sect. 3.1 for some examples). In this context, GPUs [77] were already successfully employed to achieve a considerable reduction in the computational times required by the simulation of both deterministic [53, 76, 123] and stochastic models [81, 150]. Besides accelerating single simulations of such models, these methods prove to be particularly useful when there is a need of running multiple independent simulations of the same model. Hundreds (or even thousands) of simulations are often necessary to perform a wide variety of analysis on validated models (e.g., sensitivity analysis of kinetic parameters, or parameter sweep analysis), but also to perform parameter estimation (PE) during the definition of such models (please, refer to Sect. 4.2 for an extensive description). This kind of tasks leverages at most the availability of the many cores of the GPUs, greatly reducing the overall running time that is required to perform them [82, 83].

3 Modeling Approach

In the field of Systems Biology, several modeling approaches have been defined [114, 118]. Each approach exploits a different mathematical formalism and was developed to address the challenges posed by a specific (subset of) biochemical processes (e.g. metabolism [117], gene regulation, or signaling). The definition of a single, homogeneous mathematical framework to model and simulate a whole-cell seems currently unfeasible, while the integration of multiple formalisms has already proved to be able to achieve outstanding results [58]. Following this principle, we decided to define our human cell modeling framework by integrating multiple modeling approaches, namely: *(i)* mechanism-based models (in particular reaction-based and agent-based models); *(ii)* constraint-based models; *(iii)* logic-based models (in particular boolean and fuzzy logic-based models). These approaches, together with their peculiarities and limitations, will be briefly described in the following subsections.

3.1 Reaction-Based Modeling

Biochemical systems are traditionally formalized as mechanistic and fully parameterized reaction-based models (RBMs) [12]. A RBM is defined by specifying the following sets:

- the set $\mathcal{S} = \{S_1, \dots, S_N\}$ of molecular species;
- the set $\mathcal{R} = \{R_1, \dots, R_M\}$ of biochemical reactions that describe the interactions among the species in \mathcal{S} ;
- the set $\mathcal{K} = \{k_1, \dots, k_M\}$ of kinetic constants associated with the reactions in \mathcal{R} ;

- the set of the initial concentration $Y_i \in \mathbb{R}_0^+$, with $i = 1, \dots, N$, for each species $S_i \in \mathcal{S}$.

Any RBM can be represented in a compact matrix-vector form $\mathbf{A}\mathbf{S} \xrightarrow{\mathbf{K}} \mathbf{B}\mathbf{S}$, where $\mathbf{S} = (S_1, \dots, S_N)^\top$, $\mathbf{K} = (k_1, \dots, k_M)^\top$, and $\mathbf{A}, \mathbf{B} \in \mathbb{N}^{M \times N}$ are the stoichiometric matrices whose elements $[A]_{i,j}$ and $[B]_{i,j}$ represent the number of reactants and products occurring in the reactions, respectively. Given an RBM and assuming the law of mass-action [22], the system of coupled Ordinary Differential Equations (ODEs) describing the variation in time of the species concentrations is obtained as follows:

$$\frac{d\mathbf{Y}}{dt} = (\mathbf{B} - \mathbf{A})^\top [\mathbf{K} \odot \mathbf{Y}^{\mathbf{A}}], \quad (1)$$

where $\mathbf{Y} = (Y_1, \dots, Y_N)$ represents the state of the system at time t , $\mathbf{Y}^{\mathbf{A}}$ denotes the vector-matrix exponentiation form [22], while the symbol \odot denotes the Hadamard product. The system can then be simulated using a numerical method, which is usually based on implicit integration (e.g., Backward Differentiation Formulae [19]) due to the stiffness that characterizes these models.

When the chemical species have a low concentration, the dynamics of the system becomes intrinsically stochastic and the biochemical system should be simulated using specific approaches like Gillespie's Stochastic Simulation Algorithm (SSA) [43]. In SSA, the simulation proceeds one reaction at a time. Both the reaction to be fired and the time interval τ before the reactions occur are determined in a probabilistic fashion. Thus, the simulated trajectory of the system can radically diverge from the one predicted by a deterministic simulation, allowing the investigation of the emergent effects due to the intrinsic noise and providing a deeper knowledge of the system's behavior. In the case of stochastic modeling, the state of the system represents the exact number of molecules; \mathbf{K} denotes the vector of the stochastic constants, encompassing all the physical and chemical properties of the reactions. These parameters are used to calculate the propensity functions, ultimately determining the probability of each reaction R_m to occur. Propensity functions are defined as:

$$a_m(\mathbf{Y}) = k_m \cdot d_m(\mathbf{Y}), \quad (2)$$

where $d_m(\mathbf{Y})$ is the number of distinct combinations of reactant molecules occurring in R_m . The delay time τ before the next reaction will occur is calculated according to the following equation:

$$\tau = \frac{1}{a_0(\mathbf{Y})} \cdot \ln \frac{1}{rnd}, \quad (3)$$

where $a_0(\mathbf{Y}) = \sum_{m=1}^M a_m(\mathbf{Y})$ and rnd is random number sampled with uniform distribution in $[0, 1)$.

Mechanistic modeling is considered the most likely candidate to achieve a detailed comprehension of biological systems [20], since it can lead to quantitative predictions of cellular dynamics, thanks to its capability to reproduce the

temporal evolution of all molecular species occurring in the model. Nonetheless, the computational complexity of the simulation and analysis of such models increases with the size (in terms of components and interactions) of the systems, limiting the feasibility of this approach. Moreover the usual lack of quantitative parameters (e.g., kinetic constants, initial molecular concentrations of the species) and the partial lack of knowledge about the molecular mechanisms, sometimes due to the difficulty or impossibility to perform *ad hoc* experiments, represent further limits to a wide applicability of this modeling approach. The problems of simulation performances and parameter estimation are discussed in the next Sections.

3.2 Constraint-Based Modeling

Constraint-Based Modeling (CBM) is based on the idea that phenotypes of a given biological system must satisfy a number of constraints. Hence, by restricting the space of all possible systems states, it is possible to determine the functional states that a biochemical (in particular, metabolic) network can or cannot achieve. The fundamental assumption of constraint-based modeling is that the organism will reach a quasi-steady state that satisfies the given constraints [20].

The starting point of CBM is the transposed stoichiometric matrix $\mathbf{S} = (\mathbf{B} - \mathbf{A})^T$, i.e., a matrix in which each row corresponds to a chemical species (e.g., metabolites), while columns correspond to reactions involving those species. Since metabolic networks typically include more reactions (“fluxes”) than metabolites, the stoichiometric constraints and the steady assumption alone lead to an under-determined system in which a bounded solution space of all feasible flux distributions can be identified. Additional constraints should be incorporated to further restrict the solution space; this is usually performed by specifying linear bounds to minimum and maximum values of fluxes. Additional capacity constraints are generally set according to experimental data.

On top of CBM, Flux Balance Analysis (FBA) can be used to identify optimal distribution of fluxes with respect to a given objective function. Thanks to the linear definitions of fluxes, constraints and objective function, the solution space is a multi-dimensional convex polytope. FBA exploits a simplex method to efficiently identify the optimal fluxes that maximize, or minimize, the objective function (e.g., the maximization of ATP [128] in the context of mitochondria energy metabolism). CBM methods do not perform an actual simulation of the biochemical system, but can be used—under a quasi-steady state assumption—to investigate the distribution of fluxes. Interestingly, FBA has a very limited computational complexity, so that it can be leveraged to study the behavior of a metabolic systems on a whole-cell level.

3.3 Markovian Agents

Markovian agents [13] are a modeling tool that is specially suitable for large scale phenomena composed of groups of single entities that behave as Markov chains. Such entities, said *agents*, are individuals belonging to *classes* that are

characterized by a common description of their dynamics. Agents may influence each other by means of a technique called *induction*, which accounts for their position in a logic map that represents the space in which they can move or be positioned in the system. The system is described by considering for each class the density of agents in each state and the probability of transition between states, so that, thanks to a *mean-field* approach, the evolution in time of the density in states may be approximately described by differential equations and a closed form solution may be obtained, with the significant advantage that the higher is the number of agents in a class, the best the approximation describes the system. The communication mechanism acts by enabling or disabling transitions, thus influencing the probability of transitions between states. This analytical description is suitable to study both transient and regime behavior of the system.

Markovian agents may be used to describe the interactions of reactions that happen in a cell in a large number of independent instances, including the effects of inhibiting factors, as well as for describing the expression of cells in tissues and organs. The technique has been applied to study biological pathways [27], cancer cells [28], whole ecosystems, such as forestry landscape [142], and other complex real-world systems [7, 21, 47]. The Markovian property make them suitable to describe processes that are characterized by exponentially distributed interarrival time in their evolution.

From the formal point of view, let a Markovian agents model be composed by different classes, with each class c characterized by a Markov chain with n_c states: the space Σ in which agents are located and can move is finite and can be continuous or discrete. The distribution of agents in the space can be represented by a density function $\delta : \Sigma \rightarrow \mathbb{R}^+$ so that, considering any subspace $U \subset \Sigma$, the number of agents in U is described by a Poisson distribution with mean $\int \int_U \delta(x) dx$. The model evolves by accounting for state changes of agents in their class and induction effects, birth of agents and death of agents: its evaluation can be obtained as a counting process per each class that counts the number of agents in each state of its Markov chain, in each position in space and in each instant.

Let $\chi_c(l, t) = |\chi_i^{[c]}(l, t)|$ be a vector of size $n^{[c]}$, with each element $\chi_i^{[c]}(l, t)$ representing the average number of agents of class c in state i at time t and in location l . If the space is discrete, the evolution of the counting process is thus described by a set of ordinary differential Equations 4 for each class c and in location l :

$$\frac{d\chi_c(l, t)}{dt} = b_c([\chi], l, t) + \chi_c(l, t) \cdot K_c([\chi], l, t) \quad (4)$$

where $[\chi]$ denotes the dependency on all the state of all agents in the model in any time instant, matrix K_c is the main transition kernel that accounts for spontaneous and induced actions contribution and b_c is the birth vector of new agents for the class in a state.

If the space is continuous, movement of agents is described by a diagonal velocity matrix ω_c , described in Eq. 5, that can be obtained by summing the contributions for each direction:

$$\frac{\partial(\omega_c([\chi], l, t) \cdot \chi_c(l, t))}{\partial l} = \frac{\partial(\omega_{xc}([\chi], l, t) \cdot \chi_c(l, t))}{\partial l_x} + \frac{\partial(\omega_{yc}([\chi], l, t) \cdot \chi_c(l, t))}{\partial l_y} + \dots \quad (5)$$

and Eq. 4 is modified accordingly and becomes Eq. 6:

$$\frac{\partial \chi_c(l, t)}{\partial t} + \frac{\partial(\omega_c([\chi], l, t) \cdot \chi_c(l, t))}{\partial l} = b_c([\chi], l, t) + \chi_c(l, t) \cdot K_c([\chi], l, t) \quad (6)$$

in which the second term accounts for the effects of agents movement by v_c .

3.4 Logic-Based Modeling

In contrast with mechanism- and constraint-based models, logic-based model do not require kinetic or stoichiometric information to be defined. Although these models can describe the system under consideration only in qualitative terms, they provide an efficient way to simulate the dynamic evolution of complex systems, even when precise kinetic information is not available. Thanks to their closeness to human language, logic-based models are able to leverage qualitative and semi-quantitative data and they are generally regarded as more interpretable by human experts. Moreover, their flexibility allow modelers to represent in the same model highly heterogeneous components and the interactions existing among them.

Logic-based models are defined by a set of v variables \mathcal{V} and a set of ϕ IF-THEN logic rules \mathcal{F} , describing the interactions existing between the components. Evaluation of the rules in discrete time steps drives the system's dynamics: this can be achieved by either a synchronous (deterministic) or asynchronous (stochastic) update policy [141]. Logic-based models are commonly employed in systems biology to model gene regulatory networks and signal processing [74]. Among them, Boolean models are the most simple and widely used: in this kind of models, variables can assume only two discrete states, often represented as 0 and 1, active or inactive, present or not present. Different Boolean logic models were successful in predicting cellular behaviours [141], however these assumptions often limit their ability of representing biomolecular processes.

In order to overcome these limitations, more recently fuzzy logic was proposed as an alternative to the modeling of complex biochemical systems [3]. Fuzzy logic is a powerful, multi-valued extension of boolean logic, which allows variables to assume multiple states in a continuous manner (i.e., between [0,1]) and deal with any uncertainty related to the system. More in particular, fuzzy IF-THEN inference systems are composed of ϕ rules of type:

IF v_1 is in $V_{1,1}$ and v_2 is in $V_{1,2}$ and ... and v_v is in $V_{1,v}$ THEN o is in O_1
 IF v_1 is in $V_{2,1}$ and v_2 is in $V_{2,2}$ and ... and v_v is in $V_{2,v}$ THEN o is in O_2

...

IF v_1 is in $V_{\sigma,1}$ and v_2 is in $V_{\sigma,2}$ and ... and v_v is in $V_{\sigma,v}$ THEN o is in O_σ ,

where $v_j, o \in \mathcal{V}$, with $i = 1, \dots, v$, while the sets $V_{i,j}$ and O_i , with $i = 1, \dots, \sigma$, and $j = 1, \dots, v$ are *fuzzy (sub-)sets*, that is, the membership of the value assumed by a generic variable $v \in \mathcal{V}$ for the fuzzy subset V is equal to a degree $\alpha \in [0, 1]$. This is denoted by $\mu_V(v) = \alpha$. If all the considered sets are classical sets (i.e. always holds $\mu_V(v) \in \{0, 1\}$), then the inference system is boolean.

An advantage of fuzzy reasoning is that, thanks to the fuzzy sets, it can handle uncertainty and conflicting conclusions drawn from the logic rules [126]. Thus, it can allow for the dynamic simulation of qualitative and semiquantitative models, even when precise kinetic information is missing. Fuzzy logic has been applied to vastly different fields of research, ranging from automatic control [36] to medicine [99], but it was successfully applied also in the field of cellular biology, for example, to model signaling pathways [3] and gene regulatory networks [63].

We plan to exploit fuzzy logic in our hybrid framework to overcome the lack of kinetic parameters [14, 66] and model those cellular processes that still are not understood in mechanistic detail, or whose components cannot be represented by crisp, real-valued variables (e.g., complex phenotype as apoptosis/survival, microscopy imaging data, etc.).

4 A Unified Modeling Approach

In principle, the SSA algorithm described in Sect. 3.1 can be used to simulate a stochastic trajectory of any biological model, including a whole-cell model, and such dynamics would be exact with respect to the Chemical Master Equation (CME) underlying the corresponding set of biochemical reactions. This approach could be even extended to consider the diffusion of molecules inside the cell, like in the case of the Next Subvolume Method (NSM) [38]). However, both SSA and NSM perform the simulations by applying a single reaction at a time, proceeding with time steps that are inversely proportional to the sum of the propensities (see Eq. 3) which, in turn, is proportional to the amount of reactants in the system. These circumstances generally cause an explosion of the computational effort due to exact stochastic simulation, making it unfeasible for whole-cell simulation.

An approximate but faster version of SSA, called tau-leaping [44], was proposed by Gillespie to reduce the computational burden typical of SSA: by assuming that the propensities do not change during a given time-interval (the so-called *leap condition*) the number of reactions firing can be approximated by Poisson random variables.

When the number of estimated reaction firings for all reactions increases, the Poisson processes can be approximated by a normal distribution with same mean and variance [44]. In this case, Stochastic Differential Equations (SDEs) like the Langevin equations can be exploited to model the system, which is then simulated using numeric solvers like the Euler-Maruyama method [119], strongly reducing the overall computational effort. Finally, when the propensities become extremely large, the noise term in the SDEs becomes negligible and can be removed, so that the system can be modeled using simple ODEs [44].

The proper modeling approach must be carefully selected according to the characteristics of the chemical system. Unfortunately, cellular mechanisms are

controlled by reactions and pathways spanning over multiple scales, so that none of these modeling methods is really adequate. By partitioning the reactions set \mathcal{R} into multiple regimes, according to their characteristics (e.g., their propensity values), it is possible to simulate each subsystem using the optimal modeling approach. It is clear that the firing of reactions in one regime can have a huge impact to the others, so that the synchronization phase—necessary to propagate the information across the regimes—becomes a mandatory and very delicate phases of multi-scale hybrid simulators, like in the case of the Partitioned Leaping Algorithm (PLA) [52].

By extending PLA by considering the additional modeling approaches described in Sect. 3.1, it is possible to achieve whole-cell models [58]. In this project, we pursue the integration of these modeling approaches, pushing the limits of human cells simulation. In order to mitigate the huge computational requirements, we plan to exploit model reduction and automatic simplification algorithms. We also plan to perform an automatic inference of some missing parts of the model (e.g., reactions, rules, parameters), exploiting state-of-the-art evolutionary and statistical methods. Finally, we will test multi-agent approaches to work on multiple scales (e.g., multiple cells or tissue simulation). All these approaches will be described in the next subsections.

4.1 Model Reduction and Simplification

The complexity of cellular systems poses some limitations on the scale of the models that can be simulated. In this context, model reduction techniques can be used to tame the complexity before the execution of simulation algorithms is performed.

The theory of complex networks has raised a great development over the recent years. The empirical and theoretical results analyzing several real systems show that complex networks can be classified using its probability distribution function $P(k)$, i.e. the probability that a node is connected to k nodes of a network. A scale-free network has the grades distribution function fitting the power-law function [57]. Several studies examining the cellular metabolism of different organisms have been conducted for determining the topological structure of a metabolic network [57]. In this direction, studies of Barabási and Albert have also analyzed many issues in scale-free networks [2].

In many organism, the metabolic networks are composed of interconnected functional modules and follow the scale-free model [49, 61]. Three statistical measures can be considered in a scale-free network: the connectivity degree, the diameter of the graph, and the clustering coefficient [2]. The connectivity degree of a node is the number of incident arcs and it allows also for calculating the distribution function of the connectivity degree. The diameter provides an estimation of the average number of hops between any pair of nodes in the network. It is also linked to the shortest paths between each node pair as well as to the number of paths in the network. Finally, the clustering coefficient gives a measure of the properties of nodes to form agglomerates. In addition, metabolic network nodes can be classified into distinct groups considering the following parameters

[49]: the within-module degree, i.e., the membership degree of a node into its functional module, and the participation coefficient, i.e., a measure of the node interaction with network functional modules. The above parameters can be used to define non-hub and hub nodes as well as peripheral, provincial, connector, and kinless nodes [49,64]. These metrics pave the way to the topological analysis of a network, providing information on the connectivity and the participation degrees of each node within the network.

The topological analysis of a network can be completed by functional analysis. A cellular network is hierarchically organized with several functional modules [5,57]. Methods for a rational decomposition of the network into independent functional subsets are essential to understand their modularity and organization principles.

Using the modularization approach commonly used in the area of control theory, a cellular network can be viewed as an assembly of basic building blocks with its specific structures, characteristics, and interactions [103,135]. Modularization reduces the difficulty in investigating a complex network. Network decomposition is also needed for cellular functional analysis through pathway analysis methods that are often troubled by the problem of combinatorial explosion due to the complexity of those networks.

Two main methods can be used for network functional analysis and, as a consequence, for network model reduction and simplification: Flux Balance Analysis (FBA) and Extreme Pathways Analysis (ExPA) [59,103,137].

FBA is a mathematical technique based on fundamental physical and chemical laws that quantitatively describe the metabolisms of living cells. FBA is a constraint-based modeling approach [96]: it assumes that an organism reaches a steady-state (under any given environmental condition) that satisfies the physicochemical constraints and uses the mass and energy balance to describe the potential cellular behavior. FBA model has been developed considering the mass and energy conservation law: for each node/metabolite, the sum of incoming fluxes must be equal to the sum of the outgoing ones. The space of all feasible solutions of a linear equation constrained system lies within a three-dimensional convex polyhedron, in which each point of this space satisfies the constraints of the system [96]. When the system has an optimal and limited solution, this is unique and it is located on a polyhedron vertex. However, the system can have multiple optimal solutions (axis or plan) that are used to detect network redundancies [96].

ExPA analysis detects the vital pathways in a network. They are the unique set of vectors that completely characterize the steady-state capabilities of a network. A network steady-state operation is constrained to the region within a cone, defined as the feasible set. In some special cases, under certain constraints, this feasible set collapse in a single point inside the cone. The algorithm detects the extreme rays/generating vectors of convex polyhedral cones. Algorithm time execution is proportional to the number of nodes and pathways [137].

Many software frameworks for cellular networks analysis and simulation have been developed. Some solutions, such as Pajek [34], allows for either large

complex networks analysis and visualization or network structural properties and quantities analysis. CellNetAnalyzer [62] is a MATLAB package for performing biochemical networks functional and structural analysis.

The BIAM framework implements an integrated analysis methodology based on topological analysis, FBA analysis, and Extreme Pathways analysis [26, 134]. The framework supplies the needed tools for drawing a network and analyzing its structural and functional properties. Several scale-free network architectures, dealing with different application domains, have been simulated and validated [26, 136]. Topological and functional analysis can be combined to select the main functional nodes and paths of a cellular network. Redundant nodes and non-vital paths could be ignored before the execution of time-constrained simulation algorithms, reducing the overall computational complexity of large scale simulation.

4.2 Parameter Estimation

Mechanistic models are characterized by a kinetic parameterization (i.e., the \mathbf{K} vector described in Sect. 3.1). A precise estimation of such parameters is mandatory to perform faithful simulations of the system's dynamics. The problem of Parameter Estimation (PE) can be formulated as a minimization problem: the goal is to reduce to zero a distance between the target experimental discrete-time time-series and a simulated dynamics performed with the optimal vector of parameters [83]. Due to the characteristics of the fitness landscapes defined by the PE problem (i.e., multi-modal, non-linear, non-convex, noisy), classic optimization methods cannot be employed efficiently. On the contrary, Computational Intelligence (CI) methods based on evolutionary computation or swarm intelligence were shown to be effective for this problem [35, 83], in particular the settings-free variant of PSO named Fuzzy Self-Tuning PSO [80]. Moreover, CI methods can be combined with probabilistic frameworks (e.g. expectation-maximization methods [55]) to efficiently tackle the PE of stochastic models (see for example [95]). However, when the number of missing parameters in the model becomes extremely large, like in the case of whole-cell models, conventional CI methods can show some limitations and large-scale methods must be employed.

Among the existing CI algorithms for large number of parameters, Differential Evolution (DE) [116] variants like the recent DISH [132] algorithm could be exploited. DE algorithm was introduced in 1995 by Storn and Price [116] and since then formed a basis for a set of successful algorithms for optimization domains, such as continuous, discrete, mixed-integer, or other search spaces and features [146]. The whole encompassing research field around DE was surveyed most recently in [32] and even since then, several other domain- and feature-specific surveys, studies, and comparisons have also followed [1, 90, 92, 93]. Theoretical insight and insights to inner workings and behaviors of DE during consecutive generations has been studied in the works like [87, 122, 133, 144].

As the continuing research in DE enhancements and insight supports a much vigorous research community, the DE algorithm variants have also steadily placed top in competitions held annually at Congress on Evolutionary Computation (CEC) [16, 17, 31, 68, 73, 97, 98, 140]. For this reason, we expect these

advanced versions of DE to be effective for the PE problem and outperform classic algorithms, especially on high dimensional problem.

The most recent variants' strain of DE is the Success-History based Adaptive Differential Evolution (SHADE) [120], which has a line of recent improvements following a taxonomy [1] stemming from JADE [149] that is based on jDE [16, 144], upgraded as L-SHADE [121], SPS-L-SHADE-EIG [50], LSHADE-cnEpSin [4], jSO [18], aL-SHADE [91], and most recently, DISH [132]. These algorithms include different mechanisms and to describe the basic outline working principle to apply DE, from the following paragraph on, the basic canonical 1995 DE is described.

The canonical 1995 DE is based on parameter estimation through evolution from a randomly generated set of solutions using population \mathbf{P} , which has a preset size of NP . Each individual (a set of estimated parameter values) in this population \mathbf{P} consists of a vector \mathbf{x} with a of length D . Each vector \mathbf{x} component corresponds to one attribute of the optimized task for which parameters are being estimated. The objective function value $f(\mathbf{x})$ evaluates quality of the solution. The individuals in the population create improved offspring for the next generation. This process is repeated until the stopping criterion is met (either the maximum number of generations, or the maximum number of objective function evaluations, or the population diversity lower limit, or overall computational time), creating a chain of subsequent generations, where each following generation consists of eventually better solutions than those in previous generations.

Some of most used computational operators operating on population \mathbf{P} over each generation and its vectors, are parameter adaptation, mutation [149], crossover [121], selection [132], and population restructuring including adaptation of population size [144]. First, all vectors in the initial population are uniformly generated at random between bounds $[x_{\text{lower},j}, x_{\text{upper},j}]$, $\forall j = 1, \dots, D$:

$$\mathbf{x}_i = \{\mathcal{U}[x_{\text{lower},j}, x_{\text{upper},j}]\}; \forall j = 1, \dots, D; \forall i = 1, \dots, NP, \quad (7)$$

then, three mutually and from current vector index i different, indices r_1 , r_2 , and r_3 , are used to computing a differential vector (hence the name DE for algorithm) and combine it in a scaled difference manner:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}), \quad (8)$$

which is then taken into crossover with the current vector at index i :

$$u_{j,i} = \begin{cases} v_{j,i} & \text{if } \mathcal{U}[0, 1] \leq CR_i \text{ or } j = j_{\text{rand}} \\ x_{j,i} & \text{otherwise} \end{cases}, \quad (9)$$

finally through selection yielding a new vector $\mathbf{x}_{i,G+1}$ at this location i for next generation $G + 1$:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G} & \text{if } f(\mathbf{u}_{i,G}) \leq f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G} & \text{otherwise} \end{cases}. \quad (10)$$

As mentioned in the beginning of this subsection, the work on DE is ongoing and still challenging. To apply DE most efficiently on a new challenge for

parameter estimation like the discussed simulation in this chapter, one of effective DE variants should be taken and adapted for the domain challenge at hand, following recent experiences on DE applications in e.g. image processing [143], energy scheduling [145], and autonomous vehicle navigation [147, 148].

To assess the feasibility of DISH for the large-scale PE problem, we plan to compare its performances against state-of-the-art methods, in particular the aforementioned variants of DE and those algorithms that were shown effective for the PE in previous studies (i.e., PSO [35] and FST-PSO [83]).

Another approach for DE that may be beneficial for the given application is through unconventional synergy of the DE with several different research fields belonging to the computational intelligence paradigm, which are the stochastic processes, complex chaotic dynamics, and complex networks (CN).

As the key operation in metaheuristic algorithms is the randomness, the popularity of hybridizing them with deterministic chaos is growing every year, due to its unique features. Recent research in chaotic approach for metaheuristics mostly uses straightforwardly various chaotic maps in the place of pseudo-random number generators. The observed performance of enhanced optimizer is (significantly) different, mostly the chaotic maps secured very fast progress towards function extreme, but often followed by premature convergence, thus overall statistics has given mixed results. Nevertheless, as reported in [106], the the chaos driven heuristics is performing very well [104, 107], especially for some instances in the discrete domain [33, 72].

The CN approach is utilized to show the linkage between different individuals in the population. Interactions in a swarm/evolutionary algorithms during the optimization process can be considered like user interactions in social networks or just people in society. The population is visualized as an evolving CN that exhibits non-trivial features - e.g., degree distribution, clustering, and centralities. These features can be then utilized for the adaptive population control as well as parameter control during the metaheuristic run. Analysis of CNs from DE algorithm can be found in [105, 108, 109, 130, 131]; and also in a comprehensive study discussing the usability of network types [110].

4.3 Automatic Inference of Fuzzy Rules

Fuzzy IF-THEN inference systems are typically constructed by consulting human experts, who give the related fuzzy rules, shapes of the corresponding fuzzy sets and all the other required information. However, when human experts are not available or in the presence of numerous system components and/or rules, the definition of the inference system results to be particularly time consuming and laborious. An alternative approach is exploiting data mining methods, in order to automatically build inference systems by leveraging available data.

In particular, here we focus on GUHA (General Unary Hypotheses Automaton), a method of automatic generation of hypotheses based on empirical data. GUHA is based on a particular first order logic language, which allows to treat symbolically sentences such as α appears often simultaneously with β , in most cases α implies β , α makes β very probable, etc. The GUHA method

is implemented in the LISpMiner software [127], which is freely downloadable from <https://lispminer.vse.cz/>. Once the user provides relevant *analytic questions* regarding the data, the LISpMiner software outputs the dependencies between the variables that are supported by the data. In practice, LISpMiner runs through millions of fourfold contingency tables, from which it outputs those which support the dependence provided by the user. From these findings, the IF-THEN inference system can then be constructed.

GUHA and LISpMiner were already successfully employed in different fields [127]: in the context of human cell modeling, this approach could be exploited in order to automatically build large fuzzy inference systems. In particular, this data mining method could leverage the vast availability of transcriptomic data [54], which nowadays can be generated in short time, for a reasonable cost and at single-cell resolution [113]. In such a way, we envision that the automatic generation of large-scale dynamic fuzzy models of cellular processes would be feasible. Such models would represent a significant step forward towards the integration of cellular processes that are not known in full mechanistic detail, or necessitate of a qualitative or semi-quantitative representation, inside a unified framework for human cell modelling and simulation.

4.4 Multiformalism Approaches

Given the complexity and the heterogeneity of the sub-problems that characterize the challenge posed by whole-cell modeling, a promising approach can be provided by *multiformalism modeling* [46]. Multiformalism modeling offers the possibility of obtaining complex models by allowing the coexistence of different modeling formalisms in the same model, using model composition, model generation, model abstraction on the basis of different supporting mechanisms. Multiformalism approaches allow the representation of each subsystem with the most appropriate representation, or with the description that is more familiar for the developer of that submodel, easing the interaction between experts from different domains without forcing any of them to relinquish established modeling practices: this allows to preserve existing know how and minimizes the effort needed to integrate the overall model, that is a process that is supported by a proper specialist in formalism design. Multiformalism models may be supported by closed frameworks [25, 102, 125], that support a predefined set of formalisms, or by open frameworks [6, 42], that are designed to allow the definition of new formalisms.

The solution, or analysis, of multiformalism models may be performed by generating a specific solvable model, by generating or instantiating a simulation tool, by orchestrating specific solvers for different submodels, by producing executable code. Solution can be obtained by means of simulation, analytical techniques or by applying *multisolution*, that is the possibility of using alternate tools, explicitly decided by the modeler or automatically chosen according to the characteristics of the model, to perform the analysis. This approach also preserves, in general, tracking of numerical results back to logical elements in the model, and can provide model-wide or submodel-wide results, such as properties

of parts of the system that emerge from element-related results, and may also be used to interface existing tools with new solvers, extending their applicability [10]. Multiformalism modeling approaches may support combinatorial formalisms [125], logic modeling [25], discrete state space based formalisms [6, 42], continuous state space based formalisms [6], and hybrid formalisms [8] (that may use specialized solution techniques [9]). More details about multiformalism modeling concepts and principles are available for the reader in [46] and [101]. For a similar and wider concept, namely *multiparadigm modeling*, the reader can refer to [75].

5 Future Developments

In this chapter we described a putative hybrid modeling and simulation framework—exploiting several different approaches (e.g., RMBs, CBMs, boolean and fuzzy rules) and leveraging High-Performance Computing—designed to perform large-scale cell simulations. In this context, we highlighted some issues that prevent the simulation of whole-cell models, proposing some approaches in order to achieve this challenging task.

In particular, we propose the use of population-based metaheuristics for global optimization to estimate the large number of missing kinetic parameters. The emphasis in future research will be on modifying and testing robust algorithms based on DE/DISH inspired by techniques successfully adopted for solving highly constrained, large-scale and multi-objective problems. We will compare this class of algorithms against swarm intelligence techniques (e.g., PSO [94] and FST-PSO [80]) that were shown to be the most effective in previous empirical studies [35, 83].

Furthermore, a thorough analysis of the relatively good results of genetic algorithms can help to develop powerful metaheuristics. Moreover, it is necessary to emphasize the fact that, like most of the above mentioned metaheuristic methods, they are inspired by natural evolution, and their development can be considered as a form of evolution. Such a fact is mentioned in the paper [93] that even incremental steps in algorithm development, including failures, may be the inspiration for the development of robust and powerful metaheuristics. Future directions in DE can be discussed not only in the journals like *Swarm and Evolutionary Computation*, *IEEE Transactions on Evolutionary Computation*, or *Evolutionary Computation*, but also at forthcoming conferences like *Swarm, Evolutionary and Memetic Computing Conference (SEMCCO)*, *IEEE Congress on Evolutionary Computation (CEC)*, and *The Genetic and Evolutionary Computation Conference (GECCO)*, all forthcoming also for year 2019.

A lot of work still needs to be done, in order to achieve a faithful representation of a human cell *in silico*. The unified approach that we propose in this work, although challenging to achieve and possibly able to capture a wide variety of cellular behaviors, must be considered just as a starting point. As a matter of fact, many additional layers of complexity can still be considered. We assume that the biochemical systems is well-stirred, but this is often not

the case. Spatial modeling and simulation can be leveraged to capture the organization in space of molecules (e.g., membrane receptors), cell organelles and cell shape itself. The combinatorial complexity of the formation of huge protein complexes or bio-polymers can also be tackled by means of specific modeling [40] and simulation frameworks [111]. Moreover, cells are not closed systems: they respond to environmental cues and they continuously interact with other cells by exchanging chemical signals. Furthermore, cell's life cycle is coordinated by a complex cell cycle program, that allows them to grow and divide, and they are constantly subjected to the evolutionary pressure posed by the environment. External signals and cell cycle both require additional complex modeling approaches that are currently not considered in our approach. Whilst we envision that human cell simulation will remain a challenging task for the coming decades, we are working in that direction as it carries the promise of elucidating the very basic mechanisms governing the functioning of our bodies and life itself.

Acknowledgements. This chapter is based upon work from the COST Action IC1406 cHiPSet, supported by COST (European Cooperation in Science and Technology). The author AZ acknowledges the financial support from the Slovenian Research Agency (research core funding No. P2-0041). AZ also acknowledges EU support under project no. 5442-24/2017/6 (HPC – RIVR). This chapter is also based upon work from COST Action CA15140 “Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO)” supported by COST. The authors RS and ZKO also acknowledges that work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme Project No. LO1303 (MSMT-7778/2014), further supported by the European Regional Development Fund under the Project CEBIA-Tech no. CZ.1.05/2.1.00/03.0089. Further authors TK and AV acknowledges the support of Internal Grant Agency of Tomas Bata University under the Projects no. IGA/CebiaTech/2019/002.

References

1. Al-Dabbagh, R.D., Neri, F., Idris, N., Baba, M.S.: Algorithmic design issues in adaptive differential evolution schemes: review and taxonomy. *Swarm Evol. Comput.* (2018). <https://doi.org/10.1016/j.swevo.2018.03.008>
2. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**(1), 47 (2002)
3. Aldridge, B.B., Saez-Rodriguez, J., Muhlich, J.L., Sorger, P.K., Lauffenburger, D.A.: Fuzzy logic analysis of kinase pathway crosstalk in TNF/EGF/insulin-induced signaling. *PLoS Comput. Biol.* **5**(4), e1000340 (2009)
4. Awad, N.H., Ali, M.Z., Suganthan, P.N., Reynolds, R.G.: An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 2958–2965. IEEE (2016)
5. Barabási, A.L., Oltvai, Z.N.: Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* **5**(2), 101 (2004)
6. Barbierato, E., Bobbio, A., Gribaudo, M., Iacono, M.: Multiformalism to support software rejuvenation modeling, pp. 271–276 (2012)

7. Barbierato, E., Gribaudo, M., Iacono, M.: Modeling and evaluating the effects of Big Data storage resource allocation in global scale cloud architectures. *Int. J. Data Warehous. Min.* **12**(2), 1–20 (2016). <https://doi.org/10.4018/IJDWM.2016040101>
8. Barbierato, E., Gribaudo, M., Iacono, M.: Modeling hybrid systems in SIMTHESys. *Electron. Notes Theor. Comput. Sci.* **327**, 5–25 (2016)
9. Barbierato, E., Gribaudo, M., Iacono, M.: Simulating hybrid systems within SIMTHESys multi-formalism models. In: Fiems, D., Paolieri, M., Platis, A.N. (eds.) *EPEW 2016. LNCS*, vol. 9951, pp. 189–203. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46433-6_13
10. Barbierato, E., Gribaudo, M., Iacono, M., Jakóbbik, A.: Exploiting CloudSim in a multiformalism modeling approach for cloud based systems. *Simul. Model. Pract. Theory* (2018)
11. Benkner, S., et al.: PEPPHER: Efficient and productive usage of hybrid computing systems. *IEEE Micro* **31**(5), 28–41 (2011). <https://doi.org/10.1109/MM.2011.67>
12. Besozzi, D.: Reaction-based models of biochemical networks. In: Beckmann, A., Bienvenu, L., Jonoska, N. (eds.) *CiE 2016. LNCS*, vol. 9709, pp. 24–34. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40189-8_3
13. Bobbio, A., Cerotti, D., Gribaudo, M., Iacono, M., Manini, D.: Markovian agent models: a dynamic population of interdependent Markovian agents. In: Al-Begain, K., Bargiela, A. (eds.) *Seminal Contributions to Modelling and Simulation. SFMA*, pp. 185–203. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33786-9_13
14. Bordon, J., Moškon, M., Zimic, N., Mraz, M.: Fuzzy logic as a computational tool for quantitative modelling of biological systems with uncertain kinetic data. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **12**(5), 1199–1205 (2015)
15. Bowers, K.J., et al.: Scalable algorithms for molecular dynamics simulations on commodity clusters. In: *Proceedings of the ACM/IEEE SC 2006 Conference*, p. 43. IEEE (2006)
16. Brest, J., Greiner, S., Bošković, B., Mernik, M., Bošković, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **10**(6), 646–657 (2006)
17. Brest, J., Korošec, P., Šilc, J., Zamuda, A., Bošković, B., Maučec, M.S.: Differential evolution and differential ant-stigmergy on dynamic optimisation problems. *Int. J. Syst. Sci.* **44**(4), 663–679 (2013)
18. Brest, J., Maučec, M.S., Bošković, B.: Single objective real-parameter optimization: algorithm jSO. In: *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1311–1318. IEEE (2017)
19. Cash, J.R.: Backward differentiation formulae. In: Engquist, B. (ed.) *Encyclopedia of Applied and Computational Mathematics*, pp. 97–101. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-540-70529-1_94. *Computational Science & Engineering*, 1 (edn.) Springer, Heidelberg
20. Cazzaniga, P., et al.: Computational strategies for a system-level understanding of metabolism. *Metabolites* **4**(4), 1034–1087 (2014)
21. Cerotti, D., Gribaudo, M., Bobbio, A., Calafate, C.T., Manzoni, P.: A Markovian agent model for fire propagation in outdoor environments. In: Aldini, A., Bernardo, M., Bononi, L., Cortellessa, V. (eds.) *EPEW 2010. LNCS*, vol. 6342, pp. 131–146. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15784-4_9

22. Chellaboina, V., Bhat, S., Haddad, W., Bernstein, D.: Modeling and analysis of mass-action kinetics. *IEEE Control Syst. Mag.* **29**(4), 60–78 (2009). <https://doi.org/10.1109/MCS.2009.932926>
23. Chiappino-Pepe, A., Pandey, V., Ataman, M., Hatzimanikatis, V.: Integration of metabolic, regulatory and signaling networks towards analysis of perturbation and dynamic responses. *Curr. Opin. Syst. Biol.* **2**, 59–66 (2017)
24. Chrysos, G.: Intel® Xeon Phi™ Coprocessor-the Architecture. Intel Whitepaper (2014)
25. Ciardo, G., Jones III, R.L., Miner, A.S., Siminiceanu, R.I.: Logic and stochastic modeling with smart. *Perform. Eval.* **63**(6), 578–608 (2006)
26. Conti, V., Ruffo, S.S., Vitabile, S., Barolli, L.: BIAM: a new bio-inspired analysis methodology for digital ecosystems based on a scale-free architecture. *Soft Comput.* **23**, 1–18 (2017)
27. Cordero, F., Manini, D., Gribaudo, M.: Modeling biological pathways: an object-oriented like methodology based on mean field analysis. In: 2009 Third International Conference on Advanced Engineering Computing and Applications in Sciences, pp. 117–122, October 2009. <https://doi.org/10.1109/ADVCOMP.2009.25>
28. Cordero, F., Fornari, C., Gribaudo, M., Manini, D.: Markovian agents population models to study cancer evolution. In: Sericola, B., Telek, M., Horváth, G. (eds.) *Analytical and Stochastic Modeling Techniques and Applications*, pp. 16–32. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08219-6_2
29. Coulier, A., Hellander, A.: Orchestral: a lightweight framework for parallel simulations of cell-cell communication. arXiv preprint [arXiv:1806.10889](https://arxiv.org/abs/1806.10889) (2018)
30. Dada, J.O., Mendes, P.: Multi-scale modelling and simulation in systems biology. *Integr. Biol.* **3**(2), 86–96 (2011)
31. Das, S., Abraham, A., Chakraborty, U.K., Konar, A.: Differential evolution using a neighborhood-based mutation operator. *IEEE Trans. Evol. Comput.* **13**(3), 526–553 (2009)
32. Das, S., Mullick, S.S., Suganthan, P.N.: Recent advances in differential evolution – an updated survey. *Swarm Evol. Comput.* **27**, 1–30 (2016)
33. Davendra, D., Bialic-Davendra, M., Senkerik, R.: Scheduling the lot-streaming flowshop scheduling problem with setup time with the chaos-induced enhanced differential evolution. In: 2013 IEEE Symposium on Differential Evolution (SDE), pp. 119–126. IEEE (2013)
34. De Nooy, W., Mrvar, A., Batagelj, V.: *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, Cambridge (2018)
35. Dräger, A., Kronfeld, M., Ziller, M.J., Supper, J., Planatscher, H., Magnus, J.B.: Modeling metabolic networks in *C. glutamicum*: a comparison of rate laws in combination with various parameter optimization strategies. *BMC Syst. Biol.* **3**(1), 5 (2009)
36. Dubrovin, T., Jolma, A., Turunen, E.: Fuzzy model for real-time reservoir operation. *J. Water Resour. Plan. Manag.* **128**(1), 66–73 (2002)
37. Eldar, A., Elowitz, M.B.: Functional roles for noise in genetic circuits. *Nature* **467**(7312), 167 (2010)
38. Elf, J., Ehrenberg, M.: Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *IEE Proc.-Syst. Biol.* **1**(2), 230–236 (2004)
39. Elowitz, M.B., Levine, A.J., Siggia, E.D., Swain, P.S.: Stochastic gene expression in a single cell. *Science* **297**(5584), 1183–1186 (2002)
40. Faeder, J.R., Blinov, M.L., Hlavacek, W.S.: Rule-based modeling of biochemical systems with BioNetGen. In: Maly, I. (ed.) *Systems Biology*, pp. 113–167. Springer, Heidelberg (2009). https://doi.org/10.1007/978-1-59745-525-1_5

41. Fisher, C.P., Plant, N.J., Moore, J.B., Kierzek, A.M.: QSSPN: dynamic simulation of molecular interaction networks describing gene regulation, signalling and whole-cell metabolism in human cells. *Bioinformatics* **29**(24), 3181–3190 (2013)
42. Franceschinis, G., Gribaudo, M., Iacono, M., Marrone, S., Mazzocca, N., Vittorini, V.: Compositional modeling of complex systems: contact center scenarios in OsMoSys. In: ICATPN 2004, pp. 177–196 (2004)
43. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* **22**(4), 403–434 (1976)
44. Gillespie, D.T.: Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.* **115**(4), 1716–1733 (2001)
45. Gonçalves, E., et al.: Bridging the layers: towards integration of signal transduction, regulation and metabolism into mathematical models. *Mol. BioSyst.* **9**(7), 1576–1583 (2013)
46. Gribaudo, M., Iacono, M.: An introduction to multiformalism modeling (2013)
47. Gribaudo, M., Iacono, M., Levis, A.H.: An IoT-based monitoring approach for cultural heritage sites: the Matera case. *Concurr. Comput.: Pract. Exp.* **29**, e4153 (2017). <https://doi.org/10.1002/cpe.4153>
48. Gropp, W., Lusk, E., Skjellum, A.: Using MPI: Portable Parallel Programming with the Message-Passing Interface, vol. 1. MIT Press, Cambridge (1999)
49. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. *Nature* **433**(7028), 895 (2005)
50. Guo, S.M., Tsai, J.S.H., Yang, C.C., Hsu, P.H.: A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1003–1010. IEEE (2015)
51. Hadoop, A.: Apache Hadoop project. <https://hadoop.apache.org/>. Accessed 03 Nov 2018
52. Harris, L.A., Clancy, P.: A “partitioned leaping” approach for multiscale modeling of chemical reaction dynamics. *J. Chem. Phys.* **125**(14), 144107 (2006)
53. Harris, L.A., et al.: GPU-powered model analysis with PySB/cupSODA. *Bioinformatics* **33**(21), 3492–3494 (2017)
54. Hecker, M., Lambeck, S., Toepfer, S., Van Someren, E., Guthke, R.: Gene regulatory network inference: data integration in dynamic models review. *Biosystems* **96**(1), 86–103 (2009)
55. Horváth, A., Manini, D.: Parameter estimation of kinetic rates in stochastic reaction networks by the EM method. In: International Conference on BioMedical Engineering and Informatics, BMEI 2008, vol. 1, pp. 713–717. IEEE (2008)
56. European Bioinformatics Institute: EMBL-EBI annual scientific report 2013. https://www.embl.fr/aboutus/communication-outreach/publications/ebi-ar/ebi_ar.2013.pdf. Accessed 07 Dec 2018
57. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabási, A.L.: The large-scale organization of metabolic networks. *Nature* **407**(6804), 651 (2000)
58. Karr, J.R., et al.: A whole-cell computational model predicts phenotype from genotype. *Cell* **150**(2), 389–401 (2012)
59. Kauffman, K.J., Prakash, P., Edwards, J.S.: Advances in flux balance analysis. *Curr. Opin. Biotechnol.* **14**(5), 491–496 (2003)
60. Kessler, C., et al.: Programmability and performance portability aspects of heterogeneous multi-/manycore systems. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1403–1408. IEEE (2012)
61. Kitano, H.: Systems biology: a brief overview. *Science* **295**(5560), 1662–1664 (2002)

62. Klamt, S., Saez-Rodriguez, J., Gilles, E.D.: Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Syst. Biol.* **1**(1), 2 (2007)
63. Küffner, R., Petri, T., Windhager, L., Zimmer, R.: Petri nets with fuzzy logic (PNFL): reverse engineering and parametrization. *PLoS ONE* **5**(9), e12807 (2010)
64. Lacroix, V., Cottret, L., Thébault, P., Sagot, M.F.: An introduction to metabolic networks and their structural analysis. *IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB)* **5**(4), 594–617 (2008)
65. Li, K.B.: ClustalW-MPI: ClustalW analysis using distributed and parallel computing. *Bioinformatics* **19**(12), 1585–1586 (2003). <https://doi.org/10.1093/bioinformatics/btg192>
66. Liu, F., Heiner, M., Yang, M.: Fuzzy stochastic Petri nets for modeling biological systems with uncertain kinetic parameters. *PloS ONE* **11**(2), e0149674 (2016)
67. Macklin, D.N., Ruggero, N.A., Covert, M.W.: The future of whole-cell modeling. *Curr. Opin. Biotechnol.* **28**, 111–115 (2014)
68. Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl. Soft Comput.* **11**(2), 1679–1696 (2011)
69. Marx, V.: Biology: the big challenges of big data. *Nature* **498**, 255–260 (2013)
70. Memeti, S., Li, L., Pllana, S., Kolodziej, J., Kessler, C.: Benchmarking OpenCL, OpenACC, OpenMP, and CUDA: programming productivity, performance, and energy consumption. In: *Proceedings of the 2017 Workshop on Adaptive Resource Management and Scheduling for Cloud Computing, ARMS-CC 2017*, pp. 1–6. ACM, New York (2017). <https://doi.org/10.1145/3110355.3110356>
71. Memeti, S., Pllana, S.: Accelerating DNA sequence analysis using Intel® Xeon Phi™. In: *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 3, pp. 222–227, August 2015
72. Metlicka, M., Davendra, D.: Chaos driven discrete artificial bee algorithm for location and assignment optimisation problems. *Swarm Evol. Comput.* **25**, 15–28 (2015)
73. Mininno, E., Neri, F., Cupertino, F., Naso, D.: Compact differential evolution. *IEEE Trans. Evol. Comput.* **15**(1), 32–54 (2011)
74. Morris, M.K., Saez-Rodriguez, J., Sorger, P.K., Lauffenburger, D.A.: Logic-based models for the analysis of cell signaling networks. *Biochemistry* **49**(15), 3216–3224 (2010)
75. Mosterman, P.J., Vangheluwe, H.: Computer automated multi-paradigm modeling: an introduction. *Simulation* **80**(9), 433–450 (2004). <https://doi.org/10.1177/0037549704050532>
76. Nobile, M.S., Besozzi, D., Cazzaniga, P., Mauri, G.: GPU-accelerated simulations of mass-action kinetics models with cupSODA. *J. Supercomput.* **69**(1), 17–24 (2014)
77. Nobile, M.S., Cazzaniga, P., Tangherloni, A., Besozzi, D.: Graphics processing units in bioinformatics, computational biology and systems biology. *Brief. Bioinform.* **18**(5), 870–885 (2017). <https://doi.org/10.1093/bib/bbw058>
78. Nobile, M.S., Besozzi, D., Cazzaniga, P., Mauri, G., Pescini, D.: A GPU-based multi-swarm PSO method for parameter estimation in stochastic biological systems exploiting discrete-time target series. In: *Giacobini, M., Vanneschi, L., Bush, W.S. (eds.) EvoBIO 2012. LNCS*, vol. 7246, pp. 74–85. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29066-4_7

79. Nobile, M.S., Besozzi, D., Cazzaniga, P., Mauri, G., Pescini, D.: Estimating reaction constants in stochastic biological systems with a multi-swarm PSO running on GPUs. In: Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation, pp. 1421–1422. ACM (2012)
80. Nobile, M.S., Cazzaniga, P., Besozzi, D., Colombo, R., Mauri, G., Pasi, G.: Fuzzy self-tuning PSO: a settings-free algorithm for global optimization. *Swarm Evol. Comput.* **39**, 70–85 (2018)
81. Nobile, M.S., Cazzaniga, P., Besozzi, D., Pescini, D., Mauri, G.: cuTauLeaping: a GPU-powered tau-leaping stochastic simulator for massive parallel analyses of biological systems. *PLoS ONE* **9**(3), e91963 (2014)
82. Nobile, M.S., Mauri, G.: Accelerated analysis of biological parameters space using GPUs. In: Malyshkin, V. (ed.) *PaCT 2017*. LNCS, vol. 10421, pp. 70–81. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62932-2_6
83. Nobile, M.S., et al.: Computational intelligence for parameter estimation of biochemical systems. In: 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2018)
84. NVIDIA: CUDA C Programming Guide, September 2016. <http://docs.nvidia.com/cuda/cuda-c-programming-guide/>. Accessed 06 Nov 2018
85. NVIDIA: What is GPU-Accelerated Computing? April 2017. <http://www.nvidia.com/object/what-is-gpu-computing.html>. Accessed 03 Nov 2018
86. O'Driscoll, A., et al.: HBLAST: parallelised sequence similarity—a Hadoop MapReducable basic local alignment search tool. *J. Biomed. Inform.* **54**, 58–64 (2015). <https://doi.org/10.1016/j.jbi.2015.01.008>, <http://www.sciencedirect.com/science/article/pii/S1532046415000106>
87. Opara, K.R., Arabas, J.: Differential evolution: a survey of theoretical analyses. *Swarm Evol. Comput.* (2018). <https://doi.org/10.1016/j.swevo.2018.06.010>
88. OpenMP: OpenMP 4.0 Specifications, July 2013. <http://www.openmp.org/specifications/>. Accessed 10 Mar 2018
89. Padua, D.: *Encyclopedia of Parallel Computing*. Springer, Heidelberg (2011)
90. Piotrowski, A.P.: Review of differential evolution population size. *Swarm Evol. Comput.* **32**, 1–24 (2017)
91. Piotrowski, A.P.: aL-SHADE optimization algorithms with population-wide inertia. *Inf. Sci.* **468**, 117–141 (2018)
92. Piotrowski, A.P., Napiorkowski, J.J.: Some metaheuristics should be simplified. *Inf. Sci.* **427**, 32–62 (2018)
93. Piotrowski, A.P., Napiorkowski, J.J.: Step-by-step improvement of JADE and SHADE-based algorithms: success or failure? *Swarm Evol. Comput.* (2018). <https://doi.org/10.1016/j.swevo.2018.03.007>
94. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. *Swarm Intell.* **1**(1), 33–57 (2007)
95. Poovathingal, S.K., Gunawan, R.: Global parameter estimation methods for stochastic biochemical systems. *BMC Bioinform.* **11**(1), 414 (2010)
96. Provost, A., Bastin, G.: Metabolic flux analysis: an approach for solving non-stationary underdetermined systems. In: *CD-Rom Proceedings 5th MATHMOD Conference*, Paper, vol. 207. Citeseer (2006)
97. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13**(2), 398–417 (2009)
98. Qu, B.Y., Suganthan, P.N., Liang, J.J.: Differential evolution with neighborhood mutation for multimodal optimization. *IEEE Trans. Evol. Comput.* **16**(5), 601–614 (2012)

99. Saastamoinen, K., Ketola, J., Turunen, E.: Defining athlete's anaerobic and aerobic thresholds by using similarity measures and differential evolution. In: 2004 IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 1331–1335. IEEE (2004)
100. Sadasivam, G.S., Baktavatchalam, G.: A novel approach to multiple sequence alignment using Hadoop data grids. In: Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud, MDAC 2010, pp. 2:1–2:7. ACM, New York (2010). <https://doi.org/10.1145/1779599.1779601>
101. Sanders, W.H.: Integrated frameworks for multi-level and multi-formalism modeling. In: Proceedings 8th International Workshop on Petri Nets and Performance Models (Cat. No. PR00331), pp. 2–9, September 1999. <https://doi.org/10.1109/PNPM.1999.796527>
102. Sanders, W.H., Courtney, T., Deavours, D.D., Daly, D., Derisavi, S., Lam, V.V.: Multi-formalism and multi-solution-method modeling frameworks: the Möbius approach (2003)
103. Schilling, C.H., Letscher, D., Palsson, B.Ø.: Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *J. Theor. Biol.* **203**(3), 229–248 (2000)
104. Senkerik, R., Pluhacek, M., Oplatkova, Z.K., Davendra, D., Zelinka, I.: Investigation on the differential evolution driven by selected six chaotic systems in the task of reactor geometry optimization. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 3087–3094. IEEE (2013)
105. Senkerik, R., Viktorin, A., Pluhacek, M., Janostik, J., Davendra, D.: On the influence of different randomization and complex network analysis for differential evolution. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 3346–3353. IEEE (2016)
106. Senkerik, R., Viktorin, A., Pluhacek, M., Kadavy, T., Oplatkova, Z.K.: Differential evolution and chaotic series. In: 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 1–5. IEEE (2018)
107. Senkerik, R., Zelinka, I., Pluhacek, M., Davendra, D., Oplatková Kominkova, Z.: Chaos enhanced differential evolution in the task of evolutionary control of selected set of discrete chaotic systems. *Sci. World J.* **2014**, 1–12 (2014)
108. Skanderova, L., Fabian, T.: Differential evolution dynamics analysis by complex networks. *Soft Comput.* **21**(7), 1817–1831 (2017)
109. Skanderova, L., Fabian, T., Zelinka, I.: Differential evolution dynamics modeled by longitudinal social network. *J. Intell. Syst.* **26**(3), 523–529 (2017)
110. Skanderova, L., Fabian, T., Zelinka, I.: Analysis of causality-driven changes of diffusion speed in non-Markovian temporal networks generated on the basis of differential evolution dynamics. *Swarm Evol. Comput.* **44**, 212–227 (2018)
111. Sneddon, M.W., Faeder, J.R., Emonet, T.: Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nat. Methods* **8**(2), 177 (2011)
112. Spark, A.: Apache Spark project. <https://spark.apache.org/>. Accessed 03 Nov 2018
113. Stegle, O., Teichmann, S.A., Marioni, J.C.: Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.* **16**(3), 133 (2015)
114. Stelling, J.: Mathematical models in microbial systems biology. *Curr. Opin. Microbiol.* **7**(5), 513–518 (2004)
115. Stone, J.E., Gohara, D., Shi, G.: OpenCL: a parallel programming standard for heterogeneous computing systems. *Comput. Sci. Eng.* **12**(1–3), 66–73 (2010)

116. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**, 341–359 (1997)
117. Swainston, N., et al.: Recon 2.2: from reconstruction to model of human metabolism. *Metabolomics* **12**(7), 109 (2016)
118. Szallasi, Z., Stelling, J., Periwai, V.: *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*. The MIT Press, Cambridge (2006)
119. Talay, D.: *Numerical Solution of Stochastic Differential Equations*. Taylor & Francis, Milton Park (1994)
120. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for differential evolution. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 71–78. IEEE (2013)
121. Tanabe, R., Fukunaga, A.S.: Improving the search performance of SHADE using linear population size reduction. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1658–1665. IEEE (2014)
122. Tanabe, R., Fukunaga, A.S.: How far are we from an optimal, adaptive DE? In: 14th International Conference on Parallel Problem Solving from Nature (PPSN XIV). IEEE (2016, accepted)
123. Tangherloni, A., Nobile, M.S., Besozzi, D., Mauri, G., Cazzaniga, P.: LASSIE: simulating large-scale models of biochemical systems on GPUs. *BMC Bioinform.* **18**(1), 246 (2017)
124. Teijeiro, D., Pardo, X.C., Penas, D.R., González, P., Banga, J.R., Doallo, R.: A cloud-based enhanced differential evolution algorithm for parameter estimation problems in computational systems biology. *Clust. Comput.* **20**(3), 1937–1950 (2017)
125. Trivedi, K.S.: SHARPE 2002: symbolic hierarchical automated reliability and performance evaluator. In: *Proceedings International Conference on Dependable Systems and Networks*, p. 544, June 2002. <https://doi.org/10.1109/DSN.2002.1028975>
126. Turunen, E.: *Mathematics Behind Fuzzy Logic*. Physica-Verlag, Heidelberg (1999)
127. Turunen, E.: Using GUHA data mining method in analyzing road traffic accidents occurred in the years 2004–2008 in Finland. *Data Sci. Eng.* **2**(3), 224–231 (2017)
128. Vazquez, A., Liu, J., Zhou, Y., Oltvai, Z.N.: Catabolic efficiency of aerobic glycolysis: the Warburg effect revisited. *BMC Syst. Biol.* **4**(1), 58 (2010)
129. Viebke, A., Pllana, S.: The potential of the Intel (R) Xeon Phi for supervised deep learning. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications, pp. 758–765, August 2015. <https://doi.org/10.1109/HPCC-CSS-ICSS.2015.45>
130. Viktorin, A., Pluhacek, M., Senkerik, R.: Network based linear population size reduction in shade. In: 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), pp. 86–93. IEEE (2016)
131. Viktorin, A., Senkerik, R., Pluhacek, M., Kadavy, T.: Towards better population sizing for differential evolution through active population analysis with complex network. In: Barolli, L., Terzo, O. (eds.) *CISIS 2017. AISC*, vol. 611, pp. 225–235. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-61566-0_22
132. Viktorin, A., Senkerik, R., Pluhacek, M., Kadavy, T., Zamuda, A.: Distance based parameter adaptation for success-history based differential evolution. *Swarm Evol. Comput.* <https://doi.org/10.1016/j.swevo.2018.10.013>. Accessed 12 Nov 2018
133. Viktorin, A., Senkerik, R., Pluhacek, M., Zamuda, A.: Steady success clusters in differential evolution. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8. IEEE (2016)

134. Vitabile, S., Conti, V., Lanza, B., Cusumano, D., Sorbello, F.: Metabolic networks robustness: theory, simulations and results. *J. Interconnect. Netw.* **12**(03), 221–240 (2011)
135. Vitabile, S., Conti, V., Lanza, L., Cusumano, D., Sorbello, F.: Topological information, flux balance analysis, and extreme pathways extraction for metabolic networks behaviour investigation. In: Workshop on Italian Neural Network, vol. 234, pp. 66–73. IOS Press (2011)
136. Vitello, G., Alongi, A., Conti, V., Vitabile, S.: A bio-inspired cognitive agent for autonomous urban vehicles routing optimization. *IEEE Trans. Cogn. Dev. Syst.* **9**(1), 5–15 (2017)
137. Wiback, S.J., Palsson, B.O.: Extreme pathway analysis of human red blood cell metabolism. *Biophys. J.* **83**(2), 808–818 (2002)
138. Wienke, S., Springer, P., Terboven, C., an Mey, D.: OpenACC—first experiences with real-world applications. In: Kaklamanis, C., Papatheodorou, T., Spirakis, P.G. (eds.) Euro-Par 2012. LNCS, vol. 7484, pp. 859–870. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32820-6_85
139. Wolkenhauer, O.: Why model? *Front. Physiol.* **5**, 21 (2014)
140. Wu, G., Shen, X., Li, H., Chen, H., Lin, A., Suganthan, P.N.: Ensemble of differential evolution variants. *Inf. Sci.* **423**, 172–186 (2018)
141. Wynn, M.L., Consul, N., Merajver, S.D., Schnell, S.: Logic-based models in systems biology: a predictive and parameter-free network analysis method. *Integr. Biol.* **4**(11), 1323–1337 (2012)
142. Zamuda, A., Brest, J.: Environmental framework to visualize emergent artificial forest ecosystems. *Inf. Sci.* **220**, 522–540 (2013). <https://doi.org/10.1016/j.ins.2012.07.031>
143. Zamuda, A., Brest, J.: Vectorized procedural models for animated trees reconstruction using differential evolution. *Inf. Sci.* **278**, 1–21 (2014)
144. Zamuda, A., Brest, J.: Self-adaptive control parameters' randomization frequency and propagations in differential evolution. *Swarm Evol. Comput.* **25**, 72–99 (2015)
145. Zamuda, A., Hernández Sosa, J.D.: Differential evolution and underwater glider path planning applied to the short-term opportunistic sampling of dynamic mesoscale ocean structures. *Appl. Soft Comput.* **24**, 95–108 (2014)
146. Zamuda, A., Nicolau, M., Zarges, C.: A black-box discrete optimization benchmarking (BB-DOB) pipeline survey: taxonomy, evaluation, and ranking. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO 2018), pp. 1777–1782 (2018)
147. Zamuda, A., Sosa, J.D.H., Adler, L.: Constrained differential evolution optimization for underwater glider path planning in sub-mesoscale eddy sampling. *Appl. Soft Comput.* **42**, 93–118 (2016)
148. Zamuda, A., Sosa, J.D.H.: Success history applied to expert system for underwater glider path planning using differential evolution. *Expert. Syst. Appl.* **119**, 155–170 (2019)
149. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **13**(5), 945–958 (2009)
150. Zhou, Y., Liepe, J., Sheng, X., Stumpf, M.P., Barnes, C.: GPU accelerated biochemical network simulation. *Bioinformatics* **27**(6), 874–876 (2011)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

