Hindawi
Journal of Optimization
Volume 2018, Article ID 3213484, 21 pages
https://doi.org/10.1155/2018/3213484

Research Article

On Metaheuristics for Solving the Parameter Estimation Problem in Dynamic Systems: A Comparative Study

Gisela C. V. Ramadas,[1] **Edite M. G. P. Fernandes**,[2] **António M. V. Ramadas**,[3] **Ana Maria A. C. Rocha**,[2] **and M. Fernanda P. Costa**[4]

[1]*Research Center of Mechanical Engineering (CIDEM), School of Engineering, Polytechnic of Porto, 4200-072 Porto, Portugal*
[2]*ALGORITMI Research Centre, University of Minho, 4710-057 Braga, Portugal*
[3]*Faculty of Engineering, University of Porto, 4200-465 Porto, Portugal*
[4]*Centre of Mathematics, University of Minho, 4710-057 Braga, Portugal*

Correspondence should be addressed to Gisela C. V. Ramadas; gcv@isep.ipp.pt

Received 16 July 2017; Revised 11 December 2017; Accepted 31 December 2017; Published 29 January 2018

Academic Editor: Liwei Zhang

This paper presents an experimental study that aims to compare the practical performance of well-known metaheuristics for solving the parameter estimation problem in a dynamic systems context. The metaheuristics produce good quality approximations to the global solution of a finite small-dimensional nonlinear programming problem that emerges from the application of the sequential numerical direct method to the parameter estimation problem. Using statistical hypotheses testing, significant differences in the performance of the metaheuristics, in terms of the average objective function values and average CPU time, are determined. Furthermore, the best obtained solutions are graphically compared in relative terms by means of the performance profiles. The numerical comparisons with other results in the literature show that the tested metaheuristics are effective in achieving good quality solutions with a reduced computational effort.

1. Introduction

The present paper addresses the problem of finding a set of parameter values in a dynamic system model that calibrates the model so that it can reproduce the existing experimental data in the best possible way [1]. This is performed by minimizing an objective function that measures the goodness of the fit. Thus, an objective function, which gives the sum of the squared errors between the model predicted state values and the observed values (at certain time instants of a fixed interval), is minimized, subject to a system of ordinary differential equations (ODE). The dynamic model defined by the system of ODE simulates the time varying processes that take place in a certain time interval. This problem is known in the literature as the dynamic model based parameter estimation (DMbPE) process. Solving this problem is crucial in systems biology and medicine, with a great impact on both pharmaceutical and biotechnological industries [2]. The problem is also very common in the chemical engineering area [3] and it has been extensively used to describe physical phenomena [4]. The DMbPE problem may involve nonlinear differential-algebraic equations and a multiplicity of local solutions may exist. Nonconvexity and ill-conditioning issues of the problem may be addressed with efficient global optimization (GO) methods and proper regularization techniques [5]. The problem is even more difficult than that of an algebraic model. The nonlinear dynamic behavior makes the analytical approach rather complicated, if not impossible, for most of the real phenomena. Numerical direct methods are therefore good alternatives to solve the DMbPE problem. Like any parameter estimation problem, the main advantage is that the intended global minimum is known in advance. The most well-known methods, like Levenberg-Marquardt, gradient descent, and the Nelder-Mead method, which have been extensively used in parameter estimation problems, are local optimization methods and cannot guarantee convergence to a global solution. They are able to exploit a specified search

region around a given initial value, but the global solution is difficult to find when the initial value is far from the required solution. Algorithms for GO can be roughly divided into two categories: exact algorithms and heuristics [6].

Sörensen and Glover [7] have defined metaheuristic as follows: "A metaheuristic is a high-level problem-independent algorithmic *framework* that provides a set of guidelines or strategies to develop heuristic optimization algorithms." Further, an implementation of a heuristic optimization algorithm that follows the strategies laid out in the metaheuristic framework is also referred to as a metaheuristic. This term appeared for the first time in a publication by Glover [8]. Metaheuristics are approximate methods or heuristics that are designed to search for good solutions with less computational effort and time than the more classical algorithms. While heuristics are designed to solve a specific problem, metaheuristics are general-purpose algorithms that can be applied to solve almost any optimization problem. The metaheuristics evaluate a set of potential solutions of an optimization problem and perform a series of operations on those solutions in order to find different and hopefully better solutions. Depending on the way these operations are carried out, a metaheuristic can be classified as a local search, constructive, or population-based metaheuristic. For details, the reader is referred to [7].

Most metaheuristics use random procedures that invoke artificial intelligence tools and simulate nature behaviors and their performance does not depend on the properties of the problem to be solved. They are alternative methods to find good approximations to optimal solutions of GO problems. In addition, they are derivative-free and easy to implement [9]. Metaheuristics can locate the vicinity of global solutions with relative efficiency although global optimality cannot be guaranteed in a finite number of iterations. On the other hand, exact methods for GO can guarantee global optimality for certain problems but the required computational effort increases often exponentially with the problem size [10, 11].

Previous use of metaheuristics, mainly those that use metaphors based on natural evolution and on behavior of animal swarms, to solve some DMbPE problems showed that they are able to provide good quality solutions when real experimental data and noisy artificial data are considered. In [15], a hybrid metaheuristic algorithm that introduces evolutionary operations, namely, mutation and crossover, into the firefly algorithm (FA), is presented. Two variants of the differential evolution (DE) method—a trigonometric and a modified version—have been implemented in [16, 17]. The authors in [2] developed a new procedure based on the scatter search (SS) methodology. A slightly different version of the SS, which uses the `fmincon` solver from Matlab (Matlab is a registered trademark of MathWorks, Inc.) as an improvement method, is tested in [12] to solve three well-known DMbPE problems, including the chemical isomerization of α-pinene. This latter problem is analyzed and solved by the FA in [13]. Alternatively, exact methods for GO have also been used to solve DMbPE problems. These include the spatial branch and bound (BB) algorithm to compute a global solution of the partially discretized DMbPE problem [18], the α-BB method to obtain a global solution within an ϵ-precision of the completely discretized DMbPE problem [10], and a method based on interval analysis and on the partially discretized DMbPE [19]. In [14], a deterministic outer approximation approach is applied to the reformulation of the DMbPE problem as a finite NLP by applying a complete discretization using orthogonal collocation on finite elements.

Since the problem of estimating the parameters of a dynamic model is important, the contribution of this study is concerned with the implementation and practical comparison of five very simple and easy to implement metaheuristics, hybridized with a local intensification phase, when solving the finite nonlinear programming (NLP) problem that arises when a numerical direct method of a sequential type is used to locate a global optimal solution to the DMbPE problem. The unknown parameters are the decision variables of the NLP problem.

The selected metaheuristics are very popular and have been used to solve a variety of real-world applications. The selection includes the FA [20], the Harmony Search (HS) algorithm [21], the DE algorithm [22], the Artificial Bee Colony (ABC) algorithm [23], and the Directed Tabu Search (DTS) algorithm [24]. During the local intensification phase, the well-known Hooke-and-Jeeves (HJ) local search [25] is implemented with all the metaheuristics. These local search enhancements lead to much faster convergence and provide good quality solutions. The practical comparison carried out in the present study aims to analyze the performance of each metaheuristic in terms of the quality of the obtained solutions. To test the five metaheuristics, nine DMbPE problems were selected from the literature [4, 10, 13, 19, 26], yielding 12 instances due to different experimental data. The problems are described in the Appendix.

The remainder of the paper proceeds as follows. Section 2 addresses the DMbPE problem and the sequential numerical direct method, and Section 3 introduces the selected metaheuristics. Section 4 contains the results of all the numerical experiments and the conclusions are summarized in Section 5.

2. Sequential Direct Method to Solve the DMbPE Problem

To solve the DMbPE problem, the sequential numerical direct method is used. For completeness, the DMbPE problem reads as follows. Find p^* such that

$$\begin{aligned} J(p^*) = \min_{p} \quad & J(p) \equiv \sum_{j=1}^{D} \sum_{i=1}^{O} \left(y_j(t_i) - y_{j,i}^{\text{obs}} \right)^2 \\ \text{subject to} \quad & y'(t) = g(t, y(t), p) \quad \forall t \in [t_0, t_f] \\ & y(t_0) = (y_{10}, \ldots, y_{D0}) \\ & p^{\text{L}} \le p \le p^{\text{U}}, \end{aligned} \tag{1}$$

where $J : \mathbb{R}^n \to \mathbb{R}$ represents the objective function that depends on $p = (p_1, p_2, \ldots, p_n)$, a vector of the n decision variables (the parameters that need to be estimated); $y = (y_1, y_2, \ldots, y_D)$ is the vector (with length D) of the state variables that depend on t (the independent variable);

and y' represents the vector with the derivatives of y with respect to t. O represents the number of experimental data available for each y_j; $y_{j,i}^{\text{obs}}$, $j = 1, \ldots, D$, $i = 1, \ldots, O$, are the experimentally observed values; $y' = g(t, y, p)$ is the system of first-order ODE; $(y_{10}, \ldots, y_{D0})$ is the vector of initial values to the variables y_j, $j = 1, \ldots, D$; and p^{L} and p^{U} represent the vectors with the lower and upper bounds on the parameters, respectively. t_0 and t_f are the initial and final values, respectively, of the independent variable t.

To solve problem (1), there are indirect methods and direct methods [27]. Indirect methods use the first-order necessary conditions from Pontryagin's minimum principle to reformulate the problem as a two-point boundary value problem. The latter may become difficult to solve especially if the problem contains state variable constraints.

In direct methods, the optimization present in (1) is performed directly. This type of method discretizes the problem and applies NLP techniques to the resulting finite-dimensional optimization problem. There are two types of direct methods. In the *sequential direct method*, the DMbPE problem is transcribed into a small finite-dimensional optimization problem through the discretization of the decision variables only. Hence, the optimization is carried out in the space of the decision variables only. Given a set of values for the decision variables, the system of ODE is accurately integrated (over the entire time interval) using a specific numerical integration formula (with error control mechanisms to enforce the accuracy of the state variable values), so that the objective function value can be evaluated [3]. Thus, the ODE are satisfied at each iteration of the NLP solver. The method is called sequential, since the processes of minimizing the sum of the squared errors and solving the ODE are done in a sequential manner. This method may lead to a slow convergence process since the system of ODE is solved again and again each time an objective function value is required. In the *simultaneous direct method*, approximations to the solution of the system of ODE are computed instead. The optimization is carried out in the full space of the discretized decision and state variables, which may lead to very-large-dimensional finite NLP problem with equality constraints (emerging from the discretization of the system of ODE), in particular if a fine grid of points is required to obtain a high level of integration accuracy. Thus, very efficient NLP solvers are required when solving the finite problem.

We use the sequential direct method and the system of ODE is numerically integrated by the Matlab function `ode15s`. The default value for the scalar relative error tolerance is set to $1.0E-08$ and all the components of the vector of absolute error tolerance are set to $1.0E-08$. The sequentially discretized NLP problem is then solved by a stochastic metaheuristic. This is further elaborated in the next section.

3. Stochastic Metaheuristics

To compute global optimal solutions to NLP problems, stochastic or deterministic methods are available. Stochastic GO methods are able to provide a near-optimal solution in a short CPU time, although it may not be globally optimal.

Table 1: Nomenclature for the FA.

NP	Number of fireflies in the population
β	Attractiveness of a firefly
γ	Absorption coefficient/variation of attractiveness
α	Randomization parameter
p^1	The brightest firefly
p^{NP}	The less brighter firefly

In contrast, deterministic GO methods provide an interval within which the global optimal solution falls, although they require very large computational efforts [14]. Stochastic metaheuristics, such as FA, HS, DE, ABC, and DTS, include randomly generated numbers from probability distributions to define a set of solutions on the search region, as well as to move those solutions to hopefully better positions. They ensure convergence to a global solution in probability, while deterministic methods tend to guarantee asymptotic convergence. Global search techniques use exploration and exploitation search procedures that aim to diversify the search so that a global optimal solution is located and to intensify the search so that a good approximation is computed in a promising area of the search space.

Most stochastic metaheuristics are classified in terms of source of inspiration as nature-inspired algorithms. The well-known swarm-intelligence-based algorithms belong to a wider class called the bioinspired algorithms, and these are a subclass of nature-inspired algorithms [28]. The most popular are the algorithms based on swarm intelligence, with the FA and the ABC algorithm being two examples. On the other hand, the HS is not a bioinspired algorithm and has its inspiration from music. Although the DE algorithm is not based on any biological behavior, it can be classified as a bioinspired algorithm due to the keyword "evolution," but it is not swarm-intelligence-based [28]. Finally, the DTS algorithm is not a nature-inspired algorithm, but it has the particularity of using past information of the search to record it in memory. The majority of the metaheuristics define a set of approximate solutions at each iteration, called population, and generate new trial solutions that may be accepted as the solutions for the next iteration, if some improvement is detected relative to the current set of solutions.

We use the notation $p^i = (p_1^i, p_2^i, \ldots, p_n^i)^T$ to represent the ith point of a population of NP points, possible solutions of problem (1). The set $\Omega = \{p \in \mathbb{R}^n : p_j^{\text{L}} \leq p_j \leq p_j^{\text{U}},\ j = 1, \ldots, n\}$ defines the bound constraints of the problem.

3.1. The FA. The FA is a bioinspired metaheuristic algorithm that is capable of converging to a global solution of an optimization problem. It is inspired by the flashing behavior of fireflies at night [20, 29]. The main rules used to construct the FA are as follows: (i) all fireflies are unisex, meaning that any firefly can be attracted to any other brighter one; (ii) the brightness of a firefly is determined from the encoded objective function; (iii) attractiveness is directly proportional to brightness but decreases with distance.

Table 1 lists the most relevant nomenclature used in the basic FA. At the beginning, the positions of the fireflies in

Table 2: Nomenclature for the HS algorithm.

HM	Harmony memory
HMS	Size of the HM
p^{best}	The best solution in HM
p^{worst}	The worst solution in HM
HMCR	Harmony memory considering rate
PAR	The pitch adjusting rate
BW	Distance bandwidth

the search space are randomly generated. Then, they are evaluated by computing the corresponding objective function value and the points are ranked in ascending order; that is, p^1 is the brightest firefly, p^2 is the second brightest, and so on. In the classical FA, a firefly i ($i = 2, \ldots, \text{NP}$) is moved towards the brighter fireflies $j = 1, \ldots, i-1$ as follows:

$$p^i = p^i + \left((\beta_0 - \beta_{\min}) \exp\left(-\gamma \left\|p^i - p^j\right\|^2\right) + \beta_{\min}\right) \cdot \left(p^j - p^i\right) + \alpha\left(\lambda^i - 0.5\right) X, \tag{2}$$

where the second term on the right-hand side of (2) is due to the attraction while the third term is due to randomization with $\alpha \in (0, 1)$, β_0 is the attraction parameter when the distance is zero, λ^i is a number uniformly distributed in $[0, 1]$, and X is a vector that aims to scale the movement to the set Ω. The parameter γ is crucial to speed the convergence of the algorithm and it can take any value in the set $[0, \infty)$. When $\gamma \to \infty$, the attractiveness tends to the minimum constant value $\beta_{\min}$. To accelerate convergence and at the same time prevent premature convergence, the sequence of α values is made to slowly decrease along the iterative process. If a component j of the final position of a firefly falls outside the set $[p_j^{\text{L}}, p_j^{\text{U}}]$, it is shifted onto the boundaries.

3.2. The HS Algorithm. The HS algorithm was developed to solve GO problems in an analogy with the music improvisation process where music players improvise the pitches of their instruments to obtain better harmony [21, 30]. At each iteration k, the basic HS algorithm provides a set of solution vectors, available in the harmony memory (HM), from which the best and the worst solutions, in terms of objective function values, are identified. Table 2 lists the most relevant nomenclature used in the HS algorithm.

The HM contains HMS solution vectors that are maintained in memory throughout the iterative process. First, the solutions in the HM are randomly generated in the search space Ω, p^i, $i = 1, \ldots, \text{HMS}$. Then, they are evaluated and the best harmony, p^{best}, and the worst, p^{worst}, in terms of objective function value are identified. Hereafter, at each iteration, a new harmony is improvised; that is, a new vector v is generated, using three improvisation operators. When the *HM operator* is used, the component j of v is chosen from the HM with probability HMCR; otherwise, the *random selection operator* is used and the component is randomly generated in Ω:

$$v_j = \begin{cases} p_j^i, \ i \text{ random} \in \{1, \ldots, \text{HMS}\}, & \text{if } \lambda < \text{HMCR} \\ p_j^{\text{L}} + \lambda\left(p_j^{\text{U}} - p_j^{\text{L}}\right), & \text{otherwise.} \end{cases} \tag{3}$$

for $j = 1, \ldots, n$, where λ is a random number uniformly distributed in $[0, 1]$. Finally, the *pitch adjustment operator* is subsequently applied with a probability $0 < \text{PAR}(K) < 1$, which varies with the iteration counter K, to refine only the components j produced by the *HM operator*, as follows:

$$v_j = \begin{cases} v_j \pm \lambda \text{BW}(K), & \text{if } \lambda < \text{PAR}(K) \\ v_j, & \text{otherwise,} \end{cases} \tag{4}$$

where $0 < \text{BW}(K) \leq 1$ is the distance bandwidth that depends also on K [31]. Finally, the components of v are checked against the bounds and projected onto the boundaries if they fall outside. In the final stage, if $f(v) < f(p^{\text{worst}})$, the HM is updated since the new harmony is included in the HM, replacing the worst one.

Table 3: Nomenclature for the DE algorithm.

NP	Number of points in the population
F	Amplification parameter
$v \in \mathbb{R}^n$	Mutant point
$u \in \mathbb{R}^n$	Trial point
CR	Crossover parameter

3.3. The DE Algorithm. The DE is a bioinspired population-based algorithm that relies on three strategies—*mutation*, *crossover*, and *selection*—to define the NP solutions/points for the next iteration [22]. The most important nomenclature and parameters are shown in Table 3.

The initial population of points, $p^i \in \mathbb{R}^n$, $i = 1, \ldots, \text{NP}$, is randomly generated in the search space Ω. The most commonly used *mutation* strategy defines the mutant point i, v^i, as follows:

$$v^i = p^{r_1} + F\left(p^{r_2} - p^{r_3}\right) \tag{5}$$

with uniformly chosen random indices r_1, r_2, and r_3 from the set $\{1, 2, \ldots, \text{NP}\}$, mutually different, and F is a real parameter in $[0, 2]$ which controls the amplification of the differential variation, $p^{r_2} - p^{r_3}$. The indices r_1, r_2, and r_3 are also chosen to be different from the index i. p^{r_1} is called the base point.

The components of the mutant vector are then mixed with components of the p vector to generate the trial point, u^i. This strategy is referred to as *crossover* and can be described as follows:

$$u_j^i = \begin{cases} v_j^i, & \text{if } \lambda \leq \text{CR or } j = s_i \\ p_j^i, & \text{otherwise.} \end{cases} \tag{6}$$

Table 4: Nomenclature for the ABC algorithm.

NP	Number of bees in the colony
Nf	Number of food sources
$p \in \mathbb{R}^n$	Food source/solution
$v \in \mathbb{R}^n$	Mutant solution
"limit"	Limit for abandonment

Table 5: Nomenclature for the DTS algorithm.

TL	Tabu list
VRL	Visited region list
H	Ratio of accepting diversification point
$\text{Edg} = f_{\text{edg}} \min_j(p_j^{\text{U}} - p_j^{\text{L}})$	Edge length (with $0 < f_{\text{edg}} < 1$)
$\text{dVR} = r_{\text{dVR}}\text{Edg}$	Region radius in VRL ($r_{\text{dVR}} > 0$)

for $j = 1, \ldots, n$, where λ denotes a random number uniformly generated within the interval $[0, 1]$ and aims to perform the mixing of the component j of the points, $\text{CR} \in [0, 1]$, and s_i, an index randomly selected from $\{1, \ldots, n\}$, aims to ensure that u^i gets at least one component from v^i. Finally, the components of u are projected onto the boundaries of $[p^{\text{L}}, p^{\text{U}}]$ if they fall outside. Then, a *selection* strategy compares each trial point u^i with p^i in terms of objective function values and the best one is selected for the population of the next iteration. The variant described above is referred to as DE/rand/1/bin, where "rand" specifies the vector that is mutated, "1" defines the number of difference vectors, and "bin" means that crossover is based on independent binomial experiments [22]. There are other frequently used DE variants, for instance, the DE/best/1/bin which uses the best point of the population as the base point to define the mutant point and the DE/rand-to-best/1/bin with two differential variations:

$$v^i = p^i + F\left(p^{\text{best}} - p^i\right) + F\left(p^{r_1} - p^{r_2}\right), \tag{7}$$

where p^{best} is the best point in the current population [32].

3.4. The ABC Algorithm. The ABC algorithm is an optimization algorithm based on the intelligent behavior of honeybee swarms [23]. The colony of artificial bees of size NP includes the employed bees, the onlooker bees, and the scout bees. The first half of the colony consists of employed bees and the second half constitutes the onlookers and scouts. The position of a food source represents a possible solution of the optimization problem and the amount of nectar in the food source gives the quality of that solution. The number of food sources, Nf, is taken to be equal to the number of employed bees. The most important nomenclature and parameters in the ABC algorithm are shown in Table 4.

At the initial stage, a set of food source positions are randomly selected by the bees; that is, the positions are randomly generated in the search space Ω, and their nectar amounts are determined in terms of fitness values. During the employed bee phase, new candidate food positions, called mutant solutions, are produced as follows:

$$v_j^i = p_j^i + \lambda\left(p_j^i - p_j^k\right), \tag{8}$$

for $i = 1, \ldots, \text{Nf}$, where k (different from i) and j are randomly chosen indexes from the sets $\{1, \ldots, \text{Nf}\}$ and $\{1, \ldots, n\}$, respectively, and λ is a random number uniformly generated within the interval $[-1, 1]$. If the mutant component v_j^i falls outside $[p_j^{\text{L}}, p_j^{\text{U}}]$, it is shifted to the boundaries. Afterwards, the mutant solution v^i is compared with p^i and a greedy selection is applied to choose the one with better nectar, that is, with better fitness, as described in (9). If the current solution p^i is maintained (meaning that it has not been improved), its trial counter is increased.

On the other hand, during the onlooker bee phase, the food sources are randomly chosen according to probability values, P_i (associated with the food sources), that depend on their fitness F^i evaluated by

$$F^i = \begin{cases} \left(J\left(p^i\right) + 1\right)^{-1}, & \text{if } J\left(p^i\right) \geq 0 \\ 1 + \left|J\left(p^i\right)\right|, & \text{otherwise.} \end{cases} \tag{9}$$

The mutant solution is produced from the chosen old one as previously shown in (8) and the greedy selection is applied between the current p^i and its mutant. Finally, in the scout bee phase, if the trial counter of the solution that has not been improved mostly exceeds a predetermined value, called "limit," that solution is abandoned and replaced by a position randomly generated in the search space Ω [33].

3.5. The DTS Algorithm. The tabu search (TS) algorithm, introduced to continuous optimization in the paper [34], is capable of guiding the search out of local optima and exploring new regions for the global optimum. This is a point-by-point iterative procedure that maintains a list of the most recent movements, denoted by "tabu list" (TL), so that the movements that lead to solutions previously visited are avoided.

The DTS method developed in [24] is hybridization of the TS with a direct search method that aims to search the neighborhood of a local minimum. The method comprises three main search procedures: exploration, diversification, and intensification. The main loop consists of the exploration and the diversification search procedures. The exploration search aims to explore the search space Ω and uses direct search methods to be able to stabilize the search, in particular in the vicinity of a local minimum. Cycling is prevented by the TL and by another four TS memory elements: the multiranked tabu list, the tabu region, the semitabu region, and the visited region list (VRL). The VRL works as a diversification tool and is used to direct the search towards regions that have not been visited in the search space. The most important nomenclature and parameters of the DTS algorithm are listed in Table 5.

During the exploration procedure, DTS uses an adaptive pattern search strategy to generate an approximate descent direction (ADD) for the objective function J at the current approximation. First, based on an approximation $\overline{p}$, n pattern

directions parallel to the coordinate axes are constructed and n trial points x^i, $i = 1, \ldots, n$, are generated, along these directions with a given step length. Thus, the ADD v is obtained by

$$v = \sum_{i=1}^{n} w_i u_i \quad \text{where } w_i = \frac{J\left(x^i\right) - J\left(\overline{p}\right)}{\sum_{j=1}^{n} \left|J\left(x^j\right) - J\left(\overline{p}\right)\right|},\ u_i = -\frac{x^i - \overline{p}}{\left\|x^i - \overline{p}\right\|}. \tag{10}$$

Second, two local trial points x^{n+1} and x^{n+2} are then generated along v with two different step lengths aiming to explore the region along v [24]. The diversification procedure aims to randomly generate a new trial point $\widetilde{p} \in \Omega$ outside the previously visited regions. The VRL contains the centers of the visited regions and the frequency with which these regions are visited. If the shortest distance of $\widetilde{p}$ to a region's center, weighted by its frequency, exceeds the predefined region radius, given by H dVR, the point is accepted; otherwise, a new trial point is generated. When one of the best obtained trial solutions is sufficiently close to a global minimum, or its value has not been changed for a specified number of iterations, the DTS algorithm leaves the exploration and diversification search procedures and invokes an intensification procedure. Here, the HJ local search is used to compute a solution still closer to the global minimum (see the next section).

3.6. Hooke-and-Jeeves Local Search. A pattern search method directs the search towards a minimizer using a pattern of specific number points. At least $n+1$ points must be provided by the pattern, where n is the number of variables. Based on a current approximation, the HJ method uses a pattern search strategy to define a trial solution that gives an improvement in the objective function value, when compared with the current point [25]. Let the current approximation be denoted by $p^{(k_i)}$, where k_i represents the iteration counter in this iterative process. Then, a trial point, v^i, is generated along a search direction (starting from the current point) with a certain step size $\Delta^{(k_i)} > 0$ as follows:

$$v^i = p^{(k_i)} + \Delta^{(k_i)} d^i, \tag{11}$$

where d^i is the search direction chosen from a finite set $\mathcal{D}$ of positive spanning directions in $\mathbb{R}^n$. The most used set contains the $2n$ coordinate directions, defined as the positive and negative unit coordinate vectors $\mathcal{D} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$. The most important property of this set is that at least one of the coordinate directions is a descent direction for the objective J, when $p^{(k_i)}$ is not a stationary point of J. When the search fails to generate a trial point that is better than $p^{(k_i)}$, the iteration is called *unsuccessful*, the step size $\Delta^{(k_i)}$ is halved so that a refined search can be carried out, and $p^{(k_i+1)} \leftarrow p^{(k_i)}$. If $\Delta^{(k_i)}$ falls below a given stopping tolerance, the algorithm terminates and the current $p^{(k_i)}$ is considered an approximate minimizer of J. However, when at the end of each iteration the objective function value has reduced, then the iteration is called *successful*, $\Delta^{(k_i)}$ is not changed, and $p^{(k_i+1)} \leftarrow v^i$.

Table 6: Parameter values.

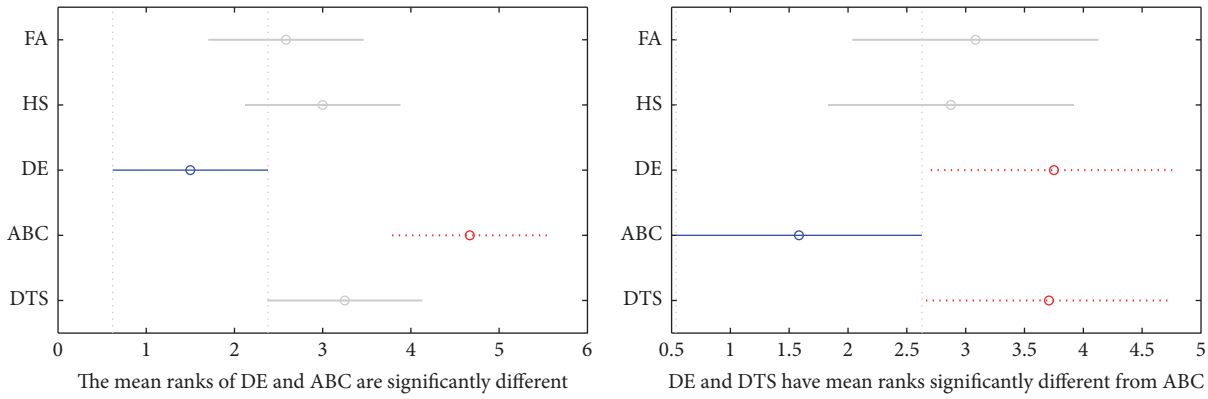
FA	
NP	$\min\{5n, 50\}$
α (initial)	0.25
γ (constant)	1
β_0	1
$\beta_{\min}$	0.2
HS	
HMS	$\min\{2n, 10\}$
HMCR	0.95
DE	
NP	$\min\{4n, 50\}$
F, CR	0.9
Mutation	DE/rand-to-best/1
ABC	
NP	$\min\{4n, 50\}$
"limit"	100
DTS	
H	1
f_{edge}	0.1
r_{dVR}	2

For a fair comparison, and to improve the quality of the produced solutions, we propose the implementation of the HJ intensification phase with all the above-mentioned metaheuristics. Based on the final solution provided by the metaheuristic, the HJ local search algorithm is invoked and allowed to run for $5n$ iterations. The number of function evaluations required by the HJ local search is added to the function evaluations of the metaheuristic, herein denoted by NFEval, to produce the total number of evaluations of the run.

4. Numerical Experiments

This section aims to present and analyze the statistical significance of the numerical results that were obtained when the metaheuristics are used to solve the DMbPE problems in the sequential direct method context. The dynamic system models were coded in the Matlab programming language and the computational application of the sequential direct method with the FA, HS, DE, ABC, and DTS codes was developed in Matlab programming environment. The computational tests were performed on a PC with a 2.2 GHz Core i7-2670QM and 8 GB of RAM. The parameter values for the tested algorithms are set as shown in Table 6. Other values have been tested, in particular for the parameter NP in the FA, DE, and ABC and the parameter HMS in HS, but those used seem adequate with reduced computational effort.

Nine case study problems have been selected to analyze the performance of the described stochastic algorithms. A comparison is made considering the quality of the solutions and the time spent to reach the solution after a threshold



(a) J_{avg} metric (b) T_{avg} metric

FIGURE 1: Estimates of the 95% confidence intervals for J_{avg} and T_{avg}, using $\text{NF}_{\max} = 50n$.

number of function evaluations. The problems and the experimental data can be found in the Appendix.

Table 7 shows J_{avg} (the average of the obtained final solutions), St.D. (the standard deviations of the solutions), and T_{avg} (the average time in seconds) after running each instance (inst.) 10 times with each metaheuristic (MH). For these experiments, each algorithm is terminated solely with the condition NFEval. $> \text{NF}_{\max}$, where we have tested three values of $\text{NF}_{\max}$: $50n$, $100n$, and $200n$. The goal here is to analyze the quality of the obtained solutions. Since the target objective value is 0, the lower the value of J_{avg} the better.

To analyze the statistical significance of the results, we use the Matlab function `friedman` that performs a nonparametric statistical test for multiple comparisons. The Friedman test aims to determine significant differences in the mean for one independent variable, for instance, J_{avg} or T_{avg}, with different levels that correspond to the five metaheuristics, and a dependent variable (corresponding to matched groups, herein taken as the 12 instances) [35]. The null hypothesis in this test is that the mean ranks assigned to the results of the metaheuristics under testing are not statistically different. Let k be the number of metaheuristics to be ranked and N be the number of groups. We note here that the distribution of the Friedman *statistic*, here denoted by Q_F, approaches the ordinary χ^2 distribution as N increases. The exact distribution of Q_F has been obtained for special cases of k and N (the reader is referred to [35] and the references cited in [36]). It has been concluded in [35] that "when the number of groups is moderately large (say greater than 5 for four or more ranks) the significance of the Friedman *statistic* can be tested by reference to the available χ^2 tables." However, in [36], it is argued that "the usual χ^2 approximation is grossly inaccurate for most commonly used combinations of k and N." Thus, hereafter, we use another *statistic* that has been recommended in [36, 37] and depends on Q_F, as well as on both k and N:

$$F_F = \frac{(N-1)\,Q_F}{N\,(k-1) - Q_F}, \tag{12}$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom.

When applied to J_{avg} for $\text{NF}_{\max} = 50n$, $Q_F = 25.27$ and the value of F_F is 12.229. With $k = 5$ and $N = 12$, F_F is distributed according to the F-distribution with 4 and 44 degrees of freedom. For a significance level of 5%, the critical value $F^{0.05}_{(4,44)}$ is 2.59 (computed by linear interpolation between $F^{0.05}_{(4,40)}$ and $F^{0.05}_{(4,50)}$). Since $12.229 > F^{0.05}_{(4,44)}$, we have enough evidence to reject the null hypothesis of "no significant differences on mean ranks"; that is, the observed differences between the five distributions of J_{avg} values are statistically significant. Pairwise comparisons may be carried out to determine which mean ranks are significantly different. The Matlab function `multcompare` is applied. The estimates of the 95% confidence intervals are shown in Figure 1(a) for each case under testing. The distributions of J_{avg} are significantly different if their intervals are disjoint and are not significantly different if their intervals overlap. Hence, we conclude that the mean rank produced by DE is significantly different from that of ABC. For the other pairs of comparisons, there are no significant differences on the mean ranks.

When the test is applied to T_{avg} for $\text{NF}_{\max} = 50n$, $Q_F = 14.97$ and $F_F = 4.985$ which exceeds the value of $F^{0.05}_{(4,44)}$, indicating that the null hypothesis is rejected at a significance level of 5%, and we conclude that the distributions of T_{avg} values have statistically significant differences. Figure 1(b) shows the estimates of the 95% confidence intervals. We conclude that the mean rank of ABC is significantly different from those of DE and DTS.

A similar analysis is made for J_{avg} and T_{avg}, when $\text{NF}_{\max} = 100n$. We obtain for the statistics $Q_F = 25.82$ with $F_F = 12.805$ and $Q_F = 7.47$ with $F_F = 2.027$, respectively. Hence, we may conclude that there is enough evidence to reject the null hypothesis of "no significant differences on mean ranks" of the distributions of J_{avg}, at a significance level of 5%. From Figure 2(a), we conclude that the mean ranks of (the distribution of J_{avg} values) ABC and DTS are significantly different from the mean ranks of DE. For the T_{avg} distributions, we conclude that there is not enough evidence

Table 7: Comparison of J_{avg}, St.D., and T_{avg}.

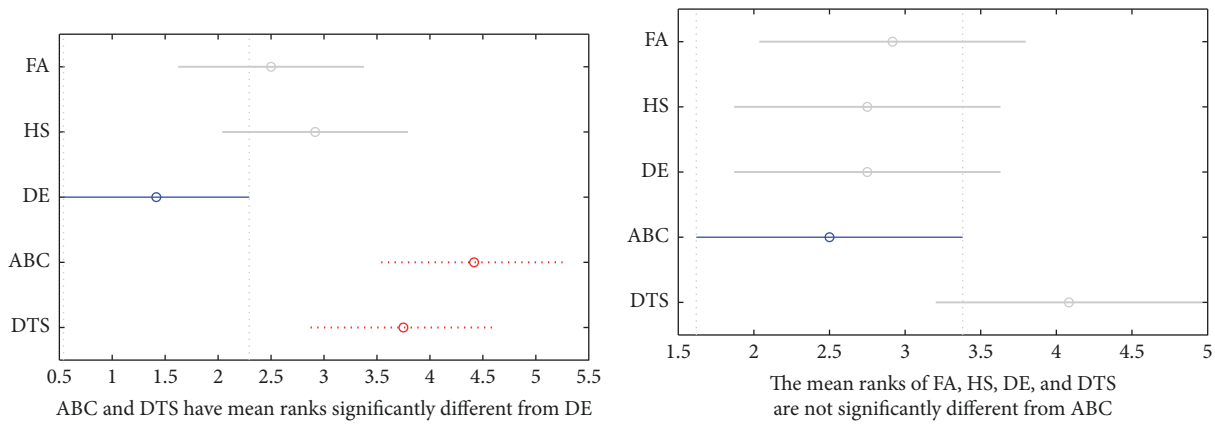
Inst.	n	MH	$NF_{max} = 50n$			$NF_{max} = 100n$			$NF_{max} = 200n$		
			J_{avg}	St.D.	T_{avg}	J_{avg}	St.D.	T_{avg}	J_{avg}	St.D.	T_{avg}
α-PIN	5	FA	$1.99583E+01$	$9.98E-02$	$2.09E+01$	$1.99145E+01$	$9.91E-02$	$3.27E+01$	$2.00343E+01$	$4.28E-01$	$5.45E+01$
		HS	$1.99172E+01$	$1.20E-01$	$2.31E+01$	$1.98832E+01$	$1.50E-02$	$3.16E+01$	$1.98974E+01$	$5.27E-02$	$6.01E+01$
		DE	$2.02428E+01$	$4.98E-01$	$2.51E+01$	$2.03504E+01$	$1.10E+00$	$3.20E+01$	$1.98827E+01$	$1.91E-02$	$5.30E+01$
		ABC	$4.40554E+02$	$6.38E+02$	$2.02E+01$	$8.97875E+01$	$7.64E+01$	$3.08E+01$	$2.30471E+01$	$3.15E+00$	$5.76E+01$
		DTS	$2.02791E+01$	$8.23E-01$	$2.56E+01$	$2.00559E+01$	$3.93E-01$	$3.94E+01$	$2.00243E+01$	$3.56E-01$	$1.03E+02$
BEL_A	2	FA	$2.21943E+01$	$2.02E-02$	$4.01E+00$	$2.21867E+01$	$6.26E-03$	$6.34E+00$	$2.21922E+01$	$2.13E-02$	$1.08E+01$
		HS	$2.22921E+01$	$2.73E-01$	$4.18E+00$	$2.22393E+01$	$1.68E-01$	$6.45E+00$	$2.21886E+01$	$9.73E-03$	$1.13E+01$
		DE	$2.22109E+01$	$3.72E-02$	$4.00E+00$	$2.21871E+01$	$6.95E-03$	$6.18E+00$	$2.21815E+01$	$1.56E-04$	$1.08E+01$
		ABC	$4.56940E+01$	$2.83E+01$	$4.06E+00$	$2.40930E+01$	$2.86E+00$	$6.50E+00$	$2.23062E+01$	$2.39E-01$	$1.12E+01$
		DTS	$2.22325E+01$	$7.77E-02$	$5.51E+00$	$2.23786E+01$	$4.51E-01$	$9.47E+00$	$2.22101E+01$	$2.84E-02$	$1.81E+01$
BEL_B	2	FA	$2.21928E+01$	$1.31E-02$	$3.67E+00$	$2.21867E+01$	$7.67E-03$	$6.43E+00$	$2.21821E+01$	$3.58E-04$	$1.09E+01$
		HS	$2.22161E+01$	$4.15E-02$	$3.67E+00$	$2.22001E+01$	$2.55E-02$	$6.02E+00$	$2.21870E+01$	$6.57E-03$	$1.12E+01$
		DE	$2.21918E+01$	$1.54E-02$	$3.80E+00$	$2.21824E+01$	$1.52E-03$	$6.12E+00$	$2.21814E+01$	$3.01E-07$	$1.09E+01$
		ABC	$3.13748E+01$	$1.32E+01$	$3.54E+00$	$2.49722E+01$	$6.36E+00$	$6.10E+00$	$2.25338E+01$	$1.11E+00$	$1.11E+01$
		DTS	$2.22010E+01$	$1.44E-02$	$4.16E+00$	$2.22058E+01$	$2.59E-02$	$7.01E+00$	$2.22025E+01$	$3.51E-02$	$1.23E+01$
CAT	3	FA	$2.6559E-03$	$3.59E-07$	$1.07E+01$	$2.6558E-03$	$1.52E-07$	$1.73E+01$	$2.6557E-03$	$1.25E-08$	$4.85E+01$
		HS	$2.6607E-03$	$1.50E-05$	$1.03E+01$	$2.6558E-03$	$1.31E-07$	$1.72E+01$	$2.6558E-03$	$2.17E-07$	$3.38E+01$
		DE	$2.6566E-03$	$1.98E-06$	$1.10E+01$	$2.6557E-03$	$1.46E-07$	$1.68E+01$	$2.6557E-03$	$6.03E-09$	$3.29E+01$
		ABC	$1.1355E-02$	$8.65E-03$	$9.73E+00$	$4.2229E-03$	$1.65E-03$	$1.65E+01$	$3.0625E-03$	$4.94E-04$	$3.46E+01$
		DTS	$2.6578E-03$	$4.72E-06$	$1.12E+01$	$2.6560E-03$	$5.19E-07$	$1.78E+01$	$2.6570E-03$	$2.55E-06$	$3.46E+01$

Table 7: Continued.

Inst.	n	MH	$\mathrm{NF}_{\max} = 50n$			$\mathrm{NF}_{\max} = 100n$			$\mathrm{NF}_{\max} = 200n$		
			J_{avg}	St.D.	T_{avg}	J_{avg}	St.D.	T_{avg}	J_{avg}	St.D.	T_{avg}
IRR1	2	FA	$2.9302E-06$	$1.83E-06$	$4.30E+00$	$2.1233E-06$	$1.38E-06$	$6.42E+00$	$1.3424E-06$	$3.49E-07$	$1.20E+01$
		HS	$3.9736E-06$	$3.22E-06$	$4.11E+00$	$2.2205E-06$	$1.21E-06$	$6.49E+00$	$1.4775E-06$	$3.14E-07$	$1.17E+01$
		DE	$4.3658E-06$	$6.02E-06$	$4.21E+00$	$1.5015E-06$	$5.28E-07$	$6.54E+00$	$1.1871E-06$	$2.86E-09$	$1.18E+01$
		ABC	$8.4434E-02$	$1.70E-01$	$3.81E+00$	$2.3661E-04$	$4.22E-04$	$6.76E+00$	$2.3818E-03$	$7.53E-03$	$1.21E+01$
		DTS	$7.2443E-06$	$8.14E-06$	$4.06E+00$	$2.5000E-06$	$1.24E-06$	$6.67E+00$	$5.4046E-06$	$5.09E-06$	$1.20E+01$
IRR2	2	FA	$6.9479E-05$	$7.39E-05$	$5.27E+00$	$2.5916E-05$	$3.78E-05$	$8.63E+00$	$6.5898E-06$	$2.81E-06$	$1.56E+01$
		HS	$2.6359E-04$	$5.05E-04$	$5.20E+00$	$3.9423E-05$	$5.12E-05$	$8.50E+00$	$2.6067E-05$	$2.47E-05$	$1.50E+01$
		DE	$2.0207E-05$	$4.06E-05$	$5.37E+00$	$6.6501E-06$	$2.66E-06$	$8.64E+00$	$4.4153E-06$	$7.19E-09$	$1.55E+01$
		ABC	$6.5399E-03$	$1.50E-02$	$4.94E+00$	$1.1674E-03$	$3.61E-03$	$8.59E+00$	$4.5931E-06$	$5.31E-07$	$1.55E+01$
		DTS	$4.5181E-05$	$5.14E-05$	$4.87E+00$	$4.5081E-05$	$3.55E-05$	$7.97E+00$	$5.6111E-05$	$8.44E-05$	$1.42E+01$
REV_A	4	FA	$8.9843E-06$	$2.48E-05$	$1.28E+01$	$1.2775E-06$	$1.44E-06$	$1.90E+01$	$2.1194E-06$	$3.47E-06$	$3.17E+01$
		HS	$3.6938E-05$	$9.88E-05$	$1.29E+01$	$4.3732E-05$	$1.08E-04$	$1.92E+01$	$1.4107E-06$	$1.41E-06$	$3.17E+01$
		DE	$5.5673E-07$	$3.03E-07$	$3.09E+01$	$1.5841E-06$	$2.92E-06$	$1.94E+01$	$6.1260E-07$	$5.65E-07$	$3.00E+01$
		ABC	$4.6219E-02$	$3.96E-02$	$9.18E+00$	$1.9570E-02$	$4.61E-02$	$1.68E+01$	$4.0868E-03$	$2.59E-03$	$3.01E+01$
		DTS	$5.8993E-04$	$1.10E-03$	$1.28E+01$	$7.9723E-05$	$7.71E-05$	$1.95E+01$	$3.5410E-05$	$4.42E-05$	$3.24E+01$
REV_B	4	FA	$1.6175E-03$	$8.04E-05$	$1.27E+01$	$1.5897E-03$	$5.59E-06$	$1.86E+01$	$1.5884E-03$	$1.39E-06$	$3.15E+01$
		HS	$1.6440E-03$	$1.07E-04$	$1.28E+01$	$1.5980E-03$	$1.01E-05$	$1.94E+01$	$1.5884E-03$	$8.81E-07$	$3.17E+01$
		DE	$1.5877E-03$	$3.90E-07$	$3.07E+01$	$1.5879E-03$	$7.44E-07$	$2.95E+01$	$1.5878E-03$	$5.17E-07$	$2.98E+01$
		ABC	$3.7122E-02$	$4.25E-02$	$9.11E+00$	$1.5726E-02$	$1.16E-02$	$1.67E+01$	$7.1333E-03$	$6.73E-03$	$3.06E+01$
		DTS	$1.8431E-03$	$3.23E-04$	$1.29E+01$	$1.6313E-03$	$6.82E-05$	$1.93E+01$	$1.7760E-03$	$2.33E-04$	$3.27E+01$

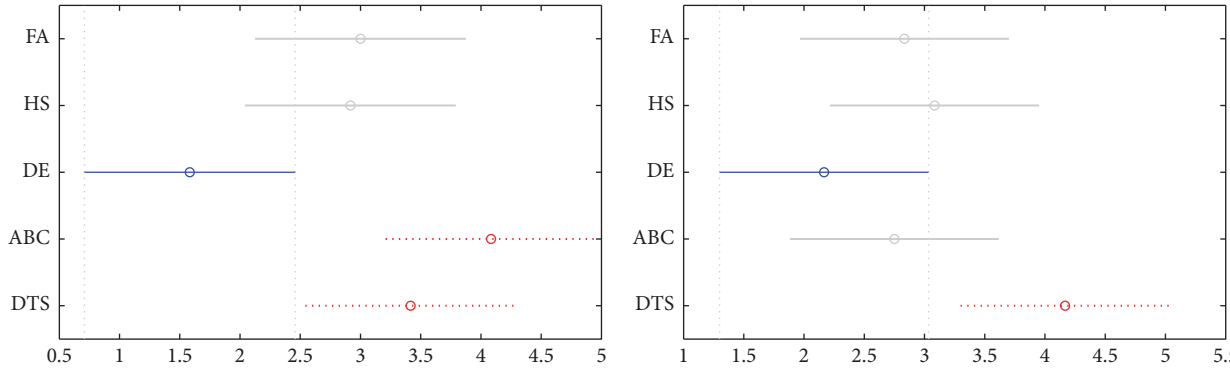
Table 7: Continued.

Inst.	n	MH	$NF_{max} = 50n$			$NF_{max} = 100n$			$NF_{max} = 200n$		
			J_{avg}	St.D.	T_{avg}	J_{avg}	St.D.	T_{avg}	J_{avg}	St.D.	T_{avg}
THEO1	3	FA	$1.0938E-03$	$8.53E-06$	$9.47E+00$	$1.0842E-03$	$8.85E-06$	$1.47E+01$	$1.0958E-03$	$9.80E-06$	$2.51E+01$
		HS	$1.0900E-03$	$1.61E-05$	$9.11E+00$	$1.0842E-03$	$1.61E-05$	$1.42E+01$	$1.0861E-03$	$1.55E-05$	$2.46E+01$
		DE	$1.0706E-03$	$1.42E-08$	$8.64E+00$	$1.0764E-03$	$9.19E-06$	$1.35E+01$	$1.0763E-03$	$9.19E-06$	$2.33E+01$
		ABC	$1.0934E-03$	$1.49E-05$	$8.07E+00$	$1.0809E-03$	$9.90E-06$	$1.38E+01$	$1.0766E-03$	$7.55E-06$	$2.41E+01$
		DTS	$1.0978E-03$	$1.05E-05$	$1.05E+01$	$1.0860E-03$	$8.10E-06$	$1.67E+01$	$1.0841E-03$	$1.31E-05$	$2.88E+01$
THEO2	2	FA	$3.5953E-02$	$1.52E-04$	$6.38E+00$	$3.5800E-02$	$1.38E-05$	$1.08E+01$	$3.5784E-02$	$4.51E-06$	$1.88E+01$
		HS	$3.5976E-02$	$1.92E-04$	$6.98E+00$	$3.5866E-02$	$9.75E-05$	$1.14E+01$	$3.5809E-02$	$4.50E-05$	$2.03E+01$
		DE	$3.5829E-02$	$6.46E-05$	$6.74E+00$	$3.5782E-02$	$8.44E-07$	$1.04E+01$	$3.5809E-02$	$4.50E-05$	$2.03E+01$
		ABC	$4.6244E-02$	$2.17E-02$	$8.47E+00$	$2.1159E-01$	$5.52E-01$	$1.30E+01$	$4.2209E-02$	$2.03E-02$	$2.12E+01$
		DTS	$3.5863E-02$	$1.24E-04$	$7.92E+00$	$3.5876E-02$	$1.15E-04$	$1.39E+01$	$3.5833E-02$	$5.42E-05$	$2.56E+01$
VOL_A	2	FA	$2.1700E-01$	$2.52E-01$	$3.13E+01$	$1.6923E-01$	$2.28E-01$	$4.50E+01$	$1.5620E-01$	$2.28E-01$	$8.31E+01$
		HS	$1.0265E-01$	$1.66E-01$	$2.90E+01$	$9.8232E-02$	$1.61E-01$	$5.51E+01$	$3.4138E-02$	$2.19E-02$	$8.89E+01$
		DE	$2.0002E-02$	$1.08E-02$	$3.31E+01$	$1.6915E-02$	$9.57E-03$	$4.76E+01$	$6.2283E-02$	$1.48E-01$	$9.26E+01$
		ABC	$1.7601E-01$	$2.94E-01$	$2.73E+01$	$1.0289E-01$	$1.56E-01$	$4.53E+01$	$3.8400E-02$	$3.45E-02$	$8.02E+01$
		DTS	$2.9206E-02$	$1.73E-02$	$2.96E+01$	$8.1213E-02$	$1.46E-01$	$4.70E+01$	$2.5780E-02$	$6.98E-03$	$8.79E+01$
VOL_B	2	FA	$1.0574E+00$	$1.77E-01$	$3.35E+01$	$1.0277E+00$	$1.66E-01$	$7.16E+01$	$9.7332E-01$	$1.61E-01$	$6.59E+01$
		HS	$9.5826E-01$	$1.03E-01$	$2.75E+01$	$1.0034E+00$	$1.58E-01$	$4.18E+01$	$9.7446E-01$	$1.33E-01$	$7.41E+01$
		DE	$9.3036E-01$	$3.68E-02$	$3.26E+01$	$9.0152E-01$	$2.33E-02$	$3.68E+01$	$9.3849E-01$	$1.11E-01$	$7.89E+01$
		ABC	$1.0558E+00$	$2.30E-01$	$2.54E+01$	$9.7174E-01$	$1.55E-01$	$4.56E+01$	$9.1724E-01$	$5.10E-02$	$6.42E+01$
		DTS	$9.7639E-01$	$1.14E-01$	$2.81E+01$	$9.7674E-01$	$1.04E-01$	$4.54E+01$	$8.9907E-01$	$2.99E-02$	$7.78E+01$



(a) J_{avg} metric (b) T_{avg} metric

FIGURE 2: Estimates of the 95% confidence intervals for J_{avg} and T_{avg}, using $NF_{max} = 100n$.



(a) J_{avg} metric (b) T_{avg} metric

FIGURE 3: Estimates of the 95% confidence intervals for J_{avg} and T_{avg}, using $NF_{max} = 200n$.

to reject the hypothesis of "no significant differences on mean ranks," at a significance level of 5%. Figure 2(b) shows the corresponding estimates of the 95% confidence intervals and it is noticed that all pairs have their intervals overlapping.

When the statistical analysis is extended to the distributions of J_{avg} and T_{avg} values, for $NF_{max} = 200n$, the statistic Q_F gives 16.34 (with $F_F = 5.677$) and 10.64 (with $F_F = 3.133$), respectively. Hence, we have evidence to reject the null hypothesis relative to the J_{avg} values, at a significance level of 5%. Figure 3(a) shows the corresponding estimates of the 95% confidence intervals (ABC and DTS have mean ranks significantly different from DE). As far as the distributions of T_{avg} values are concerned, we conclude that there is enough evidence to reject the hypothesis of "no significant differences on mean ranks" at a significance level of 5%. From Figure 3(b), we conclude that the differences are on the mean ranks of DE and DTS. We note that, at a significance level of 1%, a smaller probability of making a wrong decision, where the critical value $F^{0.01}_{(4,44)}$ is 3.79 (computed by linear interpolation between $F^{0.01}_{(4,40)}$ and $F^{0.01}_{(4,50)}$), there is no evidence to reject the null hypothesis.

We now show in Table 8 the best solution obtained by each metaheuristic, J_{best} (after the 10 runs), and the time required to reach that value, T_{best}, for the three values of NF_{max}. Based on the metrics J_{best} and T_{best}, we use a graphical procedure to visualize the performance differences among the results produced by the five metaheuristics, in relative terms on the 12 instances, known as performance profiles [38]. Each plot reports (on the vertical axis) the percentage of problems solved with each metaheuristic that is within a certain threshold, τ (on the horizontal axis), of the best result. The performance profiles of the five distributions of J_{best} values for $NF_{max} = 50n$ are shown in Figure 4(a). Figure 4(b) corresponds to the metric T_{best}.

The higher the percentage, the better. A higher value for $\tau = 1$ means that the metaheuristic achieves the lowest J_{best} value (or the smallest T_{best}) mostly.

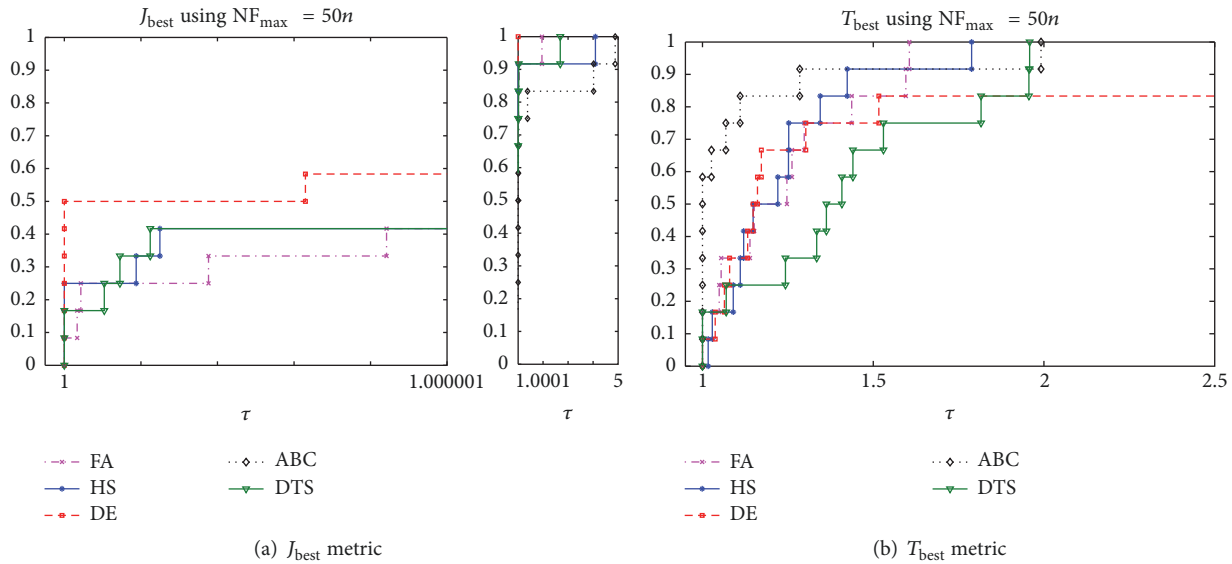
Therefore, the metaheuristic DE is the most successful since it has the highest probability of achieving the lowest J value, when $NF_{max} = 50n$. From the plot, the probability that DE wins on a given instance is 0.5 since it produces the lowest J_{best} in 6 of the 12 instances. It is followed by HS that

Table 8: Best results (T_{best} and J_{best}).

Inst.	MH	$NF_{max} = 50n$		$NF_{max} = 100n$		$NF_{max} = 200n$	
		T_{best}	J_{best}	T_{best}	J_{best}	T_{best}	J_{best}
α-PIN	FA	$2.17E+01$	$1.987279E+01$	$3.08E+01$	$1.987226E+01$	$4.80E+01$	$1.987264E+01$
	HS	$2.15E+01$	$1.987260E+01$	$3.04E+01$	$1.987228E+01$	$5.18E+01$	$1.987230E+01$
	DE	$2.29E+01$	$1.987271E+01$	$2.92E+01$	$1.987296E+01$	$5.18E+01$	$1.987263E+01$
	ABC	$1.51E+01$	$7.990355E+01$	$2.54E+01$	$2.567832E+01$	$4.82E+01$	$2.036694E+01$
	DTS	$2.31E+01$	$1.987234E+01$	$3.83E+01$	$1.987266E+01$	$6.91E+01$	$1.987274E+01$
BEL_A	FA	$4.29E+00$	$2.218250E+01$	$5.97E+00$	$2.218155E+01$	$1.10E+01$	$2.218149E+01$
	HS	$3.95E+00$	$2.218227E+01$	$6.15E+00$	$2.218149E+01$	$1.14E+01$	$2.218311E+01$
	DE	$4.48E+00$	$2.218143E+01$	$6.05E+00$	$2.218147E+01$	$1.06E+01$	$2.218142E+01$
	ABC	$3.44E+00$	$2.227154E+01$	$5.78E+00$	$2.218221E+01$	$1.07E+01$	$2.218150E+01$
	DTS	$4.59E+00$	$2.218762E+01$	$9.52E+00$	$2.218738E+01$	$1.76E+01$	$2.218458E+01$
BEL_B	FA	$5.09E+00$	$2.218269E+01$	$6.07E+00$	$2.218152E+01$	$1.09E+01$	$2.218170E+01$
	HS	$3.52E+00$	$2.218164E+01$	$6.04E+00$	$2.218142E+01$	$1.12E+01$	$2.218159E+01$
	DE	$3.68E+00$	$2.218181E+01$	$6.26E+00$	$2.218142E+01$	$1.11E+01$	$2.218142E+01$
	ABC	$3.17E+00$	$2.218809E+01$	$5.89E+00$	$2.218259E+01$	$1.15E+01$	$2.218151E+01$
	DTS	$4.32E+00$	$2.218208E+01$	$7.33E+00$	$2.218152E+01$	$1.23E+01$	$2.218187E+01$
CAT	FA	$1.03E+01$	$2.655668E-03$	$1.76E+01$	$2.655668E-03$	$4.78E+01$	$2.655668E-03$
	HS	$1.01E+01$	$2.655686E-03$	$1.61E+01$	$2.655669E-03$	$3.23E+01$	$2.655669E-03$
	DE	$1.06E+01$	$2.655683E-03$	$1.62E+01$	$2.655667E-03$	$3.11E+01$	$2.655667E-03$
	ABC	$9.82E+00$	$2.666711E-03$	$1.65E+01$	$2.946387E-03$	$3.45E+01$	$2.662231E-03$
	DTS	$1.05E+01$	$2.655667E-03$	$1.73E+01$	$2.655680E-03$	$3.40E+01$	$2.655672E-03$
IRR1	FA	$4.45E+00$	$1.221743E-06$	$6.51E+00$	$1.228962E-06$	$1.18E+01$	$1.187445E-06$
	HS	$4.29E+00$	$1.375865E-06$	$6.53E+00$	$1.190046E-06$	$1.15E+01$	$1.193409E-06$
	DE	$4.49E+00$	$1.188136E-06$	$6.41E+00$	$1.194300E-06$	$1.18E+01$	$1.185902E-06$
	ABC	$4.33E+00$	$4.204506E-05$	$6.90E+00$	$1.271379E-06$	$1.21E+01$	$1.185851E-06$
	DTS	$4.22E+00$	$1.333455E-06$	$6.77E+00$	$1.228666E-06$	$1.18E+01$	$1.228072E-06$
IRR2	FA	$5.49E+00$	$5.793696E-06$	$8.67E+00$	$5.290805E-06$	$1.55E+01$	$4.532302E-06$
	HS	$5.40E+00$	$7.799776E-06$	$8.10E+00$	$7.581183E-06$	$1.54E+01$	$4.946930E-06$
	DE	$5.65E+00$	$4.951955E-06$	$8.65E+00$	$4.512205E-06$	$1.53E+01$	$4.411032E-06$
	ABC	$5.15E+00$	$1.539352E-05$	$8.64E+00$	$4.473528E-06$	$1.52E+01$	$4.411030E-06$
	DTS	$4.82E+00$	$5.176208E-06$	$8.00E+00$	$8.305471E-06$	$1.41E+01$	$5.550882E-06$
REV_A	FA	$1.30E+01$	$2.390974E-07$	$1.95E+01$	$2.301914E-07$	$3.13E+01$	$1.933663E-07$
	HS	$1.29E+01$	$4.456872E-07$	$1.95E+01$	$2.772230E-07$	$3.09E+01$	$2.864920E-07$
	DE	$3.13E+01$	$1.960909E-07$	$1.78E+01$	$2.060756E-07$	$3.03E+01$	$2.034098E-07$
	ABC	$1.03E+01$	$1.184856E-02$	$1.64E+01$	$3.235180E-04$	$3.05E+01$	$4.414293E-04$
	DTS	$1.28E+01$	$3.003054E-07$	$1.90E+01$	$2.925082E-07$	$3.22E+01$	$4.346120E-07$
REV_B	FA	$1.16E+01$	$1.587466E-03$	$1.70E+01$	$1.587489E-03$	$3.22E+01$	$1.587486E-03$
	HS	$1.30E+01$	$1.587876E-03$	$1.86E+01$	$1.587498E-03$	$3.08E+01$	$1.587476E-03$
	DE	$3.07E+01$	$1.587467E-03$	$2.88E+01$	$1.587501E-03$	$2.89E+01$	$1.587516E-03$
	ABC	$7.27E+00$	$7.753336E-03$	$1.70E+01$	$2.810094E-03$	$3.03E+01$	$1.816843E-03$
	DTS	$1.32E+01$	$1.587997E-03$	$1.90E+01$	$1.587864E-03$	$3.36E+01$	$1.587618E-03$
THEO1	FA	$9.28E+00$	$1.089660E-03$	$1.35E+01$	$1.070636E-03$	$2.58E+01$	$1.089664E-03$
	HS	$8.95E+00$	$1.070635E-03$	$1.39E+01$	$1.070637E-03$	$2.33E+01$	$1.070635E-03$
	DE	$8.21E+00$	$1.070638E-03$	$1.31E+01$	$1.070635E-03$	$2.25E+01$	$1.070635E-03$
	ABC	$7.15E+00$	$1.071921E-03$	$1.23E+01$	$1.071302E-03$	$2.30E+01$	$1.070637E-03$
	DTS	$1.03E+01$	$1.089659E-03$	$1.61E+01$	$1.070635E-03$	$2.78E+01$	$1.070638E-03$
THEO2	FA	$6.22E+00$	$3.580257E-02$	$1.10E+01$	$3.578306E-02$	$1.89E+01$	$3.578128E-02$
	HS	$6.78E+00$	$3.579577E-02$	$1.13E+01$	$3.578748E-02$	$2.03E+01$	$3.578180E-02$
	DE	$6.45E+00$	$3.578201E-02$	$1.02E+01$	$3.578109E-02$	$2.03E+01$	$3.578180E-02$
	ABC	$7.99E+00$	$3.608823E-02$	$1.24E+01$	$3.580068E-02$	$2.09E+01$	$3.578108E-02$
	DTS	$8.76E+00$	$3.578341E-02$	$1.42E+01$	$3.578354E-02$	$2.54E+01$	$3.578140E-02$

Table 8: Continued.

Inst.	MH	$NF_{max} = 50n$		$NF_{max} = 100n$		$NF_{max} = 200n$	
		T_{best}	J_{best}	T_{best}	J_{best}	T_{best}	J_{best}
VOL_A	FA	$1.37E+01$	$7.404584E-03$	$2.56E+01$	$9.624342E-03$	$3.33E+01$	$3.602099E-03$
	HS	$1.60E+01$	$1.550778E-02$	$3.04E+01$	$2.731927E-02$	$3.82E+01$	$1.388033E-02$
	DE	$1.19E+01$	$3.787360E-03$	$1.81E+01$	$3.574318E-03$	$4.06E+01$	$3.574082E-03$
	ABC	$2.37E+01$	$5.221512E-03$	$2.73E+01$	$4.371459E-03$	$4.51E+01$	$4.326559E-03$
	DTS	$2.33E+01$	$1.018590E-02$	$4.11E+01$	$5.239623E-03$	$8.39E+01$	$1.922888E-02$
VOL_B	FA	$1.36E+01$	$9.124563E-01$	$3.72E+01$	$8.667088E-01$	$4.07E+01$	$8.684604E-01$
	HS	$1.66E+01$	$8.700917E-01$	$2.48E+01$	$8.810708E-01$	$3.90E+01$	$8.697327E-01$
	DE	$1.54E+01$	$8.709003E-01$	$1.89E+01$	$8.703102E-01$	$4.67E+01$	$8.667010E-01$
	ABC	$1.51E+01$	$8.955672E-01$	$2.02E+01$	$8.671336E-01$	$6.42E+01$	$8.700530E-01$
	DTS	$2.66E+01$	$8.919989E-01$	$3.87E+01$	$9.019120E-01$	$7.12E+01$	$8.679623E-01$



(a) J_{best} metric (b) T_{best} metric

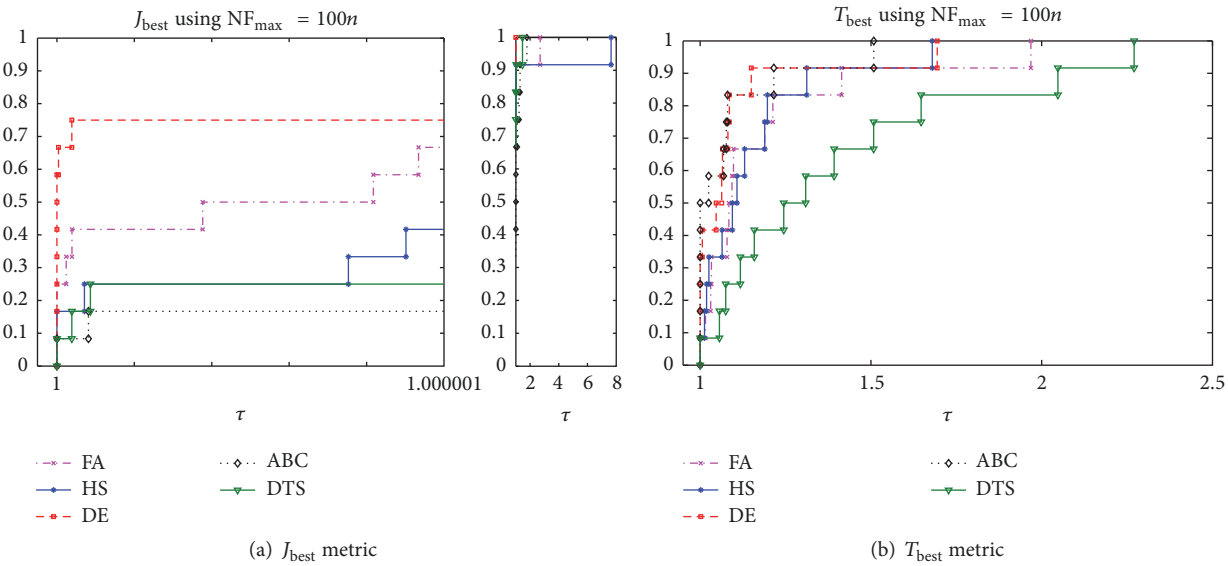
Figure 4: Performance profiles for the five metaheuristics, using $NF_{max} = 50n$.

has a probability of winning (prob_w) of 0.25, by DTS with prob_w = 0.17, and by FA with prob_w = 0.08. We note that the differences on the produced J_{best} are so small that the plots of the cumulative distribution are packed around $\tau = 1$. From Figure 4(b), it is possible to conclude that the fastest metaheuristic is ABC since the probability of achieving the smallest value of T (time in seconds) is 0.58.

Similarly, Figures 5(a) and 5(b) contain the performance profiles of the five distributions of J_{best} and T_{best}, respectively, when $NF_{max} = 100n$. From the figure on the left, we may conclude that DE is the most successful since it has prob_w = 0.58, followed by FA with prob_w = 0.25 (achieving the best results in 3 out of 12 instances), by HS with prob_w = 0.17, and by ABC and DTS with prob_w = 0.08. As far as T_{best} is concerned, it is possible to conclude that the metaheuristic with the highest probability of requiring the lowest time is ABC followed by DE.

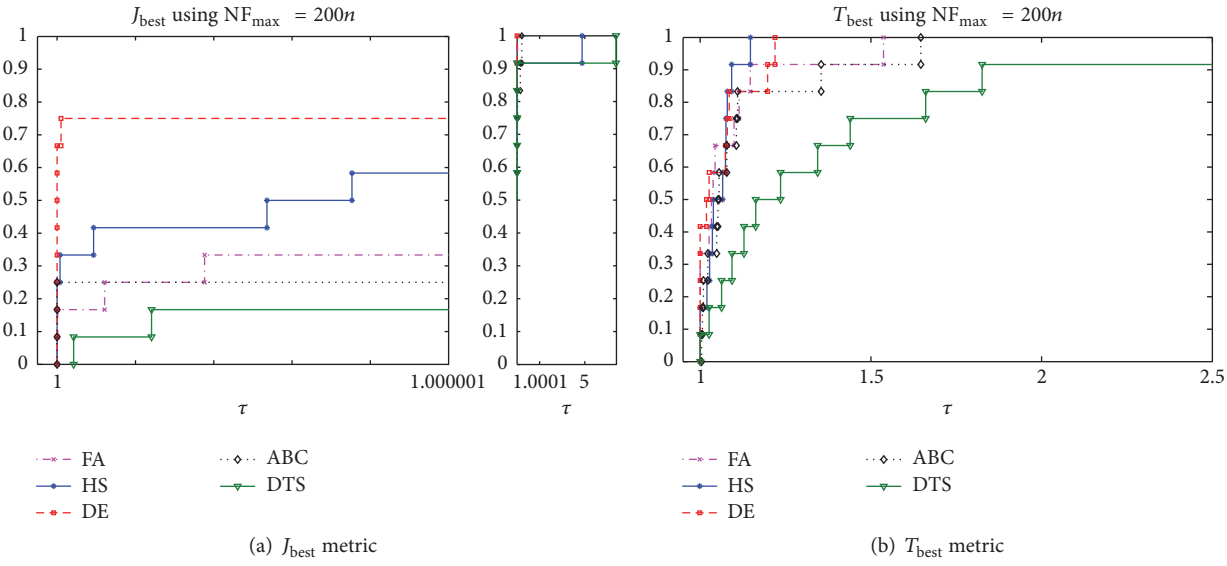
Finally, Figures 6(a) and 6(b) display the performance profiles of the metrics J_{best} and T_{best}, respectively, when $NF_{max} = 200n$. From these comparisons, we may conclude that DE is the most successful in reaching the lowest J_{best}, with prob_w = 0.5, followed by HS and ABC with prob_w = 0.25 and finally the metaheuristics FA and DTS with prob_w = 0.08. When the profiles of T_{best} are analyzed, it is possible to conclude that the metaheuristic DE is in general faster than the others but for some problems is outstripped by FA and HS.

Tables 9–19 contain comparative results relative to 11 instances. (The results obtained for instance VOL_A are not used in the comparison since the experimental data have been herein generated for this study.) Tables 9–19 show the best and the average J values (computed from the 10 runs), as well as the average CPU time and function evaluations, T_{avg} and NFE_{avg}, respectively. Our results always appear along the first row of each table and are taken from the metaheuristic that produced the lowest J_{best} with the least function evaluations. This is identified in the table caption. (We note that the implemented metaheuristics have been enhanced with the HJ local



(a) J_{best} metric (b) T_{best} metric

Figure 5: Performance profiles for the five metaheuristics, using $NF_{max} = 100n$.



(a) J_{best} metric (b) T_{best} metric

Figure 6: Performance profiles for the five metaheuristics, using $NF_{max} = 200n$.

search, and the reported NFE_{avg} considers the sum of NF_{max} with the function evaluations required by the HJ search.) In the tables, "–" means that the information is not available in the cited paper and "n.a." means "not applicable." The comparisons use the results reported in the papers [2, 4, 10, 12–14, 16–19, 26]. Results in [16] are based on a trigonometric version of the DE, and in [17], a modified version of the DE is implemented. The authors in [12] use a modified version of the metaheuristic scatter search (SSm) with a local search as an improvement method. The results shown in the table use the function `fmincon` from Matlab. SSm uses a population of 100 points and a penalty technique with a weight of 1000 to penalize infeasible solutions. In [19], the sequential approach is implemented with an interval-based method for the NLP optimization. The therein obtained solution is an ϵ-global solution. Global solutions are produced by the method of orthogonal collocation on finite elements presented in [26]. This strategy is implemented within Matlab code Dynopt after 100 runs of a multistart approach. (We note that a direct comparison of the CPU time cannot be done since the computational platforms are different.)

Overall, after all the numerical comparisons, it is possible to conclude that the selected metaheuristics when enhanced with the local intensification phase are effective in achieving good quality solutions with a reduced computational effort. Furthermore, we have also shown that the sequential direct method combined with the selected metaheuristics competes

Table 9: Comparative results for α−PIN (FA/100n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$1.987226E+01$	$1.991449E+01$	$3.27E+01$	753
[2][♭]	$1.987E+01$	–	$1.22E+02$	(9518 iter.)
[12][†]	$1.9872E+01$	$2.4747E+01$	$0.41E+02$	1163
[13][§]	$1.98772E+01$	$2.56777E+01$	–	9148
[14][‡]	$1.987E+01$	n.a.	$8.92E+03$	(2 iter.)

[♭]PC: Pentium 4, 1.80 GHz. [†]PC: Pentium 4, 3.06 GHz. [§]PC with a 2.7 GHz Core i7-4600U and 8 GB of memory. [‡]PC: AMD Athlon II, 2.99 GHz.

Table 10: Comparative results for BEL_A (loose bounds on parameters) (DE/200n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$2.218142E+01$	$2.218148E+01$	$1.08E+01$	434
[10][†]	$2.25684E+01$	n.a.	$9.71E+03$	(10000 iter.)

[†]Result with relative convergence tolerance of $7.5E-01$. Computer HP J2240.

Table 11: Comparative results for BEL_B (tight bounds on parameters) (HS/100n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$2.218142E+01$	$2.220014E+01$	$6.02E+00$	245
[10][†]	$2.25684E+01$	n.a.	$7.84E+03$	(9028 iter.)

[†]Result with relative convergence tolerance of $1.0E-03$. Computer HP J2240.

Table 12: Comparative results for CAT (DTS/50n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$2.655667E-03$	$2.657778E-03$	$1.12E+01$	256
[4]	$2.65567E-03$	n.a.	–	–
[10][♭]	$2.6384E-03$	n.a.	$1.58E+02$	(26 iter.)
[16][§]	$2.655666E-03$	$2.655666E-03$	$1.81E+00$	–
[17]	$2.6557E-03$	–	$0.82E+00$	–
[18][‡]	$2.65567E-03$	n.a.	$2.66E+04$	(67 iter.)
[19][†]	$2.6557E-03$	n.a.	$1.11E+01$	(182 iter.)

[♭]Result with relative convergence tolerance of $1.0E-02$. Computer HP J2240. [§]PC: Pentium IV, 2.4 GHz, 256 MB RAM. [‡]Ultra SPARC-II CPU (2 × 360 MHz), 512 MB RAM. Optimality margin $\epsilon = 1E-02$. [†]Relative convergence tolerance of $1.0E-03$. PC: Intel Pentium 4 with 3.2 GHz.

Table 13: Comparative results for IRR1 (ABC/200n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$1.185851E-06$	$2.381824E-03$	$1.21E+01$	449
[4]	$1.18584E-06$	n.a.	–	–
[10][♭]	$1.1858E-06$	n.a.	$1.08E+01$	(38 iter.)
[16][§]	$1.185845E-06$	$1.185845E-06$	$0.46E+00$	–
[17]	$1.1858E-06$	–	$0.33E+00$	–
[18][‡]	$1.18562E-06$	n.a.	$7.67E+02$	(37 iter.)
[19][†]	$1.1858E-06$	n.a.	$0.23E-01$	(4 iter.)

[♭]Result with relative convergence tolerance of $1.0E-02$. Computer HP J2240. [§]PC: Pentium IV, 2.4 GHz, 256 MB RAM. [‡]Ultra SPARC-II CPU (2 × 360 MHz), 512 MB RAM. Optimality margin $\epsilon = 1E-02$. [†]Relative convergence tolerance of $1.0E-03$. PC: Intel Pentium 4 with 3.2 GHz.

very favorably with exact methods and other metaheuristics available in the literature.

Table 14: Comparative results for IRR2 (ABC/200n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$4.411030E-06$	$4.593098E-06$	$1.55E+01$	447
[26][†]	$1.7604E-04$	$1.7604174E-04$	–	–

[†]Best solution obtained after 100 multistarts running Dynopt.

5. Conclusions

In this paper, we have analyzed the performance of five well-known metaheuristics when solving parameter estimation problems in dynamic system models. The sequential numerical direct method is applied to the DMbPE problem and the resulting optimization is performed directly making use of a numerical integration formula for solving the system of ODE. Using nine DMbPE problems and different experimental data, error-free and random error-added data, a total of 12 instances have been used in the comparative experiments. The solutions produced by the metaheuristics have been analyzed in terms of quality, by stopping the algorithms after a specified number of function evaluations, and compared using statistical hypotheses testing. The statistical tests show that the average obtained solutions and the average CPU time are considered to be mostly significantly different, at a significance level of 5%. The best solutions produced by the metaheuristics are analyzed by means of the performance profiles. After the graphical comparisons for different numbers of allowed function evaluations, we are able to conclude that the DE is the metaheuristic that has the highest probability of giving the lowest value of the objective function J mostly. It is followed by HS. The FA, ABC, and DTS metaheuristics are, in this sequence, the least effective. Thus,

Table 15: Comparative results for REV_A (FA/200n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$1.933663E-07$	$2.119351E-06$	$3.17E+01$	953
[10]†	$3.367E-07$	n.a.	$2.73E+02$	(56 iter.)

†Result with absolute convergence tolerance of $1.0E-08$. Computer HP J2240.

Table 16: Comparative results for REV_B (FA/50n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$1.587466E-03$	$1.617494E-03$	$1.27E+01$	374
[10]†	$1.586E-03$	n.a.	$5.68E+02$	(349 iter.)
[19]‡	$1.5875E-03$	n.a.	$2.62E+03$	(40552 iter.)

†Result with relative convergence tolerance of $1.0E-02$. Computer HP J2240. ‡Relative convergence tolerance of $1.0E-03$; PC/Intel Pentium 4 with 3.2 GHz.

Table 17: Comparative results for THEO1 (HS/50n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$1.070635E-03$	$1.090058E-03$	$9.11E+00$	255
[26]†	$4.01E-03$	$4.0303E-03$	–	–

†Best local solution after 100 multistarts running Dynopt.

Table 18: Comparative results for THEO2 (ABC/200n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$3.578108E-02$	$4.220937E-02$	$2.12E+01$	448
[26]†	$2.595E-02$	$2.595E-02$	–	–

†Solution obtained after 100 multistarts running Dynopt.

Table 19: Comparative results for VOL_B (DE/200n).

Source	J_{best}	J_{avg}	T_{avg}	NFE_{avg}
	$8.667010E-01$	$9.384865E-01$	$7.89E+01$	440
[10]†	$1.3194E-03$	n.a.	$1.00E+05$	(1000 iter.)
[19]‡	$1.2492E-03$	n.a.	$4.30E+01$	(536 iter.)

†Computer HP J2240. ‡Relative convergence tolerance of $1.0E-03$; PC/Intel Pentium 4 with 3.2 GHz.

our recommendation for a reader that favors good quality solutions falls in the DE and HS metaheuristics. On the other hand, ABC and DE are in general the metaheuristics that require the least computational effort, meaning that, for a limited computational budget, our recommendation is to choose one of them. Finally, from the comparisons with other results in the literature, involving both sequential and simultaneous direct methods and stochastic as well as deterministic global optimization methods, we state that the sequential direct method combined with one of the above recommended metaheuristics is a good alternative to solve DMbPE problems. Good quality solutions with a reduced computational effort are thus provided.

Appendix

DMbPE Problems

α-Pinene First-Order Kinetics (α−PIN). A system of five ODE with five parameters is given in [13]

$$\begin{aligned} y_1' &= -(p_1 + p_2)\, y_1, \\ y_2' &= p_1 y_1, \\ y_3' &= p_2 y_1 - (p_3 + p_4)\, y_3 + p_5 y_5, \\ y_4' &= p_3 y_3, \\ y_5' &= p_4 y_3 - p_5 y_5, \end{aligned} \tag{A.1}$$

for $t \in [0, 36420]$, with initial conditions $y_1(0) = 100$, $y_2(0) = 0$, $y_3(0) = 0$, $y_4(0) = 0$, and $y_5(0) = 0$. The lower and upper bounds on the parameters are $[p_i^L, p_i^U] = [0, 5.0E-05]$, $i = 1, \ldots, 5$. Experimental data are available in Table 20.

Bellman's Problem (BEL). This ODE has one dependent variable and two parameters (see Section 7.4 in [10]). The herein tested version uses the exponential transformation:

$$y' = e^{-p_1}(126.2 - y)(91.9 - y)^2 - e^{-p_2} y^2, \tag{A.2}$$

for $t \in [0, 39.0]$, with initial condition $y(0) = 0$. The following loose lower and upper bounds on the parameters $[p_i^L, p_i^U] = [5, 15]$, $i = 1, 2$, can be considered defining the instance BEL_A. A second experiment was carried out with this problem using the following tight lower and upper bounds $[p_1^L, p_1^U] = [10, 14]$ and $[p_2^L, p_2^U] = [6, 10]$, giving the instance BEL_B. Experimental data have been generated using $t = [1, 2, 3, 4, 5, 6, 7, 9, 11, 14, 19, 24, 29, 39]$ and $p_1 = 3$ and $p_2 = 1$ with a small amount of random error added [10] (see Table 21).

Catalytic Cracking of Gas Oil (CAT). This ODE has two dependent variables and three parameters (see Section 6.3 in [19] and Section 7.3 in [10]):

$$\begin{aligned} y_1' &= -(p_1 + p_3)\, y_1^2, \\ y_2' &= p_1 y_1^2 - p_2 y_2, \end{aligned} \tag{A.3}$$

for $t \in [0, 0.95]$, with initial conditions $y_1(0) = 1$ and $y_2(0) = 0$. Lower and upper bounds on the parameters are $[p_i^L, p_i^U] = [0, 20]$, $i = 1, 2, 3$, and the experimental data in Table 22 have been generated using the following twenty time instants: 0.025, 0.05, 0.075, 0.1, 0.125, 0.150, 0.175, 0.200, 0.225, 0.250, 0.300, 0.350, 0.400, 0.450, 0.500, 0.550, 0.650, 0.750,

Table 20: Experimental values for $y_1, \ldots, y_5$ for eight time instants.

	1230.0	3060.0	4920.0	7800.0	10680.0	15030.0	22620.0	36420.0
y_1^{obs}	88.35	76.4	65.1	50.4	37.5	25.9	14.0	4.5
y_2^{obs}	7.3	15.6	23.1	32.9	42.7	49.1	57.4	63.1
y_3^{obs}	2.3	4.5	5.3	6.0	6.0	5.9	5.1	3.8
y_4^{obs}	0.4	0.7	1.1	1.5	1.9	2.2	2.6	2.9
y_5^{obs}	1.75	2.8	5.8	9.3	12.0	17.0	21.0	25.7

Table 21: Error-added generated experimental values for 14 time instants, available in [10].

	1	2	3	4	5	6	7
y^{obs}	1.4	6.3	10.4	14.2	17.6	21.4	23.0
	9	11	14	19	24	29	39
y^{obs}	27.0	30.5	34.4	38.8	41.6	43.5	45.3

0.850, and 0.950, and $p_1 = 12$, $p_2 = 8$, and $p_3 = 2$ with a small amount of random error added [10].

*First-Order Irreversible Series Reaction (*IRR1*).* A system of ODE with two dependent variables and two parameters (Section 6.1 in [19] and Section 7.1 in [10]) is given:

$$\begin{aligned} y_1' &= -p_1 y_1, \\ y_2' &= p_1 y_1 - p_2 y_2, \end{aligned} \tag{A.4}$$

for $t \in [0, 1]$, with initial conditions $y_1(0) = 1, y_2(0) = 0$. Lower and upper bounds on the parameters are $[p_i^{\text{L}}, p_i^{\text{U}}] = [0, 10]$, $i = 1, 2$, and the experimental data in Table 23 have been generated using $t = [0.1, 0.2, 0.3, \ldots, 0.9, 1.0]$ and $p_1 = 5$, $p_2 = 1$ with no added error [10].

*First-Order Irreversible Series Reaction (*IRR2*).* This is the system of ODE of the previous example with different initial conditions, different upper bounds, and a different time interval (Example 1 in [26]):

$$\begin{aligned} y_1' &= -p_1 y_1, \\ y_2' &= p_1 y_1 - p_2 y_2, \end{aligned} \tag{A.5}$$

for $t \in [0, 10]$, with the initial conditions $y_1(0) = 2$, $y_2(0) = 0$. The lower and upper bounds on the parameters are $[p_i^{\text{L}}, p_i^{\text{U}}] = [0, 1]$, $i = 1, 2$. Experimental data have been generated using the time instants $t = [0.5, 1, 1.5, 2, 2.5, \ldots, 8.5, 9, 9.5, 10]$ and $p_1 = 0.8$ and $p_2 = 0.3$ with no added error [26] (see Table 24).

*First-Order Reversible Series Reaction (*REV*).* This is a system of three ODE with four parameters available in Section 7.2 of [10]:

$$\begin{aligned} y_1' &= -p_1 y_1 + p_2 y_2, \\ y_2' &= p_1 y_1 - (p_2 + p_3)\, y_2 + p_4 y_3, \\ y_3' &= p_3 y_2 - p_4 y_3, \end{aligned} \tag{A.6}$$

for $t \in [0, 1]$, with initial conditions $y_1(0) = 1$, $y_2(0) = 0$, and $y_3(0) = 0$. Lower and upper bounds on the parameters are $[p_i^{\text{L}}, p_i^{\text{U}}] = [0, 10]$, $i = 1, 2$, and $[p_i^{\text{L}}, p_i^{\text{U}}] = [10, 50]$, $i = 3, 4$. Two sets of data have been generated using $t = [0.05, 0.1, 0.15, 0.2, 0.25, \ldots, 0.9, 0.95, 1.0]$ and $p_1 = 4$, $p_2 = 2$, $p_3 = 40$, and $p_4 = 20$ [10]. The first set has no added error (see Table 25) and yields the instance denoted by REV_A. The second set of data shown in Table 26 has been obtained by adding a small amount of random errors [10] and yields the denoted instance REV_B.

*Theoretical Kinetic Model Described by Three Reactions and Based on Three Parameters (*THEO1*).* This system of five ODE depends on three parameters (Example 2 in [26]):

$$\begin{aligned} y_1' &= -p_1 y_1 y_2 - p_2 y_1 y_3 - p_3 y_1 y_4, \\ y_2' &= -p_1 y_1 y_2, \\ y_3' &= p_1 y_1 y_2 - p_2 y_1 y_3, \\ y_4' &= p_2 y_1 y_3 - p_3 y_1 y_4, \\ y_5' &= p_3 y_1 y_4, \end{aligned} \tag{A.7}$$

for $t \in [0, 501]$, with initial conditions $y_1(0) = 0.02090$, $y_2(0) = 0.00697$, $y_3(0) = 0$, $y_4(0) = 0$, and $y_5(0) = 0$. Lower and upper bounds on the parameters are $[p_i^{\text{L}}, p_i^{\text{U}}] = [0.1, 30]$, $i = 1, 2, 3$. Only the concentration of component A $(= y_1)$ was measured. Thus, data for component y_1 alone for 22 time instants, shown in Table 27, are available in [26].

*Theoretical Kinetic Model Based on Two Parameters (*THEO2*).* The following system has five dependent variables and two parameters (see Example 3 in [26]):

$$\begin{aligned} y_1' &= -p_1 y_1 y_2 - p_2 y_1 y_4, \\ y_2' &= -p_1 y_1 y_2, \\ y_3' &= p_1 y_1 y_2, \\ y_4' &= p_1 y_1 y_2 + p_2 y_1 y_4, \\ y_5' &= p_2 \sqrt{y_1 y_4}, \end{aligned} \tag{A.8}$$

for $t \in [0, 10]$, with initial conditions $y_1(0) = 1.5$, $y_2(0) = 1$, $y_3(0) = 0$, $y_4(0) = 0$, and $y_5(0) = 0$. The lower and upper bounds on the parameters are $[p_i^{\text{L}}, p_i^{\text{U}}] = [0.01, 1]$, $i = 1, 2$. Experimental data, shown in Table 28, have been generated

Table 22: Error-added experimental values for 20 time instants, available in [10].

	0.025	0.050	0.075	0.100	0.125	0.150	0.175
y_1^{obs}	0.7307	0.5982	0.4678	0.4267	0.3436	0.3126	0.2808
y_2^{obs}	0.1954	0.2808	0.3175	0.3047	0.2991	0.2619	0.2391
	0.200	0.225	0.250	0.300	0.350	0.400	0.450
y_1^{obs}	0.2692	0.2210	0.2122	0.1903	0.1735	0.1615	0.1240
y_2^{obs}	0.2210	0.1898	0.1801	0.1503	0.1030	0.0964	0.0581
	0.500	0.550	0.650	0.750	0.850	0.950	
y_1^{obs}	0.1190	0.1109	0.0890	0.0820	0.0745	0.0639	
y_2^{obs}	0.0471	0.0413	0.0367	0.0219	0.0124	0.0089	

Table 23: Generated experimental values for 10 time instants, available in [10].

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
y_1^{obs}	0.606	0.368	0.223	0.135	0.082	0.050	0.030	0.018	0.011	0.007
y_2^{obs}	0.373	0.564	0.647	0.669	0.656	0.624	0.583	0.539	0.494	0.451

Table 24: Generated experimental values for 20 time instants, available in [26].

	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
y_1^{obs}	1.341	0.899	0.602	0.404	0.271	0.181	0.122	0.082	0.055	0.037
y_2^{obs}	0.609	0.933	1.077	1.110	1.079	1.011	0.925	0.833	0.742	0.655
	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
y_1^{obs}	0.025	0.016	0.011	0.007	0.005	0.003	0.002	0.001	0.001	0.001
y_2^{obs}	0.575	0.503	0.438	0.380	0.329	0.285	0.246	0.213	0.184	0.158

Table 25: Error-free generated experimental values for 20 time instants, available in [10].

	0.05	0.1	0.15	0.2	0.25	0.3	0.35
y_1^{obs}	0.8241	0.6852	0.5747	0.4867	0.4166	0.3608	0.3164
y_2^{obs}	0.0937	0.1345	0.1654	0.1899	0.2094	0.2249	0.2373
y_3^{obs}	0.0821	0.1802	0.2598	0.3233	0.3738	0.4141	0.4461
	0.4	0.45	0.5	0.55	0.6	0.65	0.7
y_1^{obs}	0.2810	0.2529	0.2304	0.2126	0.1984	0.1870	0.1780
y_2^{obs}	0.2472	0.2550	0.2613	0.2662	0.2702	0.2733	0.2759
y_3^{obs}	0.4717	0.4920	0.5082	0.5210	0.5313	0.5395	0.5460
	0.75	0.8	0.85	0.9	0.95	1.0	
y_1^{obs}	0.1709	0.1651	0.1606	0.1570	0.1541	0.1518	
y_2^{obs}	0.2779	0.2794	0.2807	0.2817	0.2825	0.2832	
y_3^{obs}	0.5511	0.5553	0.5585	0.5612	0.5632	0.5649	

Table 26: Error-added experimental values for 20 time instants, available in [10].

	0.05	0.1	0.15	0.2	0.25	0.3	0.35
y_1^{obs}	0.8261	0.6782	0.5721	0.4817	0.4226	0.3698	0.3114
y_2^{obs}	0.0917	0.1335	0.1644	0.1939	0.2111	0.2229	0.2313
y_3^{obs}	0.0826	0.1772	0.2628	0.3213	0.3598	0.4201	0.4511
	0.4	0.45	0.5	0.55	0.6	0.65	0.7
y_1^{obs}	0.2710	0.2499	0.2354	0.2216	0.1974	0.1890	0.1780
y_2^{obs}	0.2398	0.2510	0.2703	0.2602	0.2732	0.2733	0.2769
y_3^{obs}	0.4797	0.4990	0.5122	0.5200	0.5281	0.5305	0.5500
	0.75	0.8	0.85	0.9	0.95	1.0	
y_1^{obs}	0.1729	0.1701	0.1606	0.1490	0.1531	0.1568	
y_2^{obs}	0.2709	0.2754	0.2797	0.2817	0.2825	0.2792	
y_3^{obs}	0.5601	0.5533	0.5485	0.5612	0.5632	0.5599	

Table 27: Experimental data for 22 time instants and for y_1 alone, available in [26].

	4.5	8.67	12.67	17.75	22.67	27.08
y_1^{obs}	0.0514	0.01422	0.01335	0.01232	0.01181	0.01139
	32.00	36.00	46.33	57.00	69.00	76.75
y_1^{obs}	0.01092	0.01054	0.00978	0.009157	0.008594	0.008395
	90.00	102.00	108.00	147.92	198.00	241.75
y_1^{obs}	0.007891	0.00751	0.00737	0.006646	0.005883	0.005322
	270.25	326.25	418.00	501.00		
y_1^{obs}	0.00496	0.004518	0.004075	0.003372		

Table 28: Experimental data available in [26] for two state variables only (y_1 and y_2).

	1	2	3	4	5	6	7
y_1^{obs}	1.1529	0.9333	0.7806	0.6675	0.5801	0.5104	0.4535
y_2^{obs}	0.6747	0.4944	0.3828	0.3083	0.2558	0.2173	0.1881
	8	9	10				
y_1^{obs}	0.4060	0.3658	0.3314				
y_2^{obs}	0.1653	0.1473	0.1327				

Table 29: Error-added generated experimental values for 10 time instants.

	1	2	3	4	5	6	7
y_1^{obs}	0.8170	0.8525	1.2451	1.0423	0.7868	0.9834	1.2771
y_2^{obs}	1.1142	0.8876	0.9619	1.1419	1.0144	0.8694	1.0247
	8	9	10				
y_1^{obs}	0.8776	0.7760	1.1412				
y_2^{obs}	1.1231	0.9538	0.8797				

Table 30: Error-added experimental values for 10 time instants, available in [10].

	1	2	3	4	5	6	7
y_1^{obs}	0.7990	0.8731	1.2487	1.0362	0.7483	1.0024	1.2816
y_2^{obs}	1.0758	0.8711	0.9393	1.1468	1.0027	0.8577	1.0274
	8	9	10				
y_1^{obs}	0.8944	0.7852	1.1527				
y_2^{obs}	1.1369	0.9325	0.9074				

using $t = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ and $p_1 = 0.3$ and $p_2 = 0.1$ with added small error and are available in [26]. Only the concentrations of components A ($=y_1$) and B ($=y_2$) were measured.

Lotka-Volterra Predator-Prey Model (`VOL`*).* This is a system of ODE with two dependent variables and two parameters (see Section 6.4 in [19] and Section 7.6 in [10]):

$$\begin{aligned} y_1' &= p_1 y_1 (1 - y_2), \\ y_2' &= p_2 y_2 (y_1 - 1), \end{aligned} \tag{A.9}$$

for $t \in [0, 10]$, with initial conditions $y_1(0) = 1.2$ and $y_2(0) = 1.1$ and lower and upper bounds on the parameters: $[p_i^L, p_i^U] = [0.1, 10]$, $i = 1, 2$. Experimental data in Table 29 have been generated using $t = [1, 2, 3, 4, \ldots, 9, 10]$ and $p_1 = 3$ and $p_2 = 1$ with a small amount of random errors added, according to the normal distribution $N(0, 0.01)$, yielding the instance `VOL_A`. A second set of experimental data, shown in Table 30, is available in [10] and has also been tested, corresponding to the herein denoted instance `VOL_B`.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the financial support of CIDEM, R&D Unit, funded by the Portuguese Foundation for the Development of Science and Technology (FCT), Ministry of Science, Technology and Higher Education, under the Project UID/EMS/0615/2016, and of COMPETE:

POCI-01-0145-FEDER-007043 and FCT within the Projects UID/CEC/00319/2013 and UID/MAT/00013/2013.

References

[1] J. R. Banga, E. Balsa-Canto, C. G. Moles, and A. A. Alonso, "Dynamic optimization of bioreactors: a review," *Proceedings of the Indian National Science Academy. Part A. Physical Sciences*, vol. 69, no. 3-4, pp. 257–265, 2003.

[2] M. Rodriguez-Fernandez, J. A. Egea, and J. R. Banga, "Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems," *BMC Bioinformatics*, vol. 7, article no. 483, 2006.

[3] L. T. Biegler, *Nonlinear Programming: Concepts, Alg orithms and Applications to Chemical Processes*, Society for Industrial and Applied Mathematics, Cambridge University Press, Pittsburgh, Pa, USA, 2010.

[4] C. A. Floudas, P. M. Pardalos, C. S. Adjiman et al., *Handbook of Test Problems in Local and Global Optimization. In Series in Nonconvex Optimization and its Applications*, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1999.

[5] A. Gábor and J. R. Banga, "Robust and efficient parameter estimation in dynamic models of biological systems," *BMC Systems Biology*, vol. 9, no. 1, article no. 74, 2015.

[6] K. Sörensen, "Metaheuristics – the metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015.

[7] K. Sörensen and F. Glover, *Encyclopedia of Operations Research and Management Science*, K. Sörensen, F. Glover, S. I. Gass, and M. C. Fu, Eds., Springer, New York, NY, USA, 3rd edition, 2013.

[8] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[9] A. Zhigljavsky and A. Zilinskas, *Stochastic Global Optimization. Springer Optimization and Its Applications*, vol. 9 of *Springer*, New York, NY, USA, 2008.

[10] W. R. Esposito and C. A. Floudas, "Global optimization for the parameter estimation of differential- algebraic systems," *Industrial & Engineering Chemistry Research*, vol. 39, no. 5, pp. 1291–1310, 2000.

[11] C. G. Moles, P. Mendes, and J. R. Banga, "Parameter estimation in biochemical pathways: A comparison of global optimization methods," *Genome Research*, vol. 13, no. 11, pp. 2467–2474, 2003.

[12] J. A. Egea, M. Rodríguez-Fernández, J. R. Banga, and R. Martí, "Scatter search for chemical and bio-process optimization," *Journal of Global Optimization*, vol. 37, no. 3, pp. 481–503, 2007.

[13] A. M. A. C. Rocha, M. C. Martins, M. F. P. Costa, and E. M. G. P. Fernandes, "Direct sequential based firefly algorithm for the a-pinene isomerization problem," in *ICCSA 2016 Part I, Lecture Notes in Computer Science*, B. O. Gervasi, S. Murgante, A. M. Misra et al., Eds., vol. 9786, pp. 386–401, 2016.

[14] A. Miró, C. Pozo, G. Guillén-Gosálbez, J. A. Egea, and L. Jiménez, "Deterministic global optimization algorithm based on outer approximation for the parameter estimation of nonlinear dynamic biological systems," *BMC Bioinformatics*, vol. 13, no. 1, article no. 90, 2012.

[15] A. Abdullah, S. Deris, S. Anwar, and S. N. V. Arjunan, "An Evolutionary Firefly Algorithm for the Estimation of Nonlinear Biological Model Parameters," *PLoS ONE*, vol. 8, no. 3, Article ID e56310, 2013.

[16] R. Angira and A. Santosh, "Optimization of dynamic systems: A trigonometric differential evolution approach," *Computers & Chemical Engineering*, vol. 31, no. 9, pp. 1055–1063, 2007.

[17] R. Angira, "A comparative study of differential evolution algorithms for estimation of kinetics parameters," *Advanced Modeling and Optimization*, vol. 14, no. 1, pp. 135–145, 2012.

[18] I. Papamichail and C. S. Adjiman, "Global optimization of dynamic systems," *Computers & Chemical Engineering*, vol. 28, no. 3, pp. 403–415, 2004.

[19] Y. Lin and M. A. Stadtherr, "Deterministic global optimization for parameter estimation of dynamic systems," *Industrial & Engineering Chemistry Research*, vol. 45, no. 25, pp. 8438–8448, 2006.

[20] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, vol. 5792 of *Lecture Notes in Comput. Sci.*, pp. 169–178, Springer, Berlin, 2009.

[21] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[22] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[23] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[24] A.-R. Hedar and M. Fukushima, "Tabu search directed by direct search methods for nonlinear global optimization," *European Journal of Operational Research*, vol. 170, no. 2, pp. 329–349, 2006.

[25] R. Hooke and R. Jeeves, "Direct search solution of numerical and statistical problems," *Journal of the ACM*, vol. 8, pp. 212–229, 1961.

[26] M. Podmajersk and M. Podmajerský, *Parameter Estimation in Processes from Experimental Data. Master Thesis [Master, thesis]*, Slovak Technical University in Bratislava, 2007.

[27] B. Srinivasan, S. Palanki, and D. Bonvin, "Dynamic optimization of batch processes I. Characterization of the nominal solution," *Computers & Chemical Engineering*, vol. 27, no. 1, pp. 1–26, 2003.

[28] I. Fister Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *Elektrotehniški Vestnik*, vol. 80, no. 3, pp. 1–7, 2013.

[29] I. Fister, I. Fister Jr., X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation*, vol. 13, no. 1, pp. 34–46, 2013.

[30] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.

[31] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[32] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[33] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm

and applications," *Artificial Intelligence Review*, vol. 42, pp. 21–57, 2014.

[34] D. Cvijovic and J. Klinowski, "Taboo search: an approach to the multiple minima problem," *American Association for the Advancement of Science: Science*, vol. 267, no. 5198, pp. 664–666, 1995.

[35] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[36] R. L. Iman, "Approximations of the critical region of the friedman statistic," *Communications in Statistics—Theory and Methods*, vol. 9, no. 6, pp. 571–595, 1980.

[37] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[38] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.