Biochemical parameter estimation vs. benchmark functions: a comparative study of optimization performance and representation design

Andrea Tangherloni[a,b,c], Simone Spolaor[a], Paolo Cazzaniga[d,e], Daniela Besozzi[a], Leonardo Rundo[a,f,g,h], Giancarlo Mauri[a,e], Marco S. Nobile[a,e,*]

[a] *Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milan, Italy*
[b] *University of Cambridge, Department of Haematology, Cambridge, UK*
[c] *Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Hinxton, UK*
[d] *Department of Human and Social Sciences, University of Bergamo, Bergamo, Italy*
[e] *SYSBIO.IT Centre for Systems Biology, Milan, Italy*
[f] *Institute of Molecular Bioimaging and Physiology, Italian National Research Council, Cefalù (PA), Italy*
[g] *University of Cambridge, Department of Radiology, CB2 0QQ Cambridge, UK*
[h] *Cancer Research UK Cambridge Centre, Cambridge, UK*

Abstract

Computational Intelligence methods, which include Evolutionary Computation and Swarm Intelligence, can efficiently and effectively identify optimal solutions to complex optimization problems by exploiting the cooperative and competitive interplay among their individuals. The exploration and exploitation capabilities of these meta-heuristics are typically assessed by considering well-known suites of benchmark functions, specifically designed for numerical global optimization purposes. However, their performances could drastically change in the case of real-world optimization problems. In this paper, we investigate this issue by considering the Parameter Estimation (PE) of biochemical systems, a common computational problem in the field of Systems Biology. In order to evaluate the effectiveness of various meta-heuristics in solving the PE problem, we compare their performance by considering a set of benchmark functions and a set of synthetic biochemical models characterized by a search space with an increasing number of dimensions. Our results show that some state-of-the-art optimization

[*] Corresponding author
Email address: nobile@disco.unimib.it (Marco S. Nobile)

methods—able to largely outperform the other meta-heuristics on benchmark functions—are characterized by considerably poor performances when applied to the PE problem. We also show that a limiting factor of these optimization methods concerns the representation of the solutions: indeed, by means of a simple semantic transformation, it is possible to turn these algorithms into competitive alternatives. We corroborate this finding by performing the PE of a model of metabolic pathways in red blood cells. Overall, we can state that classic benchmark functions are not fully representative of the complexity of real-world optimization problems, such as the PE of biochemical systems.

Keywords: Benchmark functions, Parameter Estimation, Biochemical simulation, Systems Biology, Fuzzy Logic, Self-tuning algorithms

1. Introduction

Population-based meta-heuristics can efficiently and effectively identify optimal solutions of complex computational problems, by exploiting the cooperative and competitive interplay among the individuals. A variety of methodologies has been proposed, inspired by processes of natural selection (e.g., Genetic Algorithms (GAs) [1], Differential Evolution (DE) [2], Evolution Strategy (ES) [3]), as well as super-organisms and the emergent intelligence of groups of animals (e.g., Artificial Bee Colony (ABC) [4], Particle Swarm Optimization (PSO) [5]).

The performances of these meta-heuristics are typically assessed by relying on different sets of benchmark functions, specifically designed to test the search capabilities of the various optimization strategies [6]. However, when the same algorithms are applied to real-world problems pertaining to different application domains, which can involve continuous or discrete optimization tasks, their performances may considerably change. For instance, this phenomenon was observed in the case of the optimization of atomic and molecular clusters [7], building energy systems [8] and aircraft design [9].

This circumstance is coherent with the probabilistic re-formulation of the *no-free lunch theorem* (NFL) proposed by Lockett and Miikkulainen in 2017

[10]. The original version of the NFL theorem [11, 12]—which holds in the case of combinatorial optimization problems, and is automatically met for optimization algorithms running on digital computers where real values are encoded by a finite number of bits—states that no algorithm outperforms all the competitors in any optimization problem. Over the last years, the validity of NFL in continuous search spaces has been debated [13, 14], until Lockett and Miikkulainen proposed a novel probabilistic approach that can be applied to prove the validity of the theorem in continuous domains. Hence, according to this result, any meta-heuristics working in a continuous domain and showing outstanding performance when applied to benchmark test functions might not work well on (a subset of) real-world problems [15]. For instance, Da Ros *et al.* [16] compared stochastic optimization methods (i.e., ABC, DE, PSO, and Simulated Annealing) for the estimation of kinetic parameters of a biochemical model for alcoholic fermentation in bioreactors. Their results show that benchmark functions are not representative of real optimization problems, and the evaluation of global optimization meta-heuristics based on these benchmark functions alone may induce strong biases.

In this paper, we investigate the problem of the Parameter Estimation (PE) of biochemical systems, typical of Systems Biology analyses [17, 18]. This research field aims at a thorough understanding of biological processes at a system-level by explicitly considering the complex interactions among biomolecules [19]. In this context, mechanistic mathematical models and computational methods represent valuable and integrative tools to classic experimental biology, paving the way to a global-level understanding of the emergent behavior of biological processes, thanks to the elucidation of the mechanisms that govern their functioning [20]. A precise assignment of the kinetic parameters, which control the rate of the reactions and ultimately drive the emergent behavior of the biochemical system, is mandatory to accurately simulate the dynamics of these models. Unfortunately, these parameters are difficult or even impossible to measure by means of laboratory experiments, so that they are generally estimated using Computational Intelligence methods [17], whereby the PE is formulated as an optimization (minimization) problem.

The goal of this work is to show that state-of-the-art optimization algorithms, which are generally able to largely outperform the other meta-heuristics on classic benchmark functions, could be characterized by very poor performances when applied to real-world problems without considering their specific peculiarities. To this aim, we exploit six well-known benchmark functions to compare the performance of ABC [4], Covariance Matrix Adaptation ES (CMA-ES) [21], DE [22], Estimation of Distribution Algorithm (EDA) [23], GA [1], PSO [5], and its fuzzy-based settings-free variant FST-PSO [24]. Then, we apply the same algorithms to the PE problem by using a set of synthetic biochemical models of increasing size (25 or 50 molecular species and reactions). By measuring the convergence speed and the quality of the final results achieved by the different meta-heuristics, we show that the algorithms able to perform efficiently on benchmark functions can be completely unfit for the PE problem. We also point out that alternative semantics for the parameters can radically change the performances of the meta-heuristics. Specifically, we show that a simple logarithmic transformation of the parameters can turn the previously outperformed algorithms into competitive alternatives. We validate this finding by executing a PE on a mechanistic model of intracellular metabolic pathways.

The paper is structured as follows. In Section 2 we recall the definition of the optimization algorithms considered in this work, and introduce the PE problem together with the used fitness function. The results are described and discussed in Section 3. Finally, conclusions and future developments are presented in Section 4.

2. Optimization methods and problem formulation

We recall here the main concepts and the notation regarding the Computational Intelligence methods used to optimize the benchmark functions and to solve the PE problem of biochemical systems. In what follows, we denote by n the size of the population, by M the number of dimensions of the search space, and by β_m^{min} and β_m^{max} (with $m = 1, \dots, M$) the boundaries along the m-th dimension of the search space.

2.1. Computational Intelligence for global optimization

We briefly describe here the Computational Intelligence techniques considered in this work, which have been analyzed in a preliminary study concerning the PE problem [25].

Artificial Bee Colony. ABC belongs to the population-based swarm intelligence optimization algorithms [4]. In ABC, three different groups of honey bees (i.e., $n_s \in \mathbb{N}$ scouts, $n_o \in \mathbb{N}$ onlookers, and $n_e \in \mathbb{N}$ employees) compose the colony and cooperate in identifying the best food resources, which correspond to solutions with the best fitness values. This is accomplished by randomly distributing the scouts across the search space, while employees and onlookers perform a local search nearby the promising positions found by the scouts. In particular, scouts randomly establish a new food source (i.e., a position in the search space) and they become employees; then, onlookers are assigned to the position of employees' food sources, proportionally to their fitness values. When onlookers cannot improve their position anymore, the food source is abandoned and bees return to the hive to start a new search for food by randomly choosing a new position.

Covariance Matrix Adaptation Evolution Strategy. CMA-ES is an evolutionary computation method based on a stochastic continuous optimization procedure [21]. During the optimization process, at each iteration, a simple ES using mutation operator is applied to produce new individuals by perturbing either the best individual (deterministically selected) or a newly created individual. In order to improve the quality of the mutation steps, CMA introduces a dynamical adaptation of the multivariate normally distributed random deviates, by modifying an M-dimensional ellipsoid distribution, whose size and rotation are updated during the generations according to the optimization convergence. To be more precise, CMA-ES relies on an $M \times M$ symmetric positive covariance matrix, which is adapted to capture possible existing pairwise dependencies among the components of the problem under investigation. Moreover, this adaptation scheme determines the correct scaling of a given problem, allowing for invariance with respect to any orientation of the coordinate system. According to Hansen [3], the following settings must be carefully selected: *i*) the population size n, *ii*) the initial mean χ_0^{ES}, and *iii*) the initial standard deviations σ_0^{ES} (here called step-size).

Differential Evolution. DE is a population-based algorithm that exploits crossover and mutation operators during the evolutionary process [22]. Among all the existing versions of DE and according to the classic DE taxonomy, we selected the DE/rand/1/bin strategy wherein a differential weight $F \in [0, 2]$ balances the recombination of three randomly selected individuals, and a random crossover with the parent solution is applied with probability $CR \in [0, 1]$. To be more precise, DE is based on the following iterative scheme: (*i*) the mutation operator is applied to a randomly selected individual $\mathbf{x}$; (*ii*) three individuals are randomly selected and used to calculate the recombined individual $\mathbf{z}$; (*iii*) a trial vector is obtained by applying the crossover to $\mathbf{z}$ and $\mathbf{x}$, and it replaces $\mathbf{x}$ in the next generation if it is characterized by an improved fitness value.

Estimation of Distribution Algorithm. In EDA, the promising candidate solutions found during the optimization process are used to build probabilistic models, which are sampled to explore the search space [26]. At the beginning of the optimization process, a model encoding an *a priori* distribution (e.g., uniform, normal) of feasible solutions must be provided. During each iteration, λ new individuals are used, according to their fitness values, to refine the distribution and generate new offspring. Among the existing variants of EDA [23], we exploit the Population-Based Incremental Learning [27] that dynamically adapts the underlying Gaussian generative model by using the information about the $\mu < \lambda$ best individuals found during each generation. The initial distribution is initialized on the centroid χ_0^{EDA} of the search space, and for each dimension the standard deviation is set to σ_0^{EDA}.

Genetic Algorithms. GAs are a population-based optimization technique mimicking Darwinian processes on evolutionary dynamics *via* natural selection

[1]. The initial population of randomly generated individuals undergoes, during each generation, a selection process and is modified by applying crossover and mutation operators with probabilities p_{cr} and p_{mu}, respectively. Differently from the ESs that apply the selection strategy after recombination, GAs exploit a selection procedure before the crossover. The mutation operator is then employed to introduce new genetic material into the population, thus preventing the premature convergence to local optimal solutions. The mutation operator used in this work is a Gaussian perturbation with standard deviation σ^{GA}.

Particle Swarm Optimization. PSO is a population-based meta-heuristic belonging to swarm intelligence methods and suitable for real-valued optimization problems [5]. A swarm of candidate solutions (called particles) moves inside a bounded search space and cooperates to identify the optimal solution. During each iteration, each particle changes its position according to the best positions found so far by the particle itself and by the entire swarm. The social factor $c_{soc} \in \mathbb{R}^+$ and the cognitive factor $c_{cog} \in \mathbb{R}^+$ are used to balance the exploration and exploitation strategies, respectively. To avoid chaotic behaviors in the swarm, the velocity of each particle is weighted by an inertia factor $w \in \mathbb{R}^+$ [28], and limited by maximum velocity values $\mathbf{v}_{max} = (v_{max_1}, \ldots, v_{max_M})$ proportional to the distance between the boundaries of the search space. A vector of minimum velocities $\mathbf{v}_{min}$ is often used to prevent stagnation and to keep a high diversity inside the swarm [29].

Fuzzy Self-Tuning Particle Swarm Optimization. In order to dynamically adjust the PSO settings during the optimization process, Fuzzy Logic can be exploited [30]. Generally, the PSO versions hybridized with Fuzzy Logic update the settings of *all* particles by using a common set of values determined by means of a fuzzy rule-based system. Here, we consider FST-PSO [24], which makes use of Fuzzy Logic to automatically infer at run-time a different set of PSO settings for *each* particle of the swarm [31, 32].

2.2. Benchmark function suites for numerical optimization

Several benchmark function suites have been proposed in the literature to evaluate the performance of global optimization techniques, since real-world problems cannot be straightforwardly exploited to this purpose. Indeed, real-world problems are often characterized by additional complex features that basic optimization algorithms might not be able to handle [33]. Considering the specific case of real-parameter numerical optimization, every year research competitions are organized by the IEEE Congress on Evolutionary Computation (CEC) [34] and the Genetic and Evolutionary Computation Conference (GECCO). Among them, it is worth mentioning the workshop on Real-Parameter Black-Box Optimization Benchmarking (BBOB) [35] that exploits the COmparing Continuous Optimisers (COCO) benchmarking platform [36]. On the other hand, there exist real-world problems that have intrinsic discrete structure and solutions; in this context Doerr *et al.* [37] extended the COCO software by introducing pseudo-Boolean optimization problems, providing an environment to empirically analyze and evaluate the performance of pseudo-Boolean black-box heuristics. A generalized and dynamic benchmark generator was also proposed in [38] to construct dynamic environments in the binary, real and combinatorial spaces.

Even though many different benchmark functions were proposed for continuous optimization (e.g., global optimization, dynamic optimization, multimodal optimization), a complete and unified framework to generate benchmark functions that can take into account different properties pertaining to real-world problems has not been yet proposed. Recently, an attempt in this direction was made by Li *et al.*, who presented a novel framework that aims at constructing benchmark functions having features that might be met in real-world problems, such as non-linearity and discontinuity [39]. This framework is based on a multidimensional tree, which is exploited to partition the search space and to select the best set of simple functions for each subspace, according to the characteristics underlying the considered subspace.

Here, we employed the following functions from the CEC‘17 benchmark problems for single-objective real-parameter numerical optimization [40], to assess the performance of the meta-heuristics described in Section 2.1:

- the shifted/rotated Rosenbrock's (f_4) and Levy's (f_9) functions, as representatives of multimodal and non-separable problems characterized by a large number of local minima and whose optimum is not centered in $\mathbf{0}$;
- the hybrid Ackley's (f_{13}), Griewank's (f_{15}), expanded Griewank plus Rosenbrock (f_{19}), and Schaffer's F7 (f_{20}) functions.

It is worth noting that hybrid functions are composed of multiple basic functions, assigned to different components of the search space. Each component is characterized by a specific behavior and, possibly, a different sensitivity, that is, a measure of how much a variation in one (or more) individuals' elements affects the final fitness value. These characteristics, along with the shifted optimum and non-separability, are supposed to mimic real-world problems, hence including the PE of biochemical systems. The goal of this paper is to show that hybrid functions actually cannot capture all the features of the fitness landscapes defined by the PE problem.

2.3. Parameter Estimation of biochemical systems

Biochemical systems can be formalized as mechanistic and fully parameterized reaction-based models (RBMs) [41]. A RBM is defined by specifying 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 $\mathcal{R}$, and the initial concentration $X_i \in \mathbb{R}_0^+$ of each species $S_i \in \mathcal{S}$, with $i = 1, \dots, N$. Any RBM can be represented in a compact matrix-vector form $\mathbf{AS} \xrightarrow{\mathbf{K}} \mathbf{BS}$, 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 a RBM and assuming the law of mass-action [42], the system of coupled Ordinary Differential

Equations (ODEs) describing the variation in time of the species concentrations is obtained as follows:

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

where $\mathbf{X} = (X_1, \ldots, X_N)$ represents the state of the system at time t, $\mathbf{X}^{\mathbf{A}}$ denotes the vector-matrix exponentiation form [42], while the symbol $\odot$ denotes the Hadamard product.

Generally, the definition and analysis of RBMs are hindered by the lack of quantitative data related to the kinetic parameters—which are generally hard or even impossible to measure by classic laboratory experiments, but indispensable to run simulations—therefore leading to the PE problem. In what follows, we plausibly assume to have a complete knowledge of the stoichiometric matrices $\mathbf{A}$ and $\mathbf{B}$, as well as of the initial concentrations of all species, but no information about the kinetic constants. To infer the unknown parameters, we rely on the availability of discrete-time target series (DTTS) consisting of experimental data that can be measured for some species in $\mathcal{S}' \subseteq \mathcal{S}$. In what follows, we denote by $Y_q(\tau_f)$ the concentration of the species $S_q \in \mathcal{S}'$ experimentally measured at time τ_f, for some $f = 1, \ldots, F$ (where F is the number of sampled time points), and by $X_q^{\mathbf{K}}(\tau_f)$ its concentration at time τ_f obtained as the result of a simulation run by using the vector of kinetic constants $\mathbf{K}$.

To evaluate the quality of the individuals of the optimization algorithms, where an individual encodes a candidate model parameterization $\mathbf{K}$, we exploit the following fitness function:

$$\mathcal{F}(\mathbf{K}) = \sum_{f=1}^{F} \sum_{q=1}^{Q} \frac{|Y_q(\tau_f) - X_q^{\mathbf{K}}(\tau_f)|}{Y_q(\tau_f)}. \tag{2}$$

$\mathcal{F}(\cdot)$ is the relative point-to-point distance between the DTTS and the simulation dynamics related to the species belonging to $\mathcal{S}'$. Since this fitness function reflects the quality of the candidate parameterization with respect to the available experimental data, $\mathcal{F}(\mathbf{K})$ must be minimized to identify a vector $\mathbf{K}$ that is used to obtain a simulated dynamics overlapping at best the DTTS. The fitness function defined in Eq. (2) is commonly used in the field of Systems Biology, and the reason is two-fold: (*i*) the function correctly reflects the overlapping of the simulated dynamics with respect to the available experimental data; (*ii*) differently from common root mean squared error, Eq. (2) exploits the normalization term at the denominator to accumulate relative distances instead of absolute distances, in order to prevent that the most abundant chemical species could have a higher impact on the final fitness value[1].

In order to investigate the performance of the optimization algorithms listed in Section 2.1, we exploited a custom computational tool to randomly generate 12 different instances of RBMs of increasing size (6 RBMs are characterized by 25 reactions and species, 6 RBMs are characterized by 50 reactions and species). Each RBM satisfies the following characteristics:

- the initial concentrations of the molecular species are sampled from a log-uniform distribution in the interval $[10^{-6}, 1)$;
- the values of the kinetic constants are sampled from a log-uniform distribution in the interval $[10^{-8}, 10]$;
- since reactions simultaneously involving more than two reactants have a probability to take place close to zero, the stoichiometric matrix $\mathbf{A}$ is created by using only zero, first, and second-order reactions (i.e., at most 2 reactant molecules of the same or different species);
- the stoichiometric matrix $\mathbf{B}$ is created using at most 2 product molecules for each reaction.

We exploited a log-uniform distribution (i.e., a uniform distribution in the logarithmic space) since the concentrations and kinetic constants of biochemical systems generally span over multiple orders of magnitude [44, 45].

[1] The fitness function used in this work represents a variant of the Mean Absolute Percent Error (MAPE) [43]. The fitness landscape shaped by MAPE is identical to Eq. (2), although it is scaled due to the multiplicative constant used by MAPE to calculate the average of the contributions.

Deterministic simulations of RBMs and the fitness function evaluations were executed in parallel by offloading the calculations onto Graphics Processing Units (GPUs) by means of cupSODA [46, 47]. cupSODA is a simulation tool designed for biochemical systems based on mass-action kinetics [48, 42], which automatically converts the RBM provided by the user into the corresponding systems of coupled ODEs (and the associated Jacobian matrix). cupSODA can simultaneously perform multiple simulations, which are automatically executed by separate threads in the GPU. At the end of the simulations, cupSODA can also calculate in parallel the fitness function defined in Eq. (2), allowing for strongly reducing the running time of the PE.

3. Results

In this section we show the results of our study on the performance of the investigated meta-heuristics, and the semantic logarithmic transformation of the kinetic parameters to solve the PE problem. In all tests we used the implementation of ABC provided by the SwarmPackagePy library (v. 1.0.0a5). CMA-ES, DE, EDA, and GA were implemented by using the Distributed Evolutionary Algorithms in Python (DEAP) framework (v. 1.2.0) [49]. We opted for this specific library because it simplifies the integration of cupSODA (v. 1.1.0) for the fitness evaluation. We implemented PSO by using the Python programming language (v. 2.7.13) and the NumPy library (v. 1.13.3). Finally, FST-PSO (v. 1.4.8) was downloaded from the PyPI repository.

3.1. Comparison of the meta-heuristics performance

We present here the comparison of the performance of the meta-heuristics described in Section 2.1, exploited for the PE of 12 randomly generated RBMs (Section 2.3), and for the optimization of the selected benchmark functions from the CEC‘17 suite [40] (Section 2.2).

Table 1 summarizes the functioning settings of all the meta-heuristics used in this work. In the case of the algorithms implemented by using DEAP and

Table 1: Functioning settings of the exploited optimization algorithms.

Algorithm	**Settings**
ABC	$n_o = \lfloor 0.5 \cdot n \rfloor, n_e = \lfloor 0.4 \cdot n \rfloor, n_s = n - n_o - n_e$
CMA-ES	$\chi_0^{ES} = \left(\beta_m^{max} + \beta_m^{min}\right)/2, \sigma_0^{ES} = 1.0$
DE	$CR = 0.25, F = 1.0$, mutation strategy: DE/rand/1/bin
EDA	$\chi_0^{EDA} = \left(\beta_m^{max} + \beta_m^{min}\right)/2, \sigma_0^{EDA} = 1.0, \lambda = n, \mu = 32$
GA	$p_{mu} = 0.2, p_{cr} = 0.99$, tournament selection ($\kappa = 3$), two point crossover, Gaussian mutation with $\sigma^{GA} = 1.0$
PSO	$c_{cog} = c_{soc} = 1.496,\ w = 0.729, v_{min_m} = 0, v_{max_m} = 0.2 \cdot \psi_m$, where $\psi_m = \beta_m^{max} - \beta_m^{min}$
FST-PSO	—

SwarmPackagePy, we tested *off-the-shelf* optimization by using the default settings. For PSO, which was implemented from scratch for this work, we used the most widespread settings relying on the analyses conducted in [50, 51, 52]. The search space for the PE problem was set to $[10^{-10}, 100]^M$, while for the benchmark functions was set to $[-100, 100]^M$. The population size n was calculated by exploiting the following heuristic: $n = 32 \times \left\lceil \frac{\sqrt{M}}{2} \right\rceil$, which takes into account both the number of dimensions M and the CUDA warp size. The latter is critical to fully leverage the power of modern GPUs to simulate the dynamics of the RBMs by means of cupSODA [46]. A population size $n = 96$ and $n = 128$ for $M = 25$ and $M = 50$, respectively, is obtained by applying the aforementioned heuristic.

In order to collect statistically sound results, we performed 15 repetitions for each meta-heuristic keeping track of the best fitness value found during each iteration, and then calculating the average best fitness (ABF). In the following figures, each meta-heuristic is identified by a specific color: ABC is grey, CMA-ES is yellow, DE is pink, EDA is light blue, GA is blue, PSO is green and FST-PSO is orange. We also tested an alternative version of FST-PSO whose minimum velocity throttling is disabled; we refer to this version as FST-PSO (no $\mathbf{v}_{min}$) and we depict it with the magenta color in the figures.

In Figs. 1 and 2 we show the results concerning the convergence speed in terms of ABF obtained by the meta-heuristics on the benchmark functions with $M = 25$ dimensions and on the RBMs with $M = 25$ kinetic constants to be estimated. The results reveal that in the case of the benchmark functions, CMA-ES is characterized by the best performance, except for the case of the f_9 and f_{20} functions, where the classic PSO achieves the best result. Interestingly, although CMA-ES largely outperformed the other algorithms in the case of the rotated/shifted Rosenbrock's function, it ranked last in the case of the hybrid Schaffer's F7 f_{20}. We also observe that the performance of DE are good with functions f_9 and f_{20}, while they are limited with the other benchmark functions. EDA was outperformed by the other algorithms on functions f_4, f_{13} and f_{19}, having an ABF orders of magnitude higher than the competitor methods. FST-PSO is characterized by mixed performance, yielding some high quality solution (e.g., f_4, f_{15}, f_{20}) but never outperforming the other algorithms in any of the tested benchmark functions. Finally, both ABC and GA show average performances, always remaining in the middle of the ranking. Interestingly, ABC is characterized by a high variability in the results, as shown by the boxplots of the best solutions found across all runs (see Supplementary file, Figs. 1 and 3): the algorithm seems to yield either poor or very good solutions, probably according to the distribution of the bees at the beginning of the optimization phase. GA, however, seems to be characterized by a slower convergence (see, e.g., functions f_4, f_{13}, f_{15}, and f_{19}) than the other meta-heuristics.

When we consider the PE problem, the results turn out to be totally different, since the version of FST-PSO (no $\mathbf{v}_{min}$) consistently achieves the best results compared to the other meta-heuristics. CMA-ES obtains performance comparable to the other methods only in the case of Model 2 (where DE has the worst performance), and a worse performance (comparable to EDA) in all other cases. Interestingly, ABC tied FST-PSO's performance in the case of Models 2 and 6 and, differently from the case of benchmark functions, it was not characterized by a high variance in the optimal solutions found (see Supplementary file, Figs. 2 and 4), confirming that the algorithm has a different behavior in the case of PE.

We executed further tests to analyze how the performance of the meta-heuristics scale with the number of dimensions of the benchmark functions and the number of parameters to be estimated in the case of the PE problem. In Figs. 3 and 4 we show the results obtained on the benchmark functions with $M = 50$ dimensions and on the RBMs with $M = 50$ kinetic constants to be estimated. Differently from the case of $M = 25$ dimensions, we observe that CMA-ES outperforms the other techniques only in the case of f_4. In the case of f_{19}, the final ABF of CMA-ES after 500 iterations is tied with PSO and FST-PSO, even though the convergence speed is far lower in the case of CMA-ES. Interestingly, the algorithm characterized by the best performance is the classic PSO except in the case of function f_9, contradicting what was observed in the case of $M = 25$. Moreover, taking into account benchmark functions with $M = 50$, EDA is no longer able to identify optimal solutions and seems to be extremely prone to premature convergence, while ABC has a considerably slower convergence speed with respect to the other meta-heuristics.

Again, when considering the PE problem, FST-PSO (no $\mathbf{v}_{min}$) consistently achieves the best results, outperforming the other methods and showing a higher convergence speed in the case of Model 7. CMA-ES consistently holds the worst performance, besides Models 9 and 12. GA seems to scale better, maintaining good or average performance over all 6 models.

An alternative representation of the results presented above is given in Figs. 5, 6 and 7, 8 (for 25 and 50 dimensions of the benchmark functions and parameters to be estimated in the RBMs, respectively), in which we show the Kiviat diagrams (also known as radar or spider charts) [53, 54] obtained by plotting the ABF value achieved by the different meta-heuristics during the last iteration of the optimization processes. Note that the lower the area described by the plot, the better the performance of the meta-heuristic. Since the final fitness values are generally different among the benchmark functions as well as among the PE of different RBMs, we normalized these values in the range $[0, 1]$. We

Table 2: Statistical comparison of the tested algorithms in solving the benchmark functions and the PE problem considering $M = 25$ and $M = 50$, calculated using the ABF values at the last iteration. The second row shows the p-values of the Friedman's test. Since the p-values allow us to reject the null hypothesis, we performed the Bonferroni-Dunn's *post hoc* test obtaining critical differences equal to 0.895 and 0.982 considering 90% and 95% confidence levels, respectively. For each column, the results are expressed as: ranking – group (obtained with 95% confidence level and denoted with Roman numerals). Note that an algorithm can belong to more than one group.

	$M = 25$		$M = 50$	
	Benchmarks	PE	Benchmarks	PE
	p-value=0.0014	p-value=0.00025	p-value=0.00020	p-value=0.00014
ABC	6.000 – III	3.167 – II	5.833 – IV	3.500 – II
CMA-ES	3.000 – I, II	6.333 – IV	4.000 – III	6.500 – IV, V
DE	5.833 – III	5.333 – III	5.667 – IV	3.667 – II
EDA	7.833 – IV	7.500 – V	8.000 – V	7.167 – V
GA	3.833 – II	3.833 – II	4.500 – III	2.000 – I
PSO	2.833 – I	3.833 – II	1.667 – I	5.833 – III, IV
FST-PSO	3.667 – I, II	5.000 – III	3.667 – III	5.500 – III
FST-PSO (no $\mathbf{v}_{min}$)	3.000 – I, II	1.000 – I	2.667 – II	1.833 – I

observe how the performance of CMA-ES drastically decreases when applied to the PE problem, compared with the performance on the benchmark functions. The opposite holds for DE and GA, since these algorithms show better convergence properties in the case of PE with respect to benchmark functions. The performance of FST-PSO is also striking, especially when the fuzzy rules for minimum velocity are disabled. Note that this strategy leads to slightly worse results with the benchmark functions, but extremely good performance in the case of the PE, notably without the need for any functioning setting.

In order to investigate the existence of any statistical differences among the performances of the tested algorithms, we executed the Friedman's test [55] and the Bonferroni-Dunn's *post hoc* test [56]. Table 2 lists the ranks calculated using the ABF values achieved during the last iteration of all tests executed on the benchmark functions and the RBMs considered in this work. Since the p-values of the Friedman's test (reported in the table) allowed us to reject the null hypothesis (i.e., the difference in the performance of the algorithms is not statistically significant), we proceeded with the Bonferroni-Dunn's *post hoc* test to determine which algorithms are significantly better than the others. We thus calculated the critical differences (CDs) with 90% and 95% confidence levels, obtaining CDs equal to 0.895 and 0.982, respectively. Taking into account 95% confidence level, we formed groups of algorithms whose performances are not significantly different (denoted by Roman numerals in Table 2). ABC ranks in the third and fourth groups of algorithms in solving the benchmark functions with $M = 25$ and $M = 50$, respectively, while in the case of the PE problems it is capable of catching up with the second group of algorithms. PSO is characterized by an almost opposite trend; indeed, it ranks among the best algorithms (always in the first group) when solving benchmark functions (resulting the best choice in the case of $M = 50$), whereas it belongs to the second group regarding the PE problem when $M = 25$, and in the third and fourth groups when $M = 50$, showing how its performance decreases while the number of dimensions increases. The results obtained by FST-PSO are strictly comparable with those achieved by PSO, even if it generally performs better (worse) in the PE problem (benchmark functions). Moreover, FST-PSO generally outperforms DE in all cases except for the PE with $M = 50$. Disabling the fuzzy rules for the minimum velocity throttling, the results are quite different. As a matter of fact, FST-PSO (no $\mathbf{v}_{min}$) ranks always among the best algorithms, taking into account both benchmark functions and the PE problem. GA obtains the best results when the number of dimensions increases, becoming highly competitive in solving the PE problem, ranking first together with FST-PSO (no $\mathbf{v}_{min}$) when $M = 50$. It is also competitive in solving benchmark functions, placing in the second and third groups. In this case, the performance of GA decreases as the number of dimensions increases. CMA-ES shows quite good performance with benchmark functions, being always in the first three groups of algorithms, while it is not competitive in the case of the PE problem. Finally, the achieved results highlight that EDA obtains the worst performance considering both the benchmark functions and the PE problem, attaining the last group in all tests.

To summarize, the analysis conducted on the benchmark functions highlighted that the best algorithms are PSO, FST-PSO, FST-PSO (no $\mathbf{v}_{min}$) and CMA-ES. Among them, PSO might be employed due to its simplicity; however, its performance is strongly related to the values of its settings. FST-PSO and its variant FST-PSO (no $\mathbf{v}_{min}$) can overcome this limitation, thus resulting the most suitable algorithms to deal with benchmark functions. Regarding the PE problem, FST-PSO (no $\mathbf{v}_{min}$) is generally capable of outperforming the other tested algorithms, being the best meta-heuristic when $M = 25$ and ranking first together with GA when $M = 50$.

3.2. Semantic logarithmic transformation of kinetic parameters

Since both CMA-ES and EDA exploit normal distributions to generate new individuals, their performances could be affected by the peculiar log-uniform distribution of kinetic parameters [52]. In order to investigate this conjecture, we modified CMA-ES, EDA, and FST-PSO to change the semantics of the parameters to a logarithmic scale. Specifically, the putative parameters were bounded in the interval $(0, 1)$ and each value k_m was converted to the actual kinetic parameter k'_m—used for the fitness evaluation—by means of the following transformation:

$$\begin{aligned} \phi_m &= \log_{10}(\beta_m^{max}) + \left(\log_{10}(\beta_m^{min}) - \log_{10}(\beta_m^{max})\right) k_m, \\ k'_m &= 10^{\phi_m}. \end{aligned} \tag{3}$$

Note that the boundaries are mapped into the interval $(0, 1)$ during the initialization phase (for each run) of the algorithm under investigation, whilst Eq. (3) is applied at each iteration. We denote by CMA-ES-log, EDA-log, and FST-PSO-log the three modified algorithms. We show in Fig. 9 a comparison of the performances of the three modified algorithms (solid lines) with respect to the original methods (dashed lines). The test was carried out on Model 10, in which both CMA-ES and EDA showed, by far, the worst performances.

According to our results, the performance of CMA-ES-log is radically different from classic CMA-ES, with a final ABF very close to zero and an extremely quick convergence. The performance of EDA-log (whose σ_0^{EDA} was set to 0.1

because of the modified search space) strongly improved with respect to classic EDA; however, it was repeatedly unable to converge to an optimal solution, keeping the final ABF above 50. Even though CMA-ES-log was able to rapidly converge, the result achieved by FST-PSO-log is even better and highlights how the logarithmic semantics can help *all* algorithms for the PE problem. This circumstance further reveals that benchmark functions cannot capture the intrinsic complexity of biochemical PE.

In order to corroborate our findings, we performed the PE of an extended version of the model of the human intracellular core metabolic pathways in red blood cells, presented in [57]. In particular, we explicitly consider three main isoforms of the hexokinase enzyme, obtaining a RBM characterized by 78 new reactions whose kinetic constants are unknown (see [58] for additional details).

The results shown in Fig. 10 confirm that the standard CMA-ES version (yellow solid line) is not capable of achieving good optimization results in terms of ABF. On the contrary, thanks to the transformation in Eq. (3), CMA-ES (yellow dashed line) is able to achieve results similar to FST-PSO (magenta lines), in accordance with the general patterns observed with the tested synthetic models. In addition, it is worth noting that CMA-ES-log, on average, begins the optimization with a better initial distribution with respect to FST-PSO: indeed, the ABF in the case of CMA-ES-log at iteration 0 is approximately 25, while in the case of FST-PSO is approximately 32. This result highlights a further advantage of our alternative representation of parameters.

4. Conclusions

In this paper we presented an analysis of the performance of some state-of-the-art meta-heuristics (i.e., ABC, CMA-ES, DE, EDA, GA, PSO, and FST-PSO) applied to a set of well-known benchmark functions and a real-world problem related to the estimation of kinetic parameters of biochemical systems. Concerning the PE problem, we exploited a set of *in silico* generated models of increasing size (i.e., 25 and 50 molecular species/reactions), and for each size we considered 6 different models, for a total of 12 test cases.

Overall, the results achieved in this work point out that the performance of the meta-heuristics can drastically change according to the context of application, showing that the fitness landscapes identified by classic benchmark functions are completely different from those characterizing the PE problem. We argue that a novel set of benchmark functions, designed to mimic the characteristics of real-world problems, is necessary to achieve a better understanding and a thorough evaluation of the performance of the meta-heuristics. These benchmark functions should be defined attempting to resemble the fitness landscapes of a variety of real-world problems. Although some preliminary efforts were devoted to create functions similar to the PE problem [45], we are still far from a complete and reliable reproduction of its intrinsic characteristics. In principle, real-world problems should be applied for benchmarking, since they provide a valuable contribution to experimental research practice [33]. Differently from benchmark functions, the structural features underlying real-world optimization problems are often not well characterized [59]; thus, additional research must be performed to understand how novel benchmark functions could be designed to replicate their peculiarities. As a matter of fact, defining benchmark functions inspired by real-world problems is not trivial, since it requires the preliminary design and development of novel *ad hoc* methods to analyze and classify optimization problems, as well as automatic methods (by using, e.g., Genetic Programming [60] or hierarchical fitness assignment methods based on statistical tests [15, 61]) to devise arbitrary functions characterized by analogous fitness-space features.

The results of our tests highlighted that CMA-ES is one of the best choices for the optimization of benchmark functions, but its performance turned out to be worse than most of the other meta-heuristics when applied to the PE of biochemical systems in 10 out of 12 RBMs. Since both CMA-ES and EDA exploit normal distributions to generate new individuals, their performances are probably affected by the peculiar log-uniform distribution of kinetic parameters [52]. We empirically proved this conjecture by repeating the PE tasks using a logarithmic semantics for the putative parameters, showing that all algorithms benefit from this solution and, in particular, CMA-ES was now able to efficiently converge to high quality solutions. This conjecture was corroborated by the results obtained from the PE of a large-scale model of metabolic pathways in red blood cells. Since the performance of the tested meta-heuristics is affected by the semantic logarithmic transformation applied to the original space, novel hybrid functions, based on basic functions whose global optima span different orders of magnitude, should be proposed. These novel functions should allow for better resembling the fitness landscape of real-world problems, such as the biochemical PE. We will further investigate the logarithmic exploration of the parameter space, a topic that we previously tackled by considering also the population initialization [45] and particles' reboot [52] in PSO. We argue that the performance of some algorithms in specific real-world problems can be strongly improved by transforming, or adapting, the representation of the solutions. Although it was possible for us to define an effective transformation in the case of PE, this task is generally not straightforward to perform. In particular, we speculate that the automatic design of the optimal transformation for any problem might be as difficult as solving the optimization problem itself. Due to its relevance in the context of optimization problems, we plan to investigate this topic in the near future.

Finally, we observed that the version of FST-PSO where fuzzy rules for the minimum velocity throttling are disabled (i.e., not leveraging turbulence [29]) appears to be the best choice for PE, although its convergence speed in the case of the benchmark functions is worse than classic PSO. Anyway, PSO requires the selection of multiple functioning settings, which is not necessary in the case of FST-PSO. As a further extension of this work, we will define improved alternative fuzzy rules (or approaches) to automatically set the minimum velocity, in order to define a completely multi-purpose methodology effective both in the case of benchmark functions and real-world problems.

Acknowledgments

This work was conducted in part using the resources of the Advanced Computing Center for Research and Education at Vanderbilt University, Nashville, TN, USA.

References

[1] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, Cambridge, MA, USA, 1992.

[2] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359. doi:10.1023/A:1008202821328.

[3] N. Hansen, The CMA evolution strategy: a comparing review, in: J. Lozano, P. Larrañaga, I. Inza, E. Bengoetxea (Eds.), Towards a new Evolutionary Computation, Springer, 2006, pp. 75–102. doi:10.1007/3-540-32494-1_4.

[4] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (3) (2007) 459–471. doi:10.1007/s10898-007-9149-x.

[5] J. Kennedy, R. Eberhart, Particle swarm optimization, Vol. 4 of Proc. International Conference Neural Networks (ICNN), IEEE, 1995, pp. 1942–1948. doi:10.1109/ICNN.1995.488968.

[6] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimization problems, Int. J. Math. Model. Num. Opt. 4 (2) (2013) 150–194. doi:10.1504/IJMMNO.2013.055204.

[7] J. M. Dieterich, B. Hartke, Empirical review of standard benchmark functions using evolutionary global optimization, Appl. Math. 3 (2012) 1552–1564. doi:10.4236/am.2012.330215.

[8] J. H. Kämpf, M. Wetter, D. Robinson, A comparison of global optimization algorithms with standard benchmark functions and real-world applications using EnergyPlus, J. Build. Perf. Sim. 3 (2) (2010) 103–120. doi:10.1080/19401490903494597.

[9] S. E. Cox, R. T. Haftka, C. A. Baker, B. Grossman, W. H. Mason, L. T. Watson, A comparison of global optimization methods for the design of a high-speed civil transport, J. Glob. Optim. 21 (4) (2001) 415–432. doi:10.1023/A:1012782825166.

[10] A. J. Lockett, R. Miikkulainen, A probabilistic reformulation of no free lunch: Continuous lunches are not free, Evol. Comput. 25 (3) (2017) 503–528. doi:10.1162/evco_a_00196.

[11] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82. doi:10.1109/4235.585893.

[12] W. G. Macready, D. H. Wolpert, What makes an optimization problem hard?, Complexity 1 (5) (1996) 40–46. doi:10.1002/cplx.6130010511.

[13] A. Auger, O. Teytaud, Continuous lunches are free!, in: Proc. Genetic and Evolutionary Computation Conference (GECCO), ACM, 2007, pp. 916–922. doi:10.1145/1276958.1277145.

[14] A. Auger, O. Teytaud, Continuous lunches are free plus the design of optimal optimization algorithms, Algorithmica 57 (1) (2010) 121–146. doi:10.1007/s00453-008-9244-5.

[15] Y. Lou, S. Y. Yuen, On constructing alternative benchmark suite for evolutionary algorithms, Swarm Evol. Comput.doi:10.1016/j.swevo.2018.04.005.

[16] S. Da Ros, G. Colusso, T. A. Weschenfelder, L. de Marsillac Terra, F. De Castilhos, M. L. Corazza, M. Schwaab, A comparison among

stochastic optimization algorithms for parameter estimation of biochemical kinetic models, Appl. Soft Comput. 13 (5) (2013) 2205–2214. doi:10.1016/j.asoc.2013.01.019.

[17] C. Moles, P. Mendes, J. Banga, Parameter estimation in biochemical pathways: a comparison of global optimization methods, Genome Res. 13 (11) (2003) 2467–2474. doi:10.1101/gr.1262503.

[18] A. Dräger, M. Kronfeld, M. Ziller, J. Supper, H. Planatscher, J. Magnus, Modeling metabolic networks in *C. glutamicum*: a comparison of rate laws in combination with various parameter optimization strategies, BMC Syst. Biol. 3 (2009) 5. doi:10.1186/1752-0509-3-5.

[19] H. Kitano, Systems biology: a brief overview, Science 295 (5560) (2002) 1662–1664. doi:10.1126/science.1069492.

[20] P. Cazzaniga, C. Damiani, D. Besozzi, R. Colombo, M. S. Nobile, D. Gaglio, D. Pescini, S. Molinari, G. Mauri, L. Alberghina, et al., Computational strategies for a system-level understanding of metabolism, Metabolites 4 (4) (2014) 1034–1087. doi:10.3390/metabo4041034.

[21] N. Hansen, A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, Proc. International Conference on Evolutionary Computation (CEC), IEEE, 1996, pp. 312–317. doi:10.1109/ICEC.1996.542381.

[22] S. Das, P. N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31. doi:10.1109/TEVC.2010.2059031.

[23] P. Larrañaga, J. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Vol. 2 of Genetic Algorithms and Evolutionary Computation, Springer Science & Business Media, New Your, NY, USA, 2001.

[24] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, G. Pasi, Fuzzy self-tuning PSO: A settings-free algorithm for global optimization, Swarm Evol. Comput. 39 (2018) 70–85. doi:10.1016/j.swevo.2017.09.001.

[25] M. S. Nobile, A. Tangherloni, L. Rundo, S. Spolaor, D. Besozzi, G. Mauri, P. Cazzaniga, Computational intelligence for parameter estimation of biochemical systems, in: Proc. IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018, pp. 1–8. doi:10.1109/CEC.2018.8477873.

[26] M. Hauschild, M. Pelikan, An introduction and survey of estimation of distribution algorithms, Swarm Evol. Comput. 1 (3) (2011) 111–128. doi:10.1016/j.swevo.2011.08.003.

[27] M. Sebag, A. Ducoulombier, Extending population-based incremental learning to continuous search spaces, in: A. E. Eiben, T. Bäck, M. Schoenauer, H. P. Schwefel (Eds.), Proc. International Conference on Parallel Problem Solving from Nature (PPSN), Vol. 1498 of LNCS, Springer, 1998, pp. 418–427. doi:10.1007/BFb0056884.

[28] K. Harrison, A. Engelbrecht, B. Ombuki-Berman, Inertia weight control strategies for particle swarm optimization, Swarm Intell. 10 (4) (2016) 267–305. doi:10.1007/s11721-016-0128-z.

[29] A. Abraham, H. Liu, Turbulent particle swarm optimization using fuzzy parameter tuning, in: A. Abraham, A. Hassanien, P. Siarry, A. Engelbrecht (Eds.), Foundations of Computational Intelligence Volume 3, Vol. 203 of Studies in Computational Intelligence, Springer, Berlin, Germany, 2009, pp. 291–312. doi:10.1007/978-3-642-01085-9_10.

[30] Y. Shi, R. Eberhart, Fuzzy adaptive particle swarm optimization, Vol. 1 of Proc. Congress on Evolutionary Computation (CEC), IEEE, 2001, pp. 101–106. doi:10.1109/CEC.2001.934377.

[31] M. S. Nobile, G. Pasi, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, Proactive particles in swarm optimization: a self-tuning algorithm based on fuzzy logic, Proc. International Conference on Fuzzy Systems, IEEE, 2015, pp. 1–8. doi:10.1109/FUZZ-IEEE.2015.7337957.

[32] A. Tangherloni, L. Rundo, M. S. Nobile, Proactive Particles in Swarm Optimization: a settings-free algorithm for real-parameter single objective optimization problems, Proc. Congress on Evolutionary Computation (CEC), IEEE, 2017, pp. 1940–1947. doi:10.1109/CEC.2017.7969538.

[33] M. Gallagher, Towards improved benchmarking of black-box optimization algorithms using clustering problems, Soft Comput. 20 (10) (2016) 3835–3849. doi:10.1007/s00500-016-2094-1.

[34] Congress on Evolutionary Computation (CEC), Special Session & Competitions on Real-Parameter Single Objective Optimization, http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2017/CEC2017.htm, online; Accessed on November 17, 2018 (2017).

[35] Genetic and Evolutionary Computation Conference (GECCO), GECCO Workshop on Real-Parameter Black-Box Optimization Benchmarking (BBOB), http://numbbo.github.io/workshops/BBOB-2018/, online; Accessed on November 17, 2018 (2018).

[36] N. Hansen, A. Auger, O. Mersmann, T. Tusar, D. Brockhoff, COCO: A platform for comparing continuous optimizers in a black-box setting, arXiv preprint arXiv:1603.08785.

[37] C. Doerr, F. Ye, S. van Rijn, H. Wang, T. Bäck, Towards a theory-guided benchmarking suite for discrete black-box optimization heuristics: profiling (1+ λ) ea variants on onemax and leadingones, in: Proc. Genetic and Evolutionary Computation Conference (GECCO), ACM, New York, NY, USA, 2018, pp. 951–958. doi:10.1145/3205455.3205621.

[38] C. Li, S. Yang, A generalized approach to construct benchmark problems for dynamic optimization, in: X. Li, et al. (Eds.), Simulated Evolution and Learning (SEAL), LNCS, Springer, 2008, pp. 391–400. doi:10.1007/978-3-540-89694-4_40.

[39] C. Li, T. T. Nguyen, S. Zeng, M. Yang, M. Wu, An open framework for constructing continuous optimization problems, IEEE Trans. Cybern. (99) (2018) 1–15. doi:10.1109/TCYB.2018.2825343.

[40] N. Awad, M. Ali, J. Liang, B. Qu, P. N. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization, Tech. rep., http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/CEC-2017/Bound-Constrained/Definitions%20of%20%20CEC2017%20benchmark%20suite%20final%20version%20updated.pdf (2016).

[41] D. Besozzi, Reaction-based models of biochemical networks, in: A. Beckmann, L. Bienvenu, N. Jonoska (Eds.), Pursuit of the Universal. Proc. 12th Conference on Computability in Europe (CiE), no. 9709 in LNCS, Springer International Publishing, 2016, pp. 24–34. doi:10.1007/978-3-319-40189-8_3.

[42] V. Chellaboina, S. Bhat, W. Haddad, D. Bernstein, Modeling and analysis of mass-action kinetics, IEEE Control Syst. Mag. 29 (4) (2009) 60–78. doi:10.1109/MCS.2009.932926.

[43] A. De Myttenaere, B. Golden, B. Le Grand, F. Rossi, Mean absolute percentage error for regression models, Neurocomputing 192 (2016) 38–48. doi:10.1016/j.neucom.2015.12.114.

[44] H. Yue, M. Brown, F. He, J. Jia, D. Kell, Sensitivity analysis and robust experimental design of a signal transduction pathway system, Int. J. Chem. Kinet. 40 (11) (2008) 730–741. doi:10.1002/kin.20369.

[45] P. Cazzaniga, M. S. Nobile, D. Besozzi, The impact of particles initialization in PSO: parameter estimation as a case in point, Proc. Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), IEEE, 2015, pp. 1–8. doi:10.1109/CIBCB.2015.7300288.

[46] M. S. Nobile, D. Besozzi, P. Cazzaniga, G. Mauri, GPU-accelerated simulations of mass-action kinetics models with cupSODA, J. Supercomput. 69 (1) (2014) 17–24. doi:10.1007/s11227-014-1208-8.

[47] M. S. Nobile, D. Besozzi, P. Cazzaniga, G. Mauri, D. Pescini, cupSODA: a CUDA-powered simulator of mass-action kinetics, in: V. Malyshkin (Ed.), Proc. International Conference on Parallel Computing Technologies, no. 7979 in LNCS, Springer, 2013, pp. 344–357. doi:10.1007/978-3-642-39958-9_32.

[48] E. O. Voit, H. A. Martens, S. W. Omholt, 150 years of the mass action law, PLoS Comput. Biol. 11 (1) (2015) e1004012. doi:10.1371/journal.pcbi.1004012.

[49] F. Fortin, F. De Rainville, M. Gardner, M. Parizeau, C. Gagné, DEAP: Evolutionary algorithms made easy, J. Mach. Learn. Res. 13 (2012) 2171–2175.

[50] F. Van den Bergh, A. P. Engelbrecht, A study of particle swarm optimization particle trajectories, Inform. Sci. 176 (8) (2006) 937–971. doi:10.1016/j.ins.2005.02.003.

[51] I. C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Inf. Process. Lett. 85 (6) (2003) 317–325. doi:10.1016/S0020-0190(02)00447-7.

[52] S. Spolaor, A. Tangherloni, L. Rundo, M. S. Nobile, P. Cazzaniga, Reboot strategies in particle swarm optimization and their impact on parameter estimation of biochemical systems, Proc. Conference on Computational In-

telligence in Bioinformatics and Computational Biology (CIBCB), IEEE, 2017, pp. 1–8. doi:10.1109/CIBCB.2017.8058550.

[53] K. W. Kolence, P. J. Kiviat, Software unit profiles & Kiviat figures, ACM SIGMETRICS Perform. Eval. Rev. 2 (3) (1973) 2–12. doi:10.1145/1041613.1041614.

[54] R. W. Elliott, Kiviat-graphs as a means for displaying performance data for on-line retrieval systems, J. Am. Soc. Inform. Sci. 27 (3) (1976) 178. doi:10.1002/asi.4630270309.

[55] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, J. Heuristics 15 (6) (2009) 617–644. doi:10.1007/s10732-008-9080-4.

[56] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.

[57] N. Jamshidi, B. Ø. Palsson, Mass action stoichiometric simulation models: incorporating kinetics and regulation into stoichiometric models, Biophys. J. 98 (2) (2010) 175–185. doi:10.1016/j.bpj.2009.09.064.

[58] N. Totis, A. Tangherloni, M. Beccuti, P. Cazzaniga, M. S. Nobile, D. Besozzi, M. Pennisi, F. Pappalardo, Efficient and settings-free calibration of detailed kinetic metabolic models with enzyme isoforms characterization, in: Proc. Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB 2018), LNBI, Springer, Submitted.

[59] J. J. Liang, P. N. Suganthan, K. Deb, Novel composition test functions for numerical global optimization, in: Proc. Swarm Intelligence Symposium (SIS), IEEE, 2005, pp. 68–75. doi:10.1109/SIS.2005.1501604.

[60] W. B. Langdon, R. Poli, Evolving problems to learn about particle swarm optimizers and other search algorithms, IEEE Trans. Evol. Comput. 11 (5) (2007) 561–578. doi:10.1109/TEVC.2006.886448.

[61] Y. Lou, S. Y. Yuen, G. Chen, Evolving benchmark functions using Kruskal-Wallis test, in: Proc. Genetic and Evolutionary Computation Conference Companion (GECCO), ACM, 2018, pp. 1337–1341. doi:10.1145/3205651.3208257.

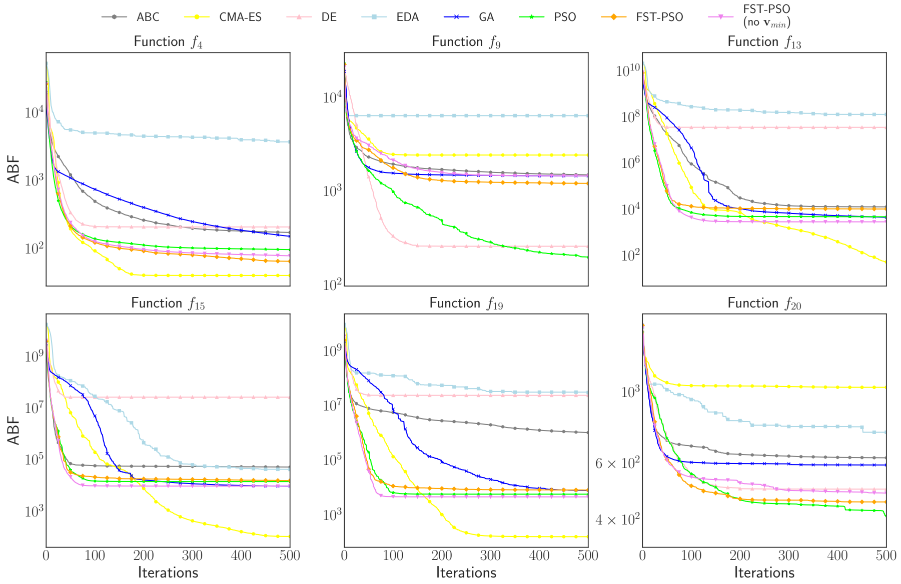

Figure 1: Comparison of the performances in terms of ABF achieved by the meta-heuristics on the benchmark functions with $M = 25$.

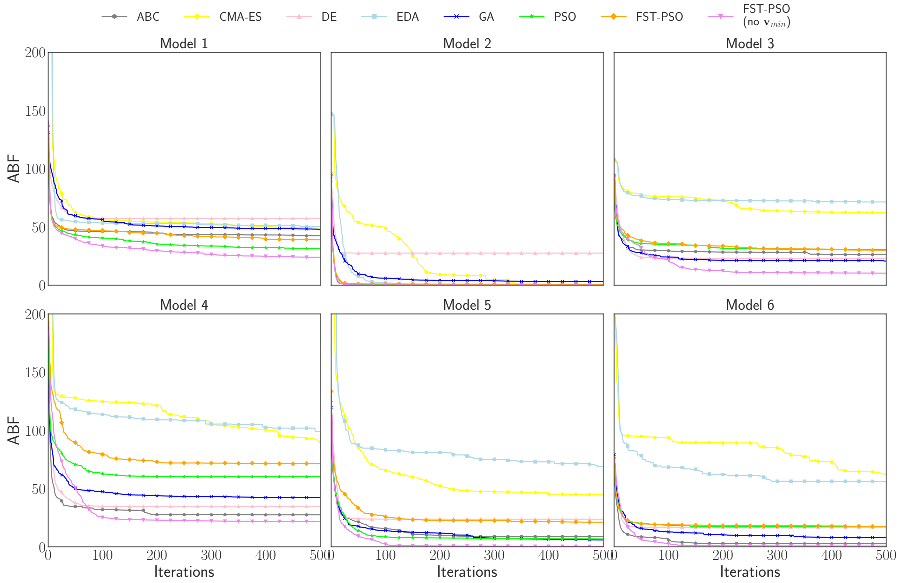

Figure 2: Comparison of the performances in terms of ABF achieved by the meta-heuristics for the PE of synthetic models characterized by 25 reactions and 25 molecular species.

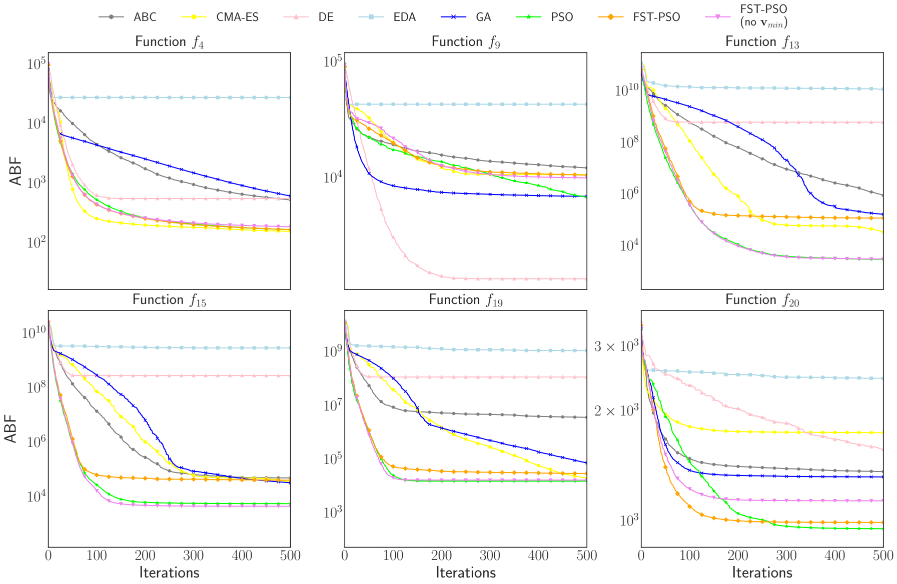

Figure 3: Comparison of the performances in terms of ABF achieved by the meta-heuristics on the benchmark functions with $M = 50$.

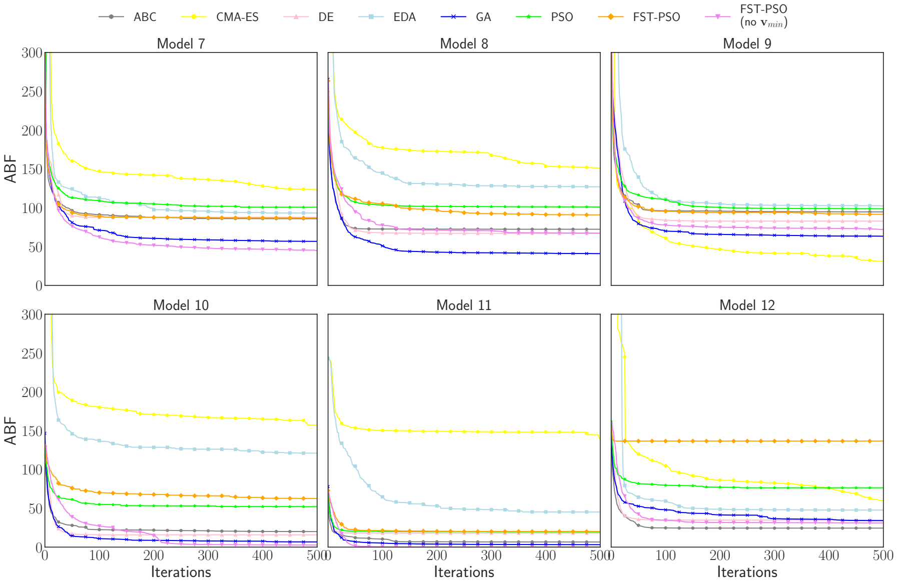

Figure 4: Comparison of the performances in terms of ABF achieved by the meta-heuristics for the PE of synthetic models characterized by 50 reactions and 50 molecular species.

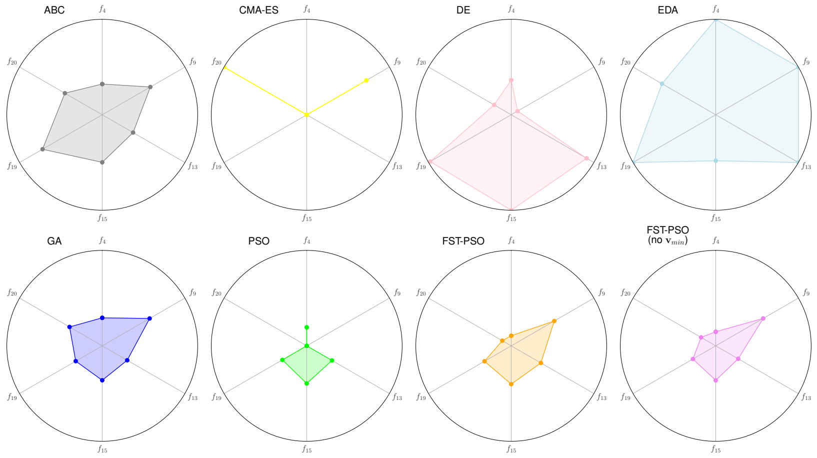

Figure 5: Kiviat diagram showing the final ABF value obtained by the meta-heuristics on the benchmark functions with $M = 25$.

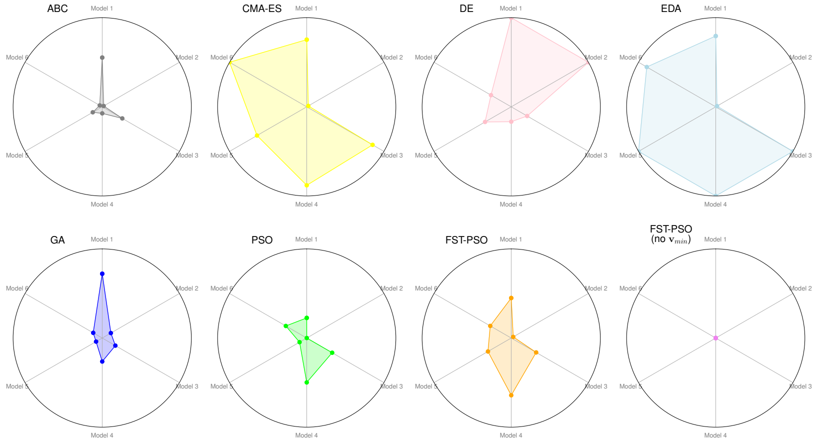

Figure 6: Kiviat diagram showing the final ABF value obtained by the meta-heuristics in the PE of synthetic models characterized by 25 reactions and 25 molecular species.

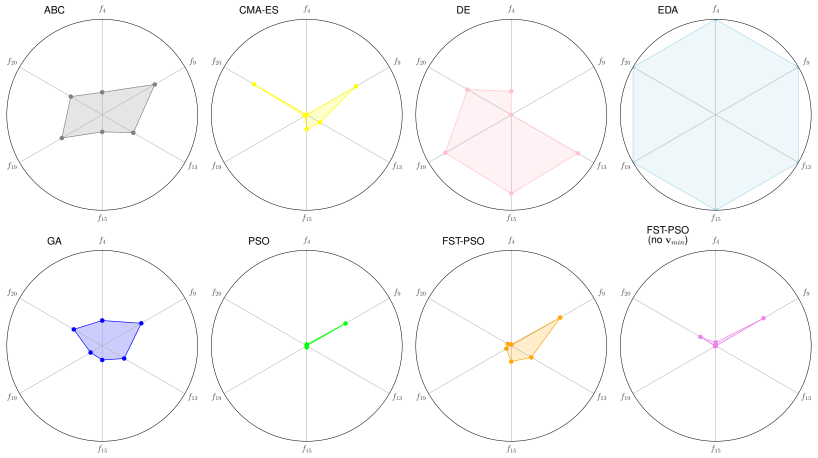

Figure 7: Kiviat diagram showing the final ABF value obtained by the meta-heuristics on the benchmark functions with $M = 50$.

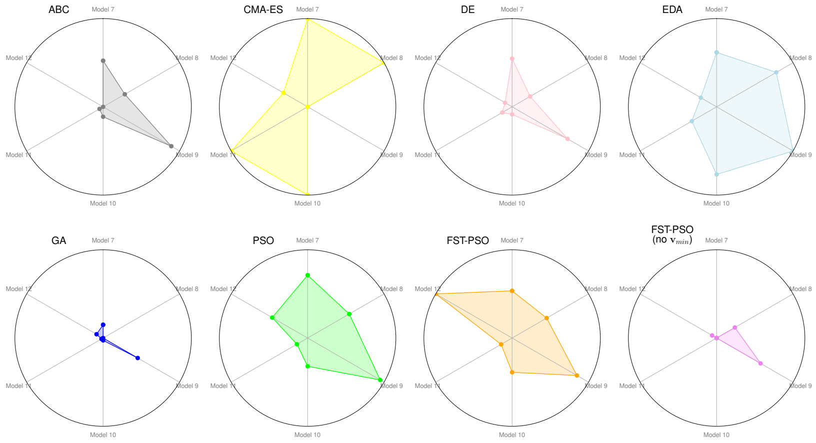

Figure 8: Kiviat diagram showing the final ABF value obtained by the meta-heuristics in the PE of synthetic models characterized by 50 reactions and 50 molecular species.

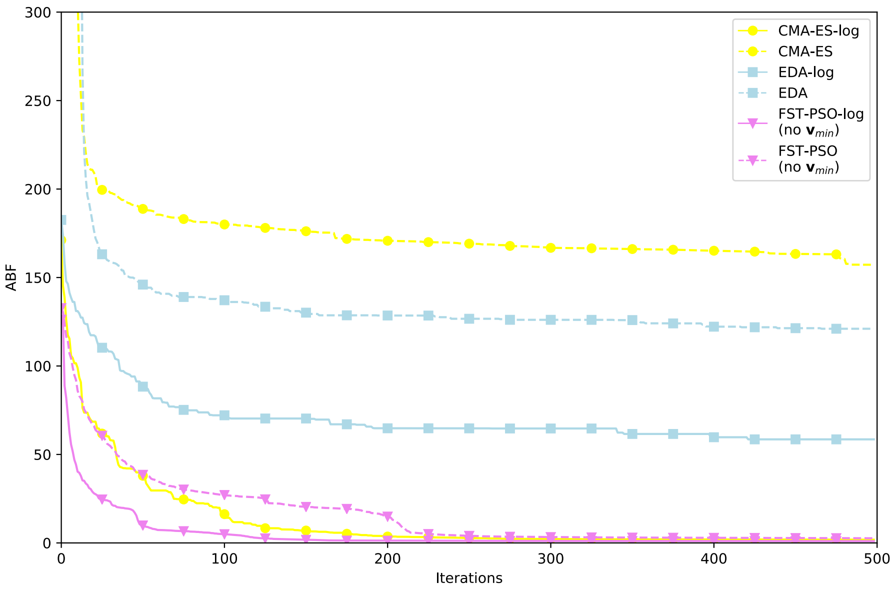

Figure 9: Comparison of the performances of CMA-ES, EDA and FST-PSO (no $\mathbf{v}_{min}$) with normal and logarithmic semantics of parameters, for the parameter estimation of Model 10.

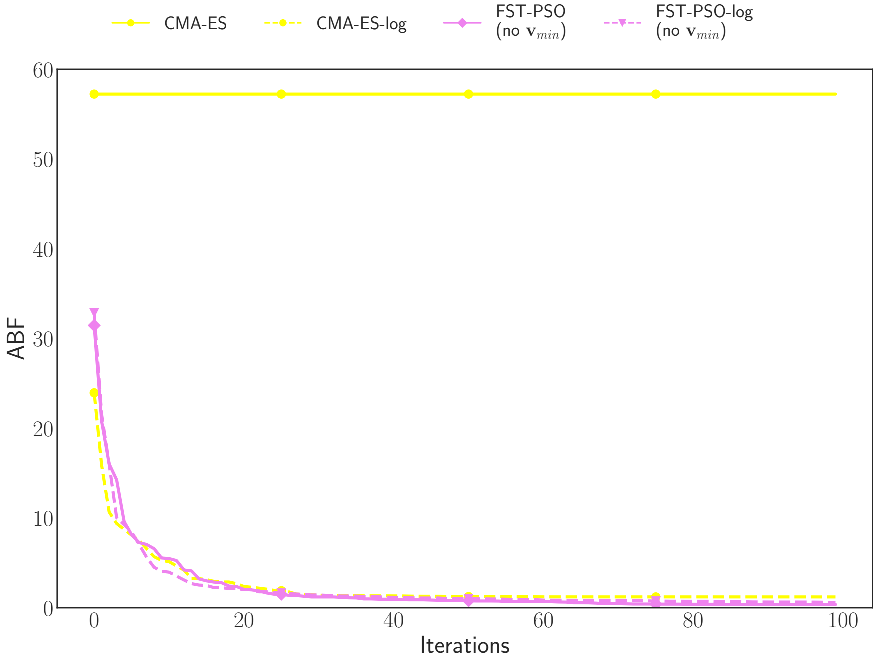

Figure 10: Comparison of the performances of CMA-ES and FST-PSO (no $\mathbf{v}_{min}$) with normal and logarithmic semantics of parameters, for the estimation of the 78 kinetic parameters of the human intracellular core metabolic pathways model.