Escherichia coli Cultivation Process Modelling Using ABC-GA Hybrid Algorithm

Olympia Roeva 1,*, Dafina Zoteva 1,2 and Velislava Lyubenova 3

1 Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria; dafinaz@fmi.uni-sofia.bg
2 Faculty of Mathematics and Informatics, Sofia University St. Kliment Ohridski, 1164 Sofia, Bulgaria
3 Department of Mechatronic Bio/Technological Systems, Institute of Robotics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria; v_lyubenova@ir.bas.bg
* Correspondence: olympia@biomed.bas.bg

Abstract: In this paper, the artificial bee colony (ABC) algorithm is hybridized with the genetic algorithm (GA) for a model parameter identification problem. When dealing with real-world and large-scale problems, it becomes evident that concentrating on a sole metaheuristic algorithm is somewhat restrictive. A skilled combination between metaheuristics or other optimization techniques, a so-called hybrid metaheuristic, can provide more efficient behavior and greater flexibility. Hybrid metaheuristics combine the advantages of one algorithm with the strengths of another. ABC, based on the foraging behavior of honey bees, and GA, based on the mechanics of nature selection, are among the most efficient biologically inspired population-based algorithms. The performance of the proposed ABC-GA hybrid algorithm is examined, including classic benchmark test functions. To demonstrate the effectiveness of ABC-GA for a real-world problem, parameter identification of an Escherichia coli MC4110 fed-batch cultivation process model is considered. The computational results of the designed algorithm are compared to the results of different hybridized biologically inspired techniques (ant colony optimization (ACO) and firefly algorithm (FA))—hybrid algorithms as ACO-GA, GA-ACO and ACO-FA. The algorithms are applied to the same problems—a set of benchmark test functions and the real nonlinear optimization problem. Taking into account the overall searchability and computational efficiency, the results clearly show that the proposed ABC–GA algorithm outperforms the considered hybrid algorithms.

Keywords: artificial bee colony; genetic algorithm; hybrid metaheuristic; parameter identification; benchmark test functions; fed-batch cultivation processes; E. coli

1. Introduction

Although many different methods of global optimization have been developed, the efficiency of an optimization method is always determined by the specific nature of the particular problem. Parameter optimization of cellular dynamics models has become a field of particular interest, as the problem is widely applicable. Identification of the parameters of a nonlinear dynamic model is more difficult than the linear case since there are no general analytical results.

Nature-inspired metaheuristic methods still receive a great deal of attention nowadays [1–7]. Among population-based nature-inspired metaheuristics are genetic algorithms (GA), ant colony optimization (ACO), firefly algorithm (FA), cuckoo search (CS), bat algorithm (BA), monarch butterfly optimization (MBO), whale optimization algorithm (WOA), salp swarm algorithm (SSA), emperor penguin optimizer (EPO), etc., all of which use multiple agents or “particles”.

Typically, metaheuristic methods can successfully solve complex identification problems. Even more effective behaviors can be achieved through a combination of different metaheuristic techniques, the so-called hybrid metaheuristics [8].
The main purpose of hybrid algorithms is to exploit the advantages of different optimization strategies while avoiding their disadvantages. By choosing an appropriate combination of metaheuristic techniques, better algorithm performance can be achieved in solving hard optimization problems [9]. There are many hybridization techniques that have been proven successful for various applications. In [10], the authors propose a hybrid metaheuristic algorithm for clustering, which combines ant brood sorting with tabu search, with emphasis on flexibility and less specificity. Rosle et al. [11] propose a hybrid of particle swarm optimization and harmony search, which outperforms the simulated annealing and the downhill simplex method. A novel hybrid integrating a support vector machine (SVM) model with GA is discussed in [12]. The designed hybrid algorithm shows the smallest amount of outlier data and the best predictive stability. Another successful hybrid, combining GA and Variable Neighborhood Search, is proposed in [13]. The algorithm’s feasibility and superiority are demonstrated in a real-world case.

An efficient combination of SVM and MBO is discussed in [14]. Its application on an imbalanced disease classification problem demonstrates improvements using the developed hybrid method. Liu et al. [15] propose a hybrid WOA enhanced with Lévy flight and differential evolution (DE). The experimental results of its application to solve a job shop scheduling problem show the superior performance of the hybrid. A combination of the advantages of the SSA and PSO algorithms are introduced in [16] and applied to the optimal power flow problem. The proposed hybrid algorithm provides more efficient solutions even for conflict constraints. A hybrid metaheuristic optimization approach between EPO and ESA is proposed in [17]. A test analysis on 53 benchmark functions shows the effectiveness and efficiency of the designed hybrid algorithm.

Following the literature, it is evident that many metaheuristic algorithms could be hybridized to improve the algorithms’ performance for a particular problem. Another example is the artificial bee colony (ABC) algorithm—one of the promising biologically inspired metaheuristic approaches introduced by Karaboga in 2005 [18] for optimizing multi-variable and multimodal numerical functions. The known results indicate that ABC can be used efficiently for various optimization problems [19–23].

Some hybrid algorithms involving ABC and its modifications that aim to improve the algorithm’s performance are the following: an ABC and quantum evolutionary algorithm hybrid is proposed in [24] for solving continuous optimization problems. The authors use ABC to increase the local search capacity as well as the randomness of the populations. A hybrid between ABC and the monarch butterfly optimization for numerical optimization problems is proposed in [25]. The hybrid is designed to boost the exploration versus exploitation balance of the original algorithms. Again, for numerical function optimization, a hybrid ABC algorithm with DE and free search is proposed in [26]. The resulting hybrid shows excellent convergence and global searchability. A powerful hybrid algorithm that combines the features of both optimization algorithms ABC and BA are proposed in [27]. An ABC algorithm with random location-updating is considered in [28]. The designed algorithm shows stronger exploration ability, improved convergence speed and optimization precision. In [29], the authors introduce an algorithm in which the crossover operator of GA is involved in improving the exploration of the canonical ABC. Crossover and mutation operators are introduced in a hybrid ABC to reduce the computational resources [30].

This literature review shows that, as a result of the ABC hybridization, the algorithm’s performance is improved, and optimal solutions to different problems are achieved. This fact, and the known successful applications of GA, particularly for parameter identification of nonlinear models of cultivation processes, inspired the authors to work on the development of a new hybrid metaheuristic algorithm between ABC and GA.

The proposed hybrid algorithm in this paper, named ABC-GA, is designed to improve both the exploration and exploitation and thus present a powerful and efficient algorithm for real-world numerical optimization problems. As a result, the speed and precision of the algorithm’s convergence are also improved.
The performance of the proposed ABC-GA hybrid algorithm is examined with two different test groups, including classic benchmark test functions and a real nonlinear optimization problem. The ABC-GA is employed for a nonlinear model parameter identification problem. As a case study, an E. coli MC4110 fed-batch cultivation process model is considered. The optimization problem is solved using real experimental data.

The ABC-GA results were compared with different hybrid algorithms from the literature, between ACO, GA and FA, for each test group. The main contributions of the research can be summarized as below:

1. Combining the features of ABC and GA, a new hybrid algorithm ABC-GA is proposed and tested on classic benchmark test functions and a real-world problem.
2. The superiority of the designed hybrid ABC-GA is shown based on the comparison of the simulation results with other hybrid metaheuristic algorithms for the problems of the two different test groups.
3. Using the hybrid metaheuristic algorithm ABC-GA the optimal parameters’ values of a nonlinear mathematical model of an E. coli cultivation process are estimated.

The paper is organized as follows. In Section 2, the background of the ABC-GA hybrid algorithm is given. The performance of the designed hybrid on different unconstrained optimization problems is studied in Section 3. The considered real-world problem—a model parameter identification—is outlined in Section 4, and the numerical results from the identification are presented and discussed. Concluding remarks are given in Section 5.

2. ABC-GA Hybrid Algorithm

Based on the foraging behavior of the honey bees, the ABC algorithm was introduced by Karaboga for the purpose of numerical optimization problems [18]. ABC operates on a set of food sources representing potential solutions to the problem under consideration. The initial set of solutions is generated during the initialization phase of the algorithm when the initial values of the algorithm’s parameters are set.

The ABC algorithm evolves in three phases on each iteration: employed bees’ phase, onlookers’ phase and scouts’ phase [31], during which a new set of food sources is formed. The employed bees search for new food sources around those stored in their memory. Based on their evaluations, the newly found food sources may replace the old ones. Only certain food sources can be modified during the onlookers’ phase. They are selected based on the probability values associated with them. The last phase is when an abandoned food source is replaced by a new random one found by a scout.

These three phases are repeated for a predetermined number of cycles called maximum cycle number (MCN). The best food source represents a reasonable solution to the considered problem.

GA was developed to model adaptation processes using a recombination operator with a mutation in the background [32]. GA maintains a population of individuals. Each individual represents a potential solution to the problem. Each solution is evaluated, so a new population is formed by selecting more fit individuals. In order to form new solutions, some elements of the new population undergo transformations by means of “genetic” operators. These are unary transformations (mutation type) which create new individuals by a small change in a single individual, and higher-order transformations (crossover type), which create new individuals by combining parts from several individuals.

After a certain number of generations, the algorithm converges and is expected to best represent a near-optimum (reasonable) solution to the problem under consideration.

The proposed hybrid algorithm is a collaborative combination between the ABC algorithm and GA. The aim is to avoid the poor convergence rate of the ABC algorithm, a disadvantage reported in [26,32,33]. The population-based GA randomly generates an initial population that can be very far from the optimal solution and may require a lot of iterations to draw close to it. To overcome these limitations, ABC is applied to the generated initial solutions for a few iterations. The ABC outcomes are then used as an
initial population of GA. Thus, GA starts with a population much closer to the optimal one compared to the randomly generated initial population. The best solution is obtained by the genetic evolution of the ABC, and the result uses fewer computational resources. This integration aims to provide a proper balance between exploration and exploitation (intensification and diversification).

The main steps of the ABC-GA hybrid algorithm are presented in Figure 1.

During the initialization phase of the algorithm, values of the input parameters of ABC and GA are set. The optimization problem is defined, as well as the problem parameters and their bounds.
The initial solutions are generated using Equation (1). Each solution is a D-dimensional vector \( x_i = \{ x_{i1}, x_{i2}, \ldots, x_{iD} \} \), limited by the lower and upper bounds of the corresponding parameter of the optimization problem:

\[
x^j_i = x^j_{\min} + \text{rand}(0, 1) (x^j_{\max} - x^j_{\min})
\]

where \( i \in [1; SN], j \in [1; D], x^j_{\min} \) and \( x^j_{\max} \) are the lower and the upper bounds of the dimension \( j \).

ABC utilizes this set of initial solutions in an attempt to get closer to finding the optimal solution to the optimization problem. ABC runs for a predetermined MCN, and each cycle evolves in three phases, defined in [31].

The search for new food sources (solutions) during the employed bees’ phase is based on Equation (2):

\[
v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj})
\]

where \( j \) is a random integer number in the range \([1, D]\); \( k \) is randomly selected index; \( k \in [1; SN], k \neq i; \phi_{ij} \in [-1, 1] \) is a random number.

The onlookers’ search around certain food sources is based on the same Equation (3). The food sources are selected taking into account a probability value \( p_i \) associated with each food source, calculated by the roulette wheel selection:

\[
p_i = \frac{f_i}{\sum_{n=1}^{SN} f_n}
\]

where \( f_i \) is the fitness value of the solution \( x_i, i \in [1; SN] \).

During the greedy selection in the employed bees’ phase and the onlookers’ phase, the number of trials for each food source is updated when it is not replaced by a better one. When the trials exceed a predefined limit, the corresponding food source is abandoned and replaced by a new food source, randomly generated using Equation (1). This new food source represents a food source found by a scout.

The best solutions accumulated by ABC for \( n \) number of runs, where \( n \) is the population size of GA, are used as the initial population \( Pop_0 \) of GA. GA iteratively improves the existing population by selecting and reproducing parents for a certain number of generations.

The selection of parents to produce next generations is performed based on the fitness of the individuals. The selection method for this particular case is the roulette wheel selection. Two genetic operators—crossover and mutation, are applied later to generate new individuals. The extended intermediate recombination and real value mutation employed here are defined in the following way.

The crossover operator combines the genetic material of two parents. The recombination process is unconditional. Let \( x \) and \( y \) be two \( D \)-dimensional vectors denoting the parents from the current population. Let \( z = \{ z^1, z^2, \ldots, z^D \} \) be the result of the recombination. The elements of \( z \) are generated using Equation (4).

\[
z^j = x^j + \alpha^j (y^j - x^j)
\]

where \( j \in [1; D]; \alpha^j \in [-\delta; 1 + \delta] \) is chosen with uniform probability. \( \delta \) indicates to what degree an offspring can be generated out of the parents’ s scope; usually, the value of \( \delta \) is 0.25.

The mutation operator is applied to each element \( x = \{ x^1, x^2, \ldots, x^D \} \) with a probability inversely proportional to the number of dimensions \( D \). Let \( z = \{ z^1, z^2, \ldots, z^D \} \) be the result of the mutation. The elements of \( z \) are evaluated using Equation (5).

\[
z^j = x^j + \text{sign} \cdot \text{range}^j \cdot \delta
\]
where \( j \in [1; D] \); \( \text{sign} \in \{-1, +1\} \), chosen with equal probability; \( \text{range}^d = \rho (x_{\text{max}}^d - x_{\text{min}}^d) \), \( \rho \in [0.1; 0.5] \) is the mutation range; \( \delta = \sum_{i=0}^{k-1} q_i 2^{-i} \), \( q_i \in \{0, 1\} \) from a Bernoulli probability distribution and \( k \in \{4, 5, \ldots, 20\} \) is the mutation precision related to the minimal step size and the distribution of mutation steps in the mutation range.

The fitness of each individual in the population is evaluated. Offspring with a better fitness evaluation move to the next generation until the maximum number of generations \( \text{MaxGen} \) is reached.

The optimal solution to the particular problem is the best individual from the last generation of GA. Since the initial population is not random but much closer to the optimal sought solution, GA converges to the optimal solution in much fewer iterations.

The hybrid ABC-GA proposed here applies the two algorithms, ABC and GA, sequentially. Therefore, the computational complexity of ABC and GA should be considered separately when the complexity of the hybrid is calculated.

For a given problem, let \( O(f) \) be the computational complexity of its fitness function evaluation. The complexity of the standard ABC can be expressed as \( O(\text{MCN} \times \text{NP} \times f) \). Since ABC is required to generate an initial population for GA, the algorithm is executed \( n \) number of times, where \( n \) is the population size of GA. The computational complexity of GA can be evaluated as \( O(\text{MaxGen} \times n \times f) \). The overall complexity of the ABC-GA hybrid algorithm can then be evaluated as \( O((\text{MCN} \times \text{NP} + \text{MaxGen}) \times n \times f) \).

It should be noted, however, that the values of the parameters \( \text{MCN}, \text{NP} \) and \( \text{MaxGen} \) used in the hybrid ABC-GA are far smaller than the values used in the standard case. The reason is that the ABC algorithm is used to give an initial push to GA, to trigger GA and the search for the optimal solution of the problem to start from a closed position rather than a random one. This way, GA needs far fewer steps to complete the search.

In order to improve the computation precision in the hybrid ABC-GA algorithm, the complexity of the algorithm is sacrificed.

### 3. ABC-GA Hybrid Algorithm Performance on Different Unconstrained Optimization Problems

The proposed algorithm ABC-GA is tested on nine well-known benchmark functions listed in Table 1. All benchmark functions are considered with a dimension of 30 except for Wood’s function (dimension of 4).

#### Table 1. Benchmark unconstrained optimization problems.

| Function      | Definition                                                                 | Range       |
|---------------|---------------------------------------------------------------------------|-------------|
| Powell        | \( F(x) = \sum_{i=1}^{D} |x_i| \)                                                                          | \((-10, 10)\) |
| Sphere        | \( F(x) = \sum_{i=1}^{D} x_i^2 \)                                         | \((-\sqrt{5}, \sqrt{5})\) |
| Wood’s        | \( F(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + 90(x_3^2 - x_4)^2 + \ldots \) | \((-\sqrt{5}, \sqrt{5})\) |
| Schaffer 1    | \( F(x) = 0.5 + \frac{\text{max}(x_1^2 + x_2^2)}{10000} \)               | \((-100, 100)\) |
| Rastrigin     | \( F(x) = 10 + \sum_{i=1}^{D} (x_i^2 - 10 \cos(2\pi x_i)) \)             | \((-5.12, 5.12)\) |
| Griewank      | \( F(x) = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(x_i/\sqrt{D}) \) | \((-600, 600)\) |
| Extended cube | \( F(x) = \sum_{i=1}^{D} (x_{i-1}^2 + (1-x_i)^2) \)                       | \((-100, 100)\) |
| Ackley        | \( F(x) = -a \exp(-h \sqrt{\sum_{i=1}^{D} x_i^2}) - \exp\left(\frac{h}{D} \sum_{i=1}^{D} \cos(x_i)\right) + a + \exp(1) \) | \((-35, 35)\) |
| Rosenbrock    | \( F(x) = \sum_{i=1}^{D} \left[ b (x_{i-1}^2 + (a-x_i)^2) \right] \)    | \((-100, 100)\) |
To verify the performance of ABC-GA, the obtained results are compared with the hybrid algorithms ACO-GA and GA-ACO [34] and ACO-FA [35]. The common parameter settings of the hybrid algorithms used in all test sets are provided in Table 2. The parameters were chosen to ensure enough configuration to find the optimal solution.

Table 2. Parameters setting of algorithms.

| Algorithm | Parameter/Function | Value                  |
|-----------|--------------------|------------------------|
| ABC       | population size $N$P | 100                    |
|           | number of iterations $MCN$ | 100                   |
|           | number of food sources $SN$ | 30                    |
|           | limit               | 100                    |
|           | Probability         | $p_i = f_i / \sum_{n=1}^{SN} f_n$ |
| GA        | population size $n$ | 100                    |
|           | number of generations $MaxGen$ | 100                   |
|           | generation gap      | 0.97                   |
|           | crossover rate      | 0.70                   |
|           | mutation rate       | 0.01                   |
|           | chromosome representation | real valued          |
|           | selection function  | roulette wheel selection |
|           | crossover function  | extended intermediate recombination |
|           | mutation function   | real-value mutation (Breeder GA) |
| FA        | $\beta_0$           | 0.3                    |
|           | $\gamma$            | 0.5                    |
|           | $\alpha$            | 0.1                    |
|           | number of fireflies  | 100                    |
|           | number of iterations | 100                    |
| ACO       | $\rho$              | 0.1                    |
|           | $\alpha$            | 0.8                    |
|           | $b$                 | 0.8                    |
|           | number of ants      | 100                    |
|           | number of iterations | 100                    |

The performance of the four compared hybrid algorithms is evaluated by calculating the best value, the mean value and the standard deviation (SD). The obtained results are presented in Table 3. The best and mean values for each problem are highlighted in bold.

According to Table 3, ABC-GA produced the best value for all nine benchmark functions and the best mean value for eight of them. For Griewank, the best mean value was obtained by the hybrid ACO-GA.

As can be seen for the unimodal Sphere function, the convergence accuracy and stability of ABC-GA are better than those of GA-ACO and ACO-FA. Only the results achieved by GA-ACO are close to the results of ABC-GA. The same results are obtained for the complex unimodal Rosenbrock. However, for the multimodal Rastrigin, Ackley, Griewank and unimodal Schaffer #1 mostly ACO-FA does not produce good results. In the case of Griewank, GA-ACO and ACO-FA results are significantly worse, too. The best mean values of GA-ACO and ACO-FA are $9.71 \times 10^{-3}$ and $8.70 \times 10^{-7}$, respectively, compared to the mean ABC-GA value of $1.08 \times 10^{-15}$. The ACO-FA shows the worse performance for all considered benchmark test functions. The results obtained by ABC-GA demonstrate the achieved good balance between algorithm’s exploration and exploitation.

To illustrate the efficiency of ABC-GA, the convergence of the compared hybrid algorithms towards the optima for 30 runs is presented in Figure 2. Two benchmark functions, Sphere and Powell, are chosen as the most indicative. For the rest of the functions, the superiority of ABC-GA is very clearly shown by the results in Table 3. ACO-FA is not included in the chart because it produces much worse results.
Table 3. Statistical results for the benchmark functions for different algorithms.

| Function     | Algorithm | Min        | Mean         | SD     |
|--------------|-----------|------------|--------------|--------|
| Powell       | ABC-GA    | $3.36 \times 10^{-15}$ | $4.38 \times 10^{-9}$ | $1.09 \times 10^{-8}$ |
|              | ACO-GA    | $2.00 \times 10^{-14}$ | $6.43 \times 10^{-7}$ | $1.19 \times 10^{-6}$ |
|              | GA-ACO    | $7.99 \times 10^{-11}$ | $2.02 \times 10^{-6}$ | $2.69 \times 10^{-6}$ |
|              | ACO-FA    | $7.43 \times 10^{-8}$  | $8.94 \times 10^{-4}$ | $1.11 \times 10^{-3}$ |
| Sphere       | AB-GA     | $2.14 \times 10^{-14}$ | $1.76 \times 10^{-12}$ | $2.10 \times 10^{-12}$ |
|              | ACO-GA    | $3.57 \times 10^{-7}$  | $2.14 \times 10^{-6}$ | $2.63 \times 10^{-6}$ |
|              | GA-ACO    | $3.28 \times 10^{-13}$ | $9.52 \times 10^{-10}$ | $2.67 \times 10^{-9}$ |
|              | ACO-FA    | $5.11 \times 10^{-3}$  | $3.22 \times 10^{-2}$ | $1.54 \times 10^{-2}$ |
| Wood’s       | ABC-GA    | $4.51 \times 10^{-5}$  | $1.23 \times 10^{-2}$ | $1.98 \times 10^{-2}$ |
|              | ACO-GA    | $7.31 \times 10^{-1}$  | $4.86 \times 10^{-1}$ | $3.63 \times 10^{-3}$ |
|              | GA-ACO    | $1.29 \times 10^{-4}$  | $5.70 \times 10^{-1}$ | $1.64 \times 10^{-1}$ |
|              | ACO-FA    | $1.09 \times 10^{-2}$  | $2.50 \times 10^{-1}$ | $2.12 \times 10^{-2}$ |
| Schaffer#1   | ABC-GA    | 0          | 0            | 0      |
|              | ACO-GA    | 0          | $7.99 \times 10^{-10}$ | $4.20 \times 10^{-9}$ |
|              | GA-ACO    | 0          | $7.99 \times 10^{-10}$ | $1.26 \times 10^{-3}$ |
|              | ACO-FA    | $9.24 \times 10^{-6}$ | $1.19 \times 10^{-2}$ | $9.43 \times 10^{-3}$ |
| Rastrigin    | ABC-GA    | 0          | 0            | 0      |
|              | ACO-GA    | 0          | $4.87 \times 10^{-8}$ | $2.48 \times 10^{-7}$ |
|              | GA-ACO    | 0          | $3.32 \times 10^{-2}$ | $1.79 \times 10^{-1}$ |
|              | ACO-FA    | $7.84 \times 10^{-9}$ | $9.81 \times 10^{-7}$ | $1.15 \times 10^{-6}$ |
| Griewank     | ABC-GA    | 0          | $1.08 \times 10^{-15}$ | $5.76 \times 10^{-15}$ |
|              | ACO-GA    | 0          | 0            | 0      |
|              | GA-ACO    | 0          | $9.71 \times 10^{-10}$ | $3.63 \times 10^{-10}$ |
|              | ACO-FA    | $3.27 \times 10^{-8}$ | $8.70 \times 10^{-7}$ | $8.28 \times 10^{-7}$ |
| Extended cube| ABC-GA    | $8.79 \times 10^{-13}$ | $6.49 \times 10^{-7}$ | $1.14 \times 10^{-6}$ |
|              | ACO-GA    | $8.66 \times 10^{-4}$ | $1.4 \times 10^{-1}$  | $1.09 \times 10^{-1}$ |
|              | GA-ACO    | $1.64 \times 10^{-1}$ | $2.03 \times 10^{-1}$ | $2.10 \times 10^{-2}$ |
|              | ACO-FA    | $1.31 \times 10^{-8}$ | $7.02 \times 10^{-3}$ | $1.81 \times 10^{-2}$ |
| Ackley       | ABC-GA    | 0          | 0            | 0      |
|              | ACO-GA    | 0          | $2.78 \times 10^{-6}$ | $8.51 \times 10^{-6}$ |
|              | GA-ACO    | $6.66 \times 10^{-15}$ | $1.10 \times 10^{-13}$ | $9.40 \times 10^{-14}$ |
|              | ACO-FA    | $2.20 \times 10^{-5}$ | $3.64 \times 10^{-4}$ | $2.11 \times 10^{-4}$ |
| Rosenbrock   | ABC-GA    | $6.30 \times 10^{-10}$ | $4.15 \times 10^{-5}$ | $1.24 \times 10^{-4}$ |
|              | ACO-GA    | $4.65 \times 10^{-3}$ | $1.25 \times 10^{-1}$ | $1.21 \times 10^{-1}$ |
|              | GA-ACO    | $2.00 \times 10^{-9}$ | $1.03 \times 10^{-2}$ | $3.09 \times 10^{-2}$ |
|              | ACO-FA    | $1.02 \times 10^{-5}$ | $3.22 \times 10^{-2}$ | $5.01 \times 10^{-2}$ |

(a) Convergence graphics of the Powell function  
(b) Convergence graphics of the Sphere function

Figure 2. Convergence graphics.
The plots show the comparative performance of the ABC-GA hybrid algorithm. In terms of convergence, it can be noticed that ABC-GA has a relatively fast convergence toward its final optimal value compared to the other hybrid algorithms.

4. Parameter Identification Problem of an *E. coli* Fed-Batch Cultivation Process

4.1. Problem Formulation

There is an increasing interest in technologies that maximize the production of various essential enzymes and therapeutic proteins based on *E. coli* cultivation [36–39]. The development of mathematical models for the description, monitoring and control of bioprocesses is a complicated task due to the inherent complexity and non-linearity of biological systems. An important part of building the model is the choice of a specific optimization procedure for parameter identification. The high accuracy of the estimations of the model parameters is essential for successful model development.

The application of the general state-space dynamical model to the fed-batch cultivation process of bacteria *E. coli* leads to the following nonlinear differential equation system [40].

\[
\frac{dX}{dt} = \mu X - \frac{F}{V} X \tag{6}
\]

\[
\frac{dS}{dt} = -\frac{1}{Y_{S/X}} \mu X + \frac{F}{V} (S_{in} - S) \tag{7}
\]

\[
\frac{dV}{dt} = F \tag{8}
\]

where \(X\) is the concentration of biomass, [g/L], \(S\) is the concentration of substrate (glucose), [g/L]; \(F\) is the feeding rate, [L/h]; \(V\) is the volume of the bioreactor, [L]; \(S_{in}\) is the initial glucose concentration in the feeding solution, [g/L]; \(\mu\) is the specific growth rate described by Monod kinetics, \(\mu = \mu_{\text{max}} \frac{S}{S + k_S}\), [1/h]; \(\mu_{\text{max}}\) is the maximum growth rate, [1/h]; \(k_S\) is a saturation constant, [g/L]; \(Y_{S/X}\) is a yield coefficient.

The parameter identification problem considered here is solved using real experimental data of an *E. coli* MC4110 fed-batch cultivation process. Offline measurements of the biomass concentration and online data for glucose concentration are used. A detailed description of the cultivation conditions and experimental data are discussed in [40].

The numerical optimization problem uses a standard objective function—a minimization of the distance measure \(J\) between the experimental data and the model-predicted values of the main process variables:

\[
J = \sum_{i=1}^{n} (X_{\text{mod}}(i) - X_{\text{exp}}(i))^2 + \sum_{i=1}^{n} (S_{\text{mod}}(i) - S_{\text{exp}}(i))^2 \rightarrow \text{min}, \tag{9}
\]

where \(n\) is the number of data points for each process variable \((X, S)\); \(X_{\text{exp}}\) and \(S_{\text{exp}}\) are the biomass and substrate experimental data; \(X_{\text{mod}}\) and \(S_{\text{mod}}\) are the biomass and substrate model predictions for a given set of the model parameters.

Although the objective function seems bi-objective, the preliminary tests show that there is no need to use some weighted method to convert the function into a single objective. The two-process variables \(X\left(\frac{dX}{dt} = f(X, S)\right)\) and \(S\left(\frac{dS}{dt} = f(X, S)\right)\) are deeply connected and dependent. Moreover, due to the specificity of the experimental data, the error of the first part of Equation (9) is in the same range as the error of the second part of the equation. Therefore, the use of any weight coefficients in the objective function leads to the deterioration of the results.

The upper and lower bounds of the model parameters \(\mu_{\text{max}}, k_S\) and \(Y_{S/X}\) (see Equation (1)) are considered as follows [41]:

\[
x_{\text{min}} = [0, 0, 0], \tag{10}
\]

\[
x_{\text{max}} = [0.8, 1, 10]. \tag{11}
\]
For the problem considered here, the constructed solution is a 3-dimensional vector \( x_i = \left\{ \mu_{max}, k_S, \frac{Y_S}{X} \right\} \).

4.2. Numerical Results and Discussion

All computations have been performed using a PC/Intel Core i7-8700 CPU @ 3.20GHz, 16 GB Memory (RAM), Windows 10 operating system and Matlab R2013a environment.

The basic parameters of both algorithms (ABC and GA) have been set based on a series of preliminary tests in accordance with the problem under consideration. The chosen algorithms’ parameters and functions are summarized in Table 4 (only those parameters that obtain different values compared to those presented in Table 2 are listed here). The parameter settings of the other comparative algorithms are taken from the respective original papers.

Table 4. ABC-GA hybrid algorithm—parameters and functions.

| Algorithm | Parameter/Function | Value |
|-----------|-------------------|-------|
| ABC       | population size NP| 10    |
|           | number of iterations MCN | 25    |
|           | number of food sources SN | 5     |
| GA        | population size n  | 50    |
|           | number of generations MaxGen | 50    |

ABC algorithm starts with a small population of ten individuals. The initial solution is evaluated for only 25 iterations. The best 50 solutions are then used as an initial GA population. Again, for a low number of generations (50), GA looks for optimal model parameter estimates.

Because of the stochastic characteristics of the hybrid algorithm, a series of 30 runs are performed. The average, best and worst results of the 30 runs for the objective function value \( J \) are summarized in Table 5.

Table 5. Performance of ABC-GA hybrid algorithm.

| Algorithm         | Objective Function, \( J \) |
|-------------------|-----------------------------|
|                   | Average | Best  | Worst  |
| ABC results       | 5.0418  | 4.6221| 5.8566 |
| Final GA results  | 4.5106  | 4.3391| 4.7016 |

The presented results show that for a small population size (10) and only 25 iterations, the ABC algorithm produces a very good initial solution for GA. To form the initial population for GA, ABC with these particular parameters’ settings is executed 50 times. Next, based on this initial population, GA converges to a near-optimal solution for 50 generations. GA alone can reach such a result using a population of 100-200 chromosomes for at least 200 iterations. For example, solving the model with (6)–(8), GA’s best solution with \( J = 4.4359 \) is obtained for a population size of 200 and 200 generations [42] and \( J = 4.4396 \) with 180 individuals for 200 generations [35].

Estimated values of the model parameters \( (\mu_{max}, k_S \text{ and } \frac{Y_S}{X}) \), as well as statistical measures as the population variance \( (\sigma^2) \) and standard deviation (SD), are presented in Table 6. The population variance is used since it is a parameter of a set of decisions (estimations) that does not depend on research methods or sampling practices.
Table 6. Obtained model parameter estimations of ABC-GA hybrid algorithm.

| Model Parameters | \( \mu_{\text{max}} \) | \( \sigma^2 \) | SD  | \( k_S \) | \( \sigma^2 \) | SD  | \( Y_{S/X} \) | \( \sigma^2 \) | SD  |
|------------------|-----------------|--------|-----|--------|--------|-----|---------|--------|-----|
|                  | 0.4754          | 0.00010| 0.0103| 0.0999 | 0.000033| 0.0018| 2.0211  | 0.0000039| 0.0020|

Similar results for the estimates of the yield of glucose per biomass \((Y_{S/X})\) are reported in [41,43]. According to [44,45], the values of the parameters \( \mu_{\text{max}} \) and \( k_S \) are also within acceptable boundaries.

The values obtained for \( \sigma^2 \) and SD show the good performance of the proposed ABC-GA hybrid algorithm.

A graphical representation of the modeled E. coli fed-batch cultivation process variables (biomass and substrate) and the measured ones (real experimental data), based on the model parameters listed in Table 6, are presented in Figures 3 and 4.

![Figure 3](image-url)  
**Figure 3.** Time profiles of the process variable biomass: real experimental data and model predicted data.

![Figure 4](image-url)  
**Figure 4.** Time profiles of the process variable substrate: real experimental data and model predicted data.

The graphical results show that the ABC-GA hybrid algorithm achieves a very good correspondence between the measured and modeled process variables. The model obtained
by the ABC-GA hybrid scheme predicts with a high degree of accuracy the biomass and substrate dynamics of the process.

The results presented here are compared to already published results from the application of different hybrid metaheuristics to the same problem. The performance of the proposed hybrid ABC-GA is compared to the performance of hybrid algorithms as ACO-GA and GA-ACO [34] and ACO-FA [35] (Table 7).

Table 7. Comparison of the objective function values obtained by ABC with the published ones.

| Algorithm | J Average | J Best | J Worst | SD   |
|-----------|----------|--------|---------|------|
| ACO-GA    | 4.5765   | 4.4903 | 4.6865  | 0.06916 |
| GA-ACO    | 4.5706   | 4.3803 | 4.6949  | 0.06325 |
| ACO-FA    | 4.5196   | 4.4013 | 4.6803  | 0.05914 |
| ABC-GA    | 4.5106   | 4.3391 | 4.7016  | 0.05644 |

The comparison of the estimated model parameters ($\mu_{\text{max}}, k_S$, and $Y_{S/X}$) is presented in Table 8.

Table 8. Comparison of the model parameter estimates obtained by ABC with the published ones.

| Algorithm | $\mu_{\text{max}}$ | SD | $k_S$ | SD | $Y_{S/X}$ | SD |
|-----------|-------------------|----|-------|----|-----------|----|
| ACO-GA    | 0.4976            | 0.0120 | 0.0135 | 0.0023 | 2.0221 | 0.0025 |
| GA-ACO    | 0.4946            | 0.0121 | 0.0123 | 0.0020 | 2.0204 | 0.0024 |
| ACO-FA    | 0.4824            | 0.0110 | 0.0114 | 0.0019 | 2.0206 | 0.0021 |
| ABC-GA    | 0.4754            | 0.0103 | 0.0099 | 0.0018 | 2.0211 | 0.0020 |

The comparison shows that the designed ABC-GA hybrid algorithm performs better than the other considered algorithms. The best result, $J = 4.3391$, is obtained by ABC-GA. A very close result is achieved by the ACO-FA hybrid, considering the observed average results.

The worst result ($J = 4.7016$) obtained by ABC-GA shows that some further improvement of the GA evolution is necessary in order to achieve better results. It has been observed in the literature that the sampling capability of GA is greatly affected by the population size [46]. Another problem of GA is the algorithm’s parameters setting. The choice of appropriate control parameters is a tedious task. These will be directions for the process of improving the performance of GA in the ABC-GA hybrid algorithm proposed here.

In order to show the underlying frequency distribution of the results obtained by the compared hybrid algorithms, histograms are presented in Figure 5.

The observed distribution of the result of the ACO-GA hybrid algorithm is a plateau or multimodal distribution. GA-ACO and ACO-FA produce results which are with a left-skewed distribution. The objective function values of 30 runs obtained by the ABC-GA hybrid algorithm considered here are with a normal distribution. The advantage of hybridizing the ABC algorithm with GA over the other considered hybrid algorithms is that the multi-modality or skewed distribution is avoided.
Figure 5. Histograms of the results of objective function values of 30 runs.

5. Conclusions

In this paper, an ABC-GA hybrid algorithm is designed and applied to the parameter identification of a cultivation model. A system of nonlinear ordinary differential equations is used to model bacteria growth and substrate utilization. Model parameter identification is performed using a real experimental data set from an E. coli MC4110 fed-batch cultivation process.

The ABC-GA algorithm uses, in the beginning, a small ABC population size (10 individuals) and obtains the GA initial set of solutions for only 25 iterations. Next, for only 50 generations, GA converges to the final solution.

The proposed hybrid algorithm is further compared to other nature-inspired population-based hybrid metaheuristics known in the literature. As competing algorithms, GA, FA and ACO hybrids are chosen and applied to the same parameter identification problem. It is shown that the ABC-GA hybrid algorithm outperforms the other competitor algorithms.

Since the hybrid uses a low number of individuals, the known dependence of sampling capability of GAs on the population size could be exhibited. For example, in the case when ABC-GA obtains the worst results (Table 7). Further improvements to the designed ABC-GA hybrid will be directed to overcoming this dependence. Another direction for improving the algorithm is the algorithm’s parameters setting. A more in-depth study on the influence of the control parameters (both for ABC and GA) on the hybrid algorithm’s performance could be made.

### Table 7. Comparison of the objective function values obtained by ABC with the published ones.

| Model          | Parameters | Objective Function | SD Max | Median |
|----------------|------------|--------------------|--------|--------|
| GA-ACO         |            | 4.5706             | 0.0103 |        |
| ACO-GA         |            | 4.4013             | 0.0023 |        |
| ACO-FA         |            | 4.4949             | 0.0120 |        |
| ABC-GA         |            | 4.4949             | 0.0135 |        |
| GA             |            | 3.8169             | 0.0021 |        |
| FA             |            | 3.8169             | 0.0023 |        |

(c) ACO-FA histogram

(d) ABC-GA histogram
The good results observed here for the ACO-FA hybrid algorithm and the results in [47] could be the basis for possible future work where hybridization between ACO and ABC, ABC and FA, or ABC and CS algorithm could be investigated.

Author Contributions: Conceptualization and methodology, O.R. and V.L.; software implementation, O.R. and D.Z.; validation, O.R. and V.L.; investigation, O.R., D.Z. and V.L.; writing—original draft preparation, O.R., D.Z. and V.L.; writing—review and editing, O.R., D.Z. and V.L.; visualization, O.R. and D.Z.; supervision, O.R. and V.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Scientific Fund of Bulgaria, Grant K11-06-H32/3 “Interactive System for Education in Modelling and Control of Bioprocesses (InSEMCoBio)”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Sörensen, K.; Sevaux, M.; Glover, F. A history of metaheuristics. In Handbook of Heuristics; Martí, R., Pardalos, P., Resende, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2017.
2. Mendes, J.M.; Oliveira, P.M.; dos Santos, F.N.; Morais dos Santos, R. Nature inspired metaheuristics and their applications in agriculture: A short review. In Progress in Artificial Intelligence; Moura, O.P., Novais, P., Reis, L., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; Volume 11804, pp. 167–179.
3. Santamaria, J.; Rivero-Cejudo, M.L.; Martos-Fernández, M.A.; Roca, F. An overview on the latest nature-inspired and metaheuristics-based image registration algorithms. Appl. Sci. 2020, 10, 1928. [CrossRef]
4. Bennis, F.; Bhattacharjya, R.K. (Eds.) Nature-inspired methods for metaheuristics optimization. Algorithms and applications in science and engineering. In Modeling and Optimization in Science and Technologies; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; Volume 16. [CrossRef]
5. Ezugwu, A.E. Nature-inspired metaheuristic techniques for automatic clustering: A survey and performance study. SN Appl. Sci. 2020, 2, 273. [CrossRef]
6. Kaur, A.; Kumar, Y. A new metaheuristic algorithm based on water wave optimization for data clustering. Evol. Intell. 2021, 1–25. [CrossRef]
7. Kaveh, A. Advances in Metaheuristic Algorithms for Optimal Design of Structures; Springer International Publishing: Berlin/Heidelberg, Germany, 2021. [CrossRef]
8. Talbi, E.G.; El-Ghazali (Eds.) Hybrid Metaheuristics, Studies in Computational Intelligence; Springer International Publishing: Berlin/Heidelberg, Germany, 2013; Volume 434, XXVI, 458p.
9. Ting, T.O.; Yang, X.S.; Cheng, S.; Huang, K. Hybrid metaheuristic algorithms: Past, present, and future. In Recent Advances in Swarm Intelligence and Evolutionary Computation. Studies in Computational Intelligence; Yang, X.S., Ed.; Springer International Publishing: Cham, Switzerland, 2015; Volume 585, pp. 71–83.
10. Oduntan, O.I.; Thulasiraman, P. Hybrid metaheuristic algorithm for clustering. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 1–9.
11. Rosle, M.S.; Mohamad, M.S.; Choon, Y.W.; Ibrahim, Z.; González-Briones, A.; Chamoso, P.; Corchado, J.M. A hybrid of particle swarm optimization and harmony search to estimate kinetic parameters in Arabidopsis thaliana. Processes 2020, 8, 921. [CrossRef]
12. Cao, J.; Zhu, S.; Li, C.; Han, B. Integrating support vector regression with genetic algorithm for hydrate formation condition prediction. Processes 2020, 8, 519. [CrossRef]
13. Luo, Y.; Pan, Y.; Li, C.; Tang, H. A hybrid algorithm combining genetic algorithm and variable neighborhood search for process sequencing optimization of large-size problem. Int. J. Comput. Integr. Manuf. 2020, 33, 10–11. [CrossRef]
14. Nalluri, M.R.; Kannan, K.; Gao, X.Z.; Roy, D.S. Multiobjective hybrid monarch butterfly optimization for imbalanced disease classification problem. Int. J. Mach. Learn. Cyber. 2020, 11, 1423–1451. [CrossRef]
15. Liu, M.; Yao, X.; Li, Y. Hybrid whale optimization algorithm enhanced with Lévy flight and differential evolution for job shop scheduling problems. Appl. Soft Comput. 2020, 87, 105954. [CrossRef]
16. Khelifi, A.; Bentouati, B.; Chettih, S.; Sehiemy, R.A.E. Application of hybrid salp swarm optimization method for solving opf problem. In Proceedings of the 2019 International Conference on Advanced Electrical Engineering, Bandung, Indonesia, 18–20 September 2019; pp. 1–6. [CrossRef]
17. Dhiman, G. ESA: A hybrid bio-inspired metaheuristic optimization approach for engineering problems. Eng. Comput. 2021, 37, 323–353. [CrossRef]
18. Karaboga, D. An Idea Based on Honeybee Swarm for Numerical Optimization; Technical Report TR06; Erciyes University, Engineering Faculty, Computer Engineering Department: Kayseri, Turkey, 2005.
19. Gu, W.; Yu, Y.; Hu, W. Artificial bee colony algorithm-based parameter estimation of fractional-order chaotic system with time delay. IEEE/CAA J. Autom. Sin. 2017, 4, 107–113. [CrossRef]

20. Panniem, A.; Puphasuk, P. A modified artificial bee colony algorithm with firefly algorithm strategy for continuous optimization problems. J. Appl. Math. 2018, 1237823. [CrossRef]

21. Rabiee, A.; Jamadi, M.; Mohammadi-Ivatloo, B.; Ahmadian, A. Optimal non-convex combined heat and power economic dispatch via improved artificial bee colony algorithm. Processes 2020, 8, 1036. [CrossRef]

22. Gao, H.; Fu, Z.; Pun, C.M.; Zhang, J.; Kwong, S. An efficient artificial bee colony algorithm with an improved linkage identification method. IEEE Trans. Cybern. 2020. [CrossRef]

23. Banharnsakun, A. Artificial bee colony algorithm for content-based image retrieval. Comput. Intell. 2020, 36, 351–367. [CrossRef]

24. Duan, H.-B.; Xu, C.-F.; Xing, Z.-H. A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems. Int. J. Neural Syst. 2010, 20, 39–50. [CrossRef]

25. Ghanem, W.A.H.S.M.; Jantaw, A. Hybridizing artificial bee colony with monarch butterfly optimization for numerical optimization problems. Neural Comput. Appl. 2016. [CrossRef]

26. Lian, L.; Fu, Z.; Yang, G.; Huang, Y. Hybrid artificial bee colony algorithm with differential evolution and free search for numerical function optimization. Int. J. Artif. Intell. Tools. 2016, 25, 1650020. [CrossRef]

27. Nithya, B.; Jeyachidra, J. Hybrid ABC-BAT optimization algorithm for localization in HWSN. Microprocess. Microsyst. 2021. [CrossRef]

28. Sun, L.; Chen, T.; Zhang, Q. An artificial bee colony algorithm with random location updating. Hindawi Sci. Program. 2018, 2018, 2767546. [CrossRef]

29. Yan, X.; Zhu, Y.; Zou, W. A hybrid artificial bee colony algorithm for numerical function optimization. In Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems (HIS), Malacca, Malaysia, 5–8 December 2011; pp. 127–132. [CrossRef]

30. Lin, Y.; Wang, J.; Li, X.; Zhang, Y.; Huang, S. An improved artificial bee colony for feature selection in QSAR. Algorithms 2021, 14, 120. [CrossRef]

31. Karaboga, D.; Akay, B. A comparative study of artificial bee colony algorithm. Appl. Math. Comput. 2009, 214, 108–132. [CrossRef]

32. Goldberg, D.E. Genetic Algorithms in Search, Optimization and Machine Learning; Addison Wesley Longman: London, UK, 2006.

33. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. 2010, 23, 689–694.

34. Tuttle, A.R.; Trahan, N.D.; Son, M.S. Growth and maintenance of Escherichia coli laboratory strains. Curr. Protoc. 2021, 1, e20. [CrossRef]

35. Elbling, K.L.; Brent, R. Recipes and tools for culture of Escherichia coli. Curr. Protoc. Mol. Biol. 2019, 125, e83. [CrossRef]

36. Cardoso, V.M.; Campani, G.; Santos, M.P.; Silva, G.G.; Pires, M.C.; Gonçalves, V.M.; de Giordano, R.C.; Sargo, C.R.; Horta, A.C.L.; Zangirolami, T.C. Cost analysis based on bioreactor cultivation conditions: Production of a soluble recombinant protein using Escherichia coli BL21(DE3). Biotechnol. Rep. 2020, 26, e00441. [CrossRef]

37. Kögerl, M.; Itkonen, J.; Viitala, T.; Casteleijn, M.G. Assessment of recombinant protein production in E. coli with Time-Gated Surface Enhanced Raman Spectroscopy (TG-SERS). Sci. Rep. 2020, 10, 2472. [CrossRef]

38. Roeya, O.; Pencheva, T.; Hitzmann, B.; Tzonkov, S. A genetic algorithms based approach for identification of Escherichia coli Fed-batch Fermentation. Int. J. Bioautom. 2004, 1, 30–41.

39. Levisauskas, D.; Galvanauskas, V.; Henrich, S.; Wilhelm, K.; Volk, N.; Lubbert, A. Model-based optimization of viral capsid protein production in fed-batch culture of recombinant Escherichia coli. Bioprocess Biosyst. Eng. 2003, 25, 255–262. [CrossRef]

40. Roeya, O.; Fidanova, S.; Paprzycki, M. Influence of the population size on the genetic algorithm performance in case of cultivation process modelling. In Proceedings of the IEEE Federated Conference on Computer Science and Information Systems (FedCSIS), WCO 2013, Kraków, Poland, 8–11 September 2013; pp. 371–376. Hindawi Sci. Program. 2018, 125, e83. [CrossRef]

41. Roeya, O.; Fidanova, S. Parameter Identification of an E. coli cultivation process model using hybrid metaheuristics. Int. J. Metaheuristics 2014, 3, 133–148. [CrossRef]

42. Contiero, J.; Beatty, C.; Kumari, S.; DeSanti, C.L.; Strohl, W.L.; Wolfe, A. Effects of mutations in acetate metabolism on high-cell-density growth of Escherichia coli. J. Ind. Microbiol. Biotechnol. 2000, 24, 421–430. [CrossRef]

43. Zelić, B.; Vasic-Racki, D.; Wandrey, C.; Takors, R. Modeling of the pyruvate production with Escherichia coli in a fed-batch bioreactor. Bioprocess Biosyst. Eng. 2004, 26, 249–258. [CrossRef]

44. Poccia, M.E.; Beccaria, A.J.; Dondo, R.G. Modeling the microbial growth of two Escherichia coli strains in a multi-substrate environment. Braz. J. Chem. Eng. 2014, 31. [CrossRef]

45. Roeya, O.; Fidanova, S.; Paprzycki, M. InterCriteria analysis of ACO and GA hybrid algorithms. Stud. Comput. Intell. 2016, 610, 107–126.

46. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. Multimed. Tools Appl. 2021, 80, 8091–8126. [CrossRef]

47. Roeya, O.; Atanassova, V. Cuckoo search algorithm for model parameter identification. Int. J. Bioautom. 2016, 20, 483–492.