IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 April 2024, accepted 8 June 2024, date of publication 3 July 2024, date of current version 19 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3422450

RESEARCH ARTICLE

Kinetic Parameters Estimation of the Escherichia Coli (E. coli) Model by Garra Rufa-Inspired Optimization Algorithm (GRO)

JASNI MOHAMAD ZAIN[1], (Senior Member, IEEE), MOHAMMED ADAM KUNNA AZRAG[1,2], SAIFUL FARIK MAT YATIN[1], GHADAH ALDEHIM[3], ZUHAIRA MUHAMMAD ZAIN[3], HADIL SHAIBA[3], NAZIK ALTURKI[3], SAPIAH SAKRI[3], (Member, IEEE), AZLINAH MOHAMED[4], AND AQEEL S. JABER[5]

[1]Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Komplek Al-Khawarizmi, Universiti Teknologi MARA (UiTM), Shah Alam 40450, Malaysia
[2]Faculty of Computing, Universiti Malaysia Pahang, Pekan 26600, Malaysia
[3]College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11564, Saudi Arabia
[4]YSL Strategic Alliance Sdn Bhd, Shah Alam 40160, Malaysia
[5]Independent Researcher, 02600 Helsinki, Finland

Corresponding author: Mohammed Adam Kunna Azrag (kunna87mohammed@gmail.com)

This work was supported in part by the Deanship of Scientific Research at Princess Nourah Bint Abdulrahman University, through the Research Groups Program, under Grant RGP-1443-0045; and in part by the Institute for Big Data Analytics and Artificial Intelligence (IBDAAI).

ABSTRACT Due to complex nature of metabolic pathways, *E. coli* metabolic model kinetic parameters are difficult to detect experimentally. Thus, obtaining accurate kinetic data for all reactions in an E. coli metabolic model is a technically-challenging process. So, Garra Rufa-inspired Optimization (GRO) Algorithm is applied to the primary metabolic network of *E. coli* as a model to estimate small-scale kinetic parameters and increase the kinetic accuracy. Also, the Differential Algebraic Equations (DAE) is used to represent the glycolysis, phosphotransferase system, pentose phosphate, the TCA cycle, gluconeogenesis, glyoxylate, and acetate production pathways of *Escherichia coli* in the metabolic network. Based on the behavior of the Garra Rufa fish, a route is modelled in which particles are sorted into groups and each group is guided by the best value. In addition, the fitness of the group leaders determines whether or not these particles are able to switch groups. In this study, experimental data was used to estimate seven kinetic parameters. However, the numerical results of The Relative Error (RE) and the Mean Error (ME) reveal that the observed and anticipated data are in line with the results. As a result of this new method, it was discovered that small-scale and even whole-cell dynamic models can be estimated accurately.

INDEX TERMS Coli E, estimation, GRO, modeling, PSO.

I. INTRODUCTION

In systems biology and bioinformatics, computing simulations and optimization play an essential role in the application of mathematical approaches to the reverse engineering of biological systems and managing uncertainty in this setting. Numerous academics have proposed various parallelization strategies in an effort to accelerate simulation, calibration, and analysis of models of realistic sizes [1] due to the massive amount of computing necessary for these tasks. Recent research has centered on the construction of larger-scale dynamic (kinetic) models, with the ultimate goal of generating whole-cell models. Calibration of models has garnered a great deal of interest, particularly in relation to global optimization metaheuristics and hybrid techniques [2].

Thus, kinetic models are being constructed in order to quantitatively characterize the kinetics of biological processes. These models consider the stoichiometry of reactions

The associate editor coordinating the review of this manuscript and approving it for publication was Min Wang.

and the kinetic expressions of each enzyme. Enzyme kinetic characteristics are determined by exposing isolated enzymes to ideal conditions in in vitro research. These circumstances are not identical to those surrounding enzymes in living cells; hence, the use of in vitro parameters in kinetic models can lead to inaccurate predictions of in-tricellular metabolite concentrations [3] when compared to in vivo-measured concentrations. By perturbing a culture and quantifying fluxes, enzyme levels, and intra- and extra-cellular metabolite concentrations as a function of time, the dynamics of cell metabolism can be fine-tuned. Advancements in experimental techniques paved the way for the development of dynamic models for metabolic networks that can assess microbial behavior [4].

Several Escherichia coli (E. coli) models were constructed and simulated in order to better understand the model's' behavior's to enable the manufacture of specific products like those reported in [5], [6], [7], [8], and [9]. The authors of [5] simulate and estimate the kinetic parameters of three pathways while disregarding the others (gluconeogenesis and glyoxylate). The (Pts) system is used in [6] to add glutamine/aspartate metabolism and fructose ingestion into the model. The model, however, does not consider the full route model or E. coli cell development. The researchers in [7] produce experimental time sequences of extracellular glucose and biomass in the E. coli model but do not include overall pathway production. According to [8] Kotte utilized the Monod equations to model glucose uptake without predicting the particular growth rate in E. coli based on a molecular mechanism; he did not consider estimating the full E. coli major metabolic route. Neglecting the complete major metabolic route may result in an erroneous simulation outcome prediction. As a result, large-scale kinetic parameters must be thoroughly explored. A complete model can only be studied thoroughly if the entire cell system is included. This is due to the feedback loops of various metabolites, including PEP, OAA, PYR, and others, which may affect other enzymes in the primary metabolic pathway, leading the concentration of other metabolites to fluctuate dynamically over time [10], [11].

Recently, the modified simplex technique [12], [13], simulated annealing, and differential evolution were tested for predicting kinetic parameters of a dynamic model of central carbon metabolism in *Escherichia coli*. The authors claim that differential evolution generated the best results. Furthermore, in [14], the researchers improved the central carbon metabolism model and derived kinetic parameters for the glycolytic enzymes from [5]. MATLAB and a weighted least squares objective function were used to tackle the parameter estimation problem.

As a result of the challenges in computing kinetic parameters, researchers are increasingly employing metaheuristic optimization [15] algorithm approaches to estimate the kinetic parameters of the *E. coli* model and other biological models. Several of these metaheuristic approaches were employed to estimate the kinetic parameters used in [16], [17], [18], and [19], and some of these algorithms relied on experimental data from [20, 5]. Furthermore, because of the enormous search space that must be studied, biological kinetic models contain hundreds or thousands of parameters, making parameters estimation problematic. High-dimensional kinetic models (with hundreds of even thousands of different kinetic parameters) are difficult to compute. As a result, the performance of the foregoing techniques suffer, resulting in low accuracy [21].

Differential Evolution (DE) is one method that is frequently used and researched for parameter estimation in metabolic models. The main weakness of the DE technique is its time consumption, which makes it difficult for the DE algorithm to determine its parameters when there are a large number of processors and several local searches involved. Same issue is observed when we use the evolutionary algorithms, such as the Genetic Algorithm (GA). This method is frequently used to calculate metabolic model parameters. The time it takes to compute is the algorithm's main drawback when compared to Particle Swarm Optimization (PSO), and other approaches [22], [23].

Since its introduction in 1995, the PSO algorithm has been employed in many disciplines, including those mentioned in [24], [25], [26], and [27]. The method described in [28] was used to represent birds and fish obtaining food. They exchange information as they progress through the search process, and then hunt for their target at random and autonomously. As a result, the search space is filled with many courses and orientations for the particles to take.

Furthermore, based on the Segment Particle Swarm Optimization (Se-PSO) algorithm, the Enhanced Segment Particle Swarm Optimization (ESe-PSO) method was conceptualized and developed with the goal of performing deep searches while keeping the accuracy of Se-PSO. This evolution is based on a knowledge of Se-PSO's local and global point initialization [2]. In this instance, segmentation divides the particles into groups, allowing them to collaborate toward the best solution. This method was developed to find large-scale kinetic parameters in an E. coli model [9] as well as governor-turbine models in a single area power plant [28], [29], [30].

The inertia weight has an effect on particle exploration and exploitation due to the linearity of the model's linear inertia weight (ω) [28]. As a result, substantial exploration is required at the start of the algorithm's execution and limited exploitation at the end. This was done to avoid the local optima trap and thus improve the efficiency of calculating kinetic parameters in order to minimize model distances in an acceptable length of time.

However, the Garra Rufa-inspired optimization algorithm is a type of optimization algorithm that is inspired by the behavior of the Garra Rufa fish, also known as "doctor fish." These fish are known for their ability to clean and exfoliate the skin of other fish in a symbiotic relationship [31].

The Garra Rufa-inspired optimization algorithm is a population-based optimization algorithm that uses the behavior of the Garra Rufa fish as a metaphor for solving optimization problems. The algorithm works by creating a population of candidate solutions, and then iteratively improving the solutions by simulating the behavior of the Garra Rufa fish. The algorithm divides the candidate solutions into two groups: active solutions and passive solutions. The active solutions are the solutions that are being actively improved upon, while the passive solutions are the solutions that are being used to improve the active solutions. The algorithm iteratively updates the active solutions by selecting the best solutions from the passive solutions, and then applying various optimization techniques. After each iteration, the active solutions are evaluated, and the best ones are selected to become the passive solutions for the next iteration. The process continues until a stopping criterion is met, such as reaching a certain level of accuracy or a certain number of iterations. The Garra Rufa-inspired optimization algorithm has been applied to various optimization problems, such as function optimization, and has shown to be effective in finding near-optimal solutions [32], [33], [34], [35], [36].

In this study, the model under consideration [9] was designed to represent *E. coli's* primary metabolic pathway. In addition to acetate synthesis, this model comprises six pathways: glycolysis, pentose phosphate, the TCA cycle, gluconeogenesis, and glyoxylate. This model was preferred to the other models in this study because of its capacity to simulate the key metabolic pathways of *E. coli* with small-scale kinetic parameters. Furthermore, due to a lack of genuine experimental data, the model's nonlinearity, and the small-scale kinetic parameters, the model's response to metabolites must be studied based on its capacity for estimating large-scale kinetic parameters [37]. The experimental dataset used in this work [20] has a large number of metabolites and has been used in other investigations for kinetic parameters, including [21, 17, 16, 38, 39]. The goal of this study is to implement the Garra Rufa algorithm to calculate small-scale kinetic parameters estimation. The sensitive kinetics were acquired to aid in estimation [4].

The rest of the paper is structured as follows. Section I contains the introduction. Section II outlines the problem statement. Section III describes the strategy. The consequence is discussed in Section IV. The conclusion is discussed in the final section. As a result of the good outcome of this study, we expect other researchers apply this technique in the estimation of large-scale dynamic models in systems biology.

II. PROBLEM FORMULATION

The enzymatic equation for Ordinary Differential Equation (ODE) functions includes kinetic metabolic models. Erroneous metabolic kinetic models can impair model behavior and process design. As a result of the model's nonlinearity, getting high data features is a difficult task because kinetics are collected from numerous laboratories and under different conditions [40], [41], [42].

In this sense, the basic E. coli kinetic metabolic model generated in [9] has a significant number of kinetic parameters divided across six pathways. In terms of understanding and replicating E. coli behavior, this model has a significant impact on model simulations. However, the researches indicated that the model needed additional research focused on its kinetic parameter estimations and responses. Similarly, additional comparisons with genuine experimental data with large-scale kinetic parameters are needed. As a result, additional research in this area is critical, hence this study. The technique of parameter estimation, on the other hand, is searching for the parameter values in a mathematical model (formulated using ODE) that best fit the experimental data. This is accomplished by minimizing the scalar distance between the model prediction and the experimental data while accounting for experimental mistakes. This problem is classified into three types: multimodal optimization, continuous optimization, and single-objective optimization. The following is a description of the objective function of kinetic parameter estimate considered in this work:

$$f = \sum_{i=1}^{nm} \sum_{j=1}^{nt} \left| \left(y^{r,1} - y^{s,1}\right) + \left(y^{r,2} - y^{s,2}\right) + \ldots + (y^{r,m} - y^{s,m}) \right| \tag{1}$$

where f is the objective function, nm is the summation over measured state of i variables at of j^{th} measurement, nt is the data points for each measured variables, $y^{r,m}$ is the actual model output r resulting from m metabolites. The $y^{s,m}$ term is the simulation model output s for m metabolites.

Since biological problems are nonlinear, estimating the appropriate parameters for this problem is difficult due to the existence of numerous local minima. As a result, most optimization methods become easily caught in local minima, resulting in a poor convergence time. Again, the resulting parameter estimation yields a poor fit for the experimental data and a low model prediction accuracy.

In conclusion, the problem of small-scale kinetic parameters estimation refers to the challenge of accurately determining the values of the kinetic parameters (e.g. rate constants) in a system with a large number of chemical reactions. This can be a complex and computationally intensive task, as it requires solving a large set of non-linear differential equations that describe the system. The accuracy of the estimates depends on the quality of the data, the choice of models, and the optimization algorithms used. In recent years, advances in computational methods and increasing availability of high-quality data have improved the accuracy of large-scale kinetic parameter estimation.

III. METHODOLOGY

A. THE STRUCTURE OF THE E. COLI KINETIC MODEL

The dynamic model of the main metabolic pathway of E. coli formulated in [9] was used as a benchmark. This model, which consists of glycolysis, pentose phosphate, the TCA cycle, gluconeogenesis, and glyoxylate pathways in addition to acetate formation with the phosphotransferase system, is used as described. This model has 23 metabolites, 28 enzymatic reactions, 10 co-factors (e.g., adenosine triphosphate (ATP), coenzyme A (COA), nicotinamide adenine dinucleotide phosphate (NADPH)) and 172 kinetic parameters. Equation 2 gives the rate at which the concentration of the metabolite in the considered model changes.

$$\frac{dC_i}{dt} = \sum\nolimits_j R_{ij} v_j - \mu C_i \tag{2}$$

where C_i is the concentration of metabolite i, R_{ij} is the stoichiometric coefficient of metabolite i in the reaction j, v_j is the rate of the reaction j, and μC_i is the growth rate on the dilution effect. The detailed methodology can be deduced from the metabolic model in Figure 1, while the model mass balance equations in Appendix Table 4, and the kinetic rate equations in Appendix Table 5 for the investigated model were listed in the Appendix.

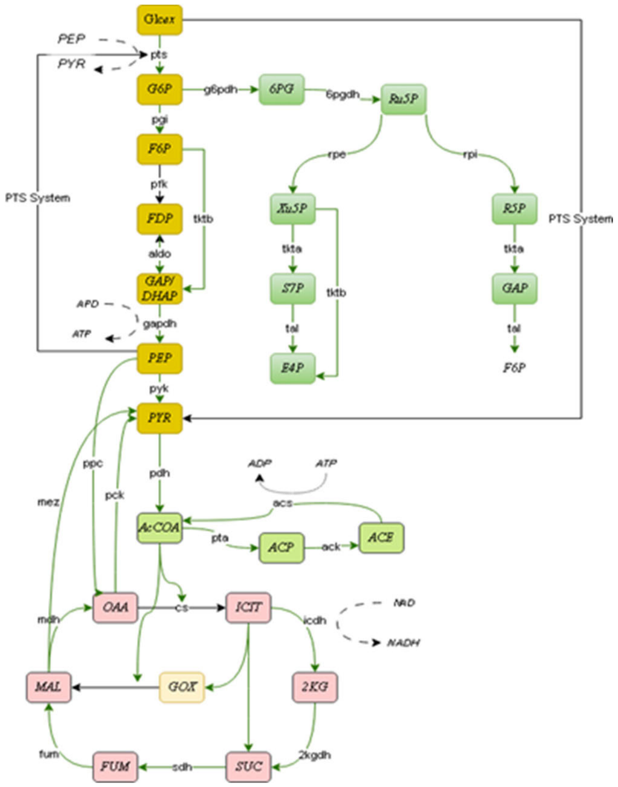

FIGURE 1. The main metabolic model of E. coli.

B. PSO ALGORITHM

Particle Swarm Optimization (PSO) is a computational optimization algorithm inspired by the social behavior of birds flocking or fish schooling. It is a metaheuristic optimization algorithm that uses a swarm of particles to search for the optimal solution of a problem. Each particle represents a potential solution to the problem, and the swarm moves through the solution space guided by the best solution found so far. The velocity and position of each particle are updated based on its own previous best solution, the best solution found by any particle in the swarm, and a random factor. The algorithm is simple to implement, easy to parallelize, and has been applied to a wide range of optimization problems in various fields including engineering, finance, and computer science [28].

For n-dimensional search space, the velocity and position of the i^{th} particle represents as: $V_i = (V_{i1}, V_{i2}, \ldots, V_{id})^t$ and $X_i = (X_{i1}, X_{i2}, \ldots, X_{id})^t$ respectively. Therefore, Particle behavior at each iteration is represented by Eq 3 & 4:

$$\begin{aligned} v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (p_{id}(t) - X_{id}(t)) \\ + c_2 r_2 (G_{id}(t) - X_{id}(t)) \end{aligned} \tag{3}$$

$$X_{id}(t+1) = X_{id}(t) + v_{id}(t+1) \tag{4}$$

where c_1 and c_2 are acceleration coefficients, ω is the inertia weight parameter, V_{id} is the velocity of the i^{th} particle at the dimension n, X_{id} is the position of the i^{th} particle at the dimension n, $d = 1, 2, \ldots, n$ represent the dimension and $i = 1, 2, \ldots, N$ represent the particle index. N is the size of the swarm, $p_{id}(t)$ is the best position of the i^{th} particle at the dimension n during the iteration (t), and $G_{id}(t)$ is the best global best position of the i^{th} particle at the dimension n during the iteration (t).

C. GRO ALGORITHM

To describe this algorithm, considering this scenario the Garra Rufa fish found in a professional foot massage aquarium [30], it is clear that the fish are organized into elaborate communities, complete with hierarchies and social strata. Further, a certain number of fishes are constantly switching places. The aquarium was taken before it was submerged. The fish are not uniformly distributed and form a single large school, as is readily apparent. As soon as the aquarium is sunk two feet into the pool, the fish are quickly separated into two groups (one for each foot). This is where we see the intriguing position transfers between the two groups, when we see a specific number of fishes go from one foot to the other at the same moment. Thus, the fish movements depict the random motions of small schools of fish between the two feet, which is consistent with the idea proposed by Azrag et al. [28] that the solution to a complicated function can be achieved by use of a nonconstant particle swarm.

In the scenario that food is located at many locations, the fish will break up into smaller groups to more evenly distribute themselves among the areas [30], [31]. Then, smaller schools of fish will swim back and forth among the larger schools to bolster the school that is most likely to have the sweet spot. When there is a considerable disparity in the availability of food, as depicted in Figure 5, the fish may select the best group.

In this experiment, we manipulated the amount of food consumed by submerging one foot in water before the other by one minute. In order to discover the global optimum for several problems using a simulation of fish herding behavior, the idealized assumptions can be reduced as follows. First, the fish will be split up into smaller groups (one for each fin), each with its own leader. Second, each fish can take on the role of leader or follower based on how close they are to the optimal position. Third, we will choose the number of leaders based on the problem's difficulty and the predicted number of optimal points for the objective functions. Fourth, at the end of each cycle, some of the fish will move to the group with the highest ideal value, based on the results of the previous cycles.

The number of fishes is held constant, but the randomness of the fish that swim around is thought to be a factor. Fifth, in each iteration, both the sub-global fitness of each leader and the global fitness will be calculated. After a certain number of cycles, all the fish will have to follow the same leader. Therefore, we can generate some wild assumptions about the number of followers using Eq (5):

$$followers\ number = \frac{n - leaders\ number}{leaders\ number} \tag{5}$$

In this equation, n represents the total number of particles. Assuming that Group 1 is the best, the next step is to determine the objective function for each fish and sort the values to determine the leaders and the best group.

Equations (6) and (7) can be used to compute the change in the number of mobile and follower fish for the poorest leaders:

$$mob_i = integer(ą * random) \tag{6}$$

$$fol_{ij} = Max(\left(fol_{ij-1} - mob_i\right), o) \tag{7}$$

where ą is the maximum number of mobile fish for one group, random is from (0-1), $f = i$ is the iteration number, and mob_i, fol_{ij} are the number of followers and mobile fish for the ith leader. The changes in mobile and follower counts for the best leader are indicated below in Eq 8 & 9:

$$mob_1 = \sum\nolimits_2^n mob_i \tag{8}$$

$$fol_{1,j} = fol_{1,j-1} + mob_1 \tag{9}$$

The GR algorithm proceeds according to the process illustrated by Algorithm 1 below.

The first step is to define the objective function that needs to be optimized. In the case of kinetic parameter estimation, the objective function could be the sum of the squared differences between the experimental data and the model predictions, where the model predictions are calculated using a set of kinetic parameters. The goal is to find the set of kinetic parameters that minimize the objective function.

Once the objective function is defined, we can apply the Garra Rufa algorithm to explore the parameter space and find the best set of kinetic parameters. Here is an implementation description of the algorithm:

1) Define the number of fish and the number of leaders based on the problem's difficulty and the predicted number of optimal points for the objective function.
2) Initialize the fish randomly in the parameter space.

Algorithm 1 The Garra Rufa Adoption

```
1. Select the initial values (number of leaders, number of particles initialization of the basic method, fitness function limits, percentage value of the maximum moving particles ($), $c_1$, $c_2$, $\omega$.
2. Initialize the kinetic boundaries
3. Set the Number of $folllowers = n/num$ ber of leaders
4. Find fitness for $n$ of particles, sort according to fitness
5. Set the first "number of leaders" optimal particles as leaders.
6. While t < iteration do
    7. For $i = 1$ to number of leaders
        8. Update particles for the followers of leaders ($i$) Using basic optimization Algorithm
        9. Sort leaders according to fitness
    10. End For
    11. For $i = 2$ to number of leaders
        12. Redistribute the followers according to the following equations
        13. Mobile fish .from worse groups ($i$) = $\$ * random$ Followers ($i$) = Max (0, followers ($i$)- mobile fish ($i$))
        14. Total mobile fish of the best leader = total mobile fishes + mobile fish ($i$)
    15. end for
    16. Followers (1) = Followers (1) + total mobile fish
    17. Finding the sub-global solution for each leader
    18. Determine the global solution
19. End While
```

3) Evaluate the fitness of each fish using the objective function.
4) Split the fish into smaller groups (one for each fin), each with its own leader.
5) Each fish can take on the role of leader or follower based on how close they are to the optimal position. The leader fish are those with the best fitness in each group.
6) Allow the follower fish to move randomly around the parameter space, guided by the leader fish.
7) At the end of each cycle, some of the fish will move to the group with the highest ideal value, based on the results of the previous cycles. This ensures that the search is focused on the best regions of the parameter space.
8) In each iteration, calculate both the sub-global fitness of each leader and the global fitness. The sub-global fitness is the fitness of each leader in its own group, while the global fitness is the best fitness among all the leaders.
9) Repeat steps 3-8 until convergence is achieved or a maximum number of iterations is reached.
10) Once convergence is achieved, the set of kinetic parameters that correspond to the leader fish with the best fitness can be used as the best estimate for the kinetic parameters that fit the experimental data.

The Garra Rufa algorithm is a bio-inspired optimization algorithm utilized for the purpose of parameter optimization. Within this particular framework, it is utilized to determine the optimal set of kinetic parameters for a specified objective function.

Procedure of the Garra Rufa Algorithm:

1. Specify the parameters:

Quantity of fish: Estimate the population size, considering the capacity for exploration.

Determine the quantity of leaders by considering the intricacy of the problem and the anticipated optimal outcomes.

2. Initialization:

Initialize fish in the parameter space in a random manner.

3. Assess Physical Condition:

Utilize the specified objective function to assess the fitness of every fish.

4. Formation of Groups:

Divide the fish into smaller clusters, with each cluster having one fin (leader). Every group possesses its own leader.

5. Process of Choosing a Leader:

Determine the dominant fish by assessing their proximity to the most favorable locations. Leaders are individuals who possess superior physical fitness within their respective groups.

6. Follower's Mobility:

Enable follower fish to navigate randomly within the defined boundaries, under the guidance of their respective leaders.

7. Conduct a targeted search:

At the conclusion of each cycle, certain fish migrate towards the cluster exhibiting the greatest ideal value, thereby prioritizing exploration in the vicinity of the most optimal regions.

8. Calculation of Physical Fitness:

During each iteration, compute the sub-global fitness for each leader within its group as well as the global fitness, which represents the highest fitness value among all leaders.

9. Sequential Procedure:

Continue executing steps 3-8 repeatedly until either convergence is achieved or the maximum number of iterations, as predetermined, is reached.

10. Convergence refers to the process of coming together or meeting at a common point.

Upon achieving convergence, the collection of kinetic parameters associated with the leader fish exhibiting the highest level of fitness is regarded as the most accurate estimation for fitting the experimental data.

Clarification:

The Garra Rufa algorithm is derived from the behavior of the Garra Rufa fish, which is renowned for its symbiotic cleaning behavior during spa treatments. In this optimization context:

Group Dynamics: Fish exhibit social behavior by forming smaller units known as groups, with each group having a designated leader. This replicates the communal investigation behavior observed in schools of fish.

The algorithm utilizes a leader-follower mechanism, in which leaders provide guidance to followers within the parameter space. The selection of leaders is based on their aptitude, guaranteeing that the most talented individuals take charge of the exploration.

Dynamic Movement: Followers exhibit stochastic motion, emulating the capriciousness of fish locomotion. The stochastic nature facilitates extensive exploration of the parameter space.

The algorithm strategically prioritizes the most favorable regions of the parameter space, thereby enhancing the likelihood of discovering optimal solutions.

Fitness Assessment: The evaluation of fitness occurs at both local (sub-global) and global levels, allowing for a thorough comprehension of the solution landscape.

Convergence Criteria: The algorithm continues iterating until it reaches a state of convergence, which occurs when it either finds a satisfactory solution or reaches a predetermined maximum iteration limit.

The quantitative evaluation of the model's estimation performance was done using two diagnostic measures that rely on the average values of the primary state variables. The first measure, known as the mean error (ME) (as shown in Eq10), indicates the model's bias and provides insight into whether the model tends to overestimate or underestimate a variable. The second measure, called the relative error (RE) (as shown in Eq 11), assesses the accuracy of the model [27], [43].

$$ME = \frac{\sum_{i=1}^{NT} (C_i - C\wedge_i)}{NT} \tag{10}$$

$$RE = \frac{\sum_{i=1}^{NT} |C_i - C\wedge_i|}{\sum_{i}^{NT} C_i} \tag{11}$$

IV. RESULT

Ensuring the algorithm explores valid regions within the parameter space is crucial, and this requires maintaining feasibility in the generated solutions. Regarding the Garra Rufa algorithm, here is a technique to ensure that the produced solutions remain viable:

Approach for Achievable Solutions:

1. Boundary Validation:

Incorporate boundary checking during the initialization and movement of fish to guarantee that any solution generated or moved randomly stays within the feasible parameter space. When a fish exceeds the specified boundaries, its position is corrected to the closest boundary.

2. Dealing with Constraints:

Incorporate limitations into the assessment of fitness. If a solution fails to meet any constraints, such as physical or practical limitations, reduce the fitness value or implement corrective measures to restore the solution to feasibility.

3. Dynamic Constraint Management:

Adapt constraints in real-time according to the changing fitness landscape. If specific areas of the parameter space become unattainable due to fish movement, adjust the constraints accordingly.

4. Mechanisms for repairing:

Develop and incorporate corrective measures for solutions that are not feasible. When a fish produces an impractical solution or moves to an impractical area, implement repair strategies to alter the solution while preserving feasibility.

This may entail modifying specific parameters or transitioning the solution to a neighboring feasible state.

5. Variable step sizes that adjust according to specific conditions:

Modify the sizes of steps or distances of movement in a flexible and responsive manner. This hinders fish from executing substantial and impractical leaps in the parameter space. Taking smaller steps increases the probability of remaining within achievable areas.

6. Mechanism for receiving feedback:

Utilize the feedback obtained from the fitness evaluation to direct the locomotion of fish. If a specific area consistently produces solutions that are not possible, the algorithm can adjust by decreasing the amount of exploration in that direction.

7. Iterative Refinement:

Execute a repetitive improvement procedure. After each motion, verify the practicality and progressively enhance the solution until it complies with the limitations. This hinders the algorithm's advancement by disallowing solutions that are initially near feasible but later violate constraints during further exploration.

8. Leader Selection with Consideration for Constraints:

When choosing a leader fish, give priority to those who have practical solutions. This guarantees that followers are directed by leaders who already occupy viable areas, thereby minimizing the likelihood of exploring impractical territories.

By integrating these tactics, the Garra Rufa algorithm can proficiently navigate the parameter space while consistently upholding the feasibility of the produced solutions. The algorithm's adaptive approach enables it to efficiently navigate intricate terrains and converge towards viable and high-caliber solutions.

In order to determine which kinetics are particularly vulnerable to change, a sensitivity analysis is performed to reveal how many outputs are impacted by alterations to individual kinetic parameters. Since the kinetic parameters span such a broad range, simultaneous optimization of all 172 of them is a daunting task. As a result, the most sensitive kinetics were isolated via sensitivity analysis. All kinetics were examined up to a disturbance of 200%. In addition, the simulation identified the seven most effective kinetic parameters out of a total of 172, and these were designated as the parameters that needed to be estimated. The remaining kinetics were left at their original values because their effectiveness dropped below 20% when the perturbation was increased to 200% [10]. Those kinetic parameters are (v_{max}^{pyk}, n_{pk}, $icdh$, k_{icdh}^{f}, $k_{icdhnadp}^{d}$, $k_{icdhnadp}^{m}$, and v_{max}^{icl}), which represent the reaction rates of (pyr, $icdh$, icl).

Moreover, Table 3 shows the nominal, the upper, the lower and the estimated kinetic parameters for GRO algorithm. While the estimated kinetic parameters for PSO was taken from [11].

The kinetic parameter estimation problem (Eq 1) was solved based on experimental data taken from literature for an *E coli* continuous culture perturbed with a glucose pulse [9].

TABLE 1. The kinetic estimation.

Kinetics	Nominal value	Lower bound	Upper bound	Estimated kinetics
v_{max}^{pyk}	1.08500	0.9000	1.4000	0.93000
n_{pk}	3.00000	2.6000	3.3000	3.19000
$icdh$	24.4210	24.100	24.900	24.5700
k_{icdh}^{f}	289800	289780	289850	289843
$k_{icdhnadp}^{d}$	0.0060	0.0030	0.0150	0.00450
$k_{icdhnadp}^{m}$	0.0170	0.0120	0.0300	0.01500
v_{max}^{icl}	3.8315	3.5000	4.1000	3.94000

TABLE 2. The relative error (RE) and the mean error (ME).

Metabolites	GRO-RE	GRO-ME	PSO-RE	PSO-ME
Glc^{ex}	2.31%	-0.013	14.29%	-0.124
$G6P$	7.63%	-0.124	23.5%	-0.271
$F6P$	41.25%	-0.437	54.32%	-0.524
FDP	3.46%	-0.06	7.81%	0.03
PEP	15.52%	-0.358	18.14%	-0.375
$6PG$	4.3%	-0.127	12.47%	-0.214
$Ru5P$	11.5%	1.230	27.72%	2.17
$Xu5P$	19.6%	-0.173	32.29%	-0.261
$2KGDH$	24.3%	0.641	35.4%	0.719
S7P	34.57%	-0.165	41.51%	-0.283

Temporal profiles for extracellular glucose (Glc^{ex}), Glucose-6-phosphate ($G6P$), Fructose 6-phosphate ($F6P$), Fructose 1,6-Phosphate (FDP), Glyceraldehyde 3-phosphate (GAP), Phosphoenol-pyruvate (PEP), 6-Phosphogluconolactone ($6PG$), Ribose 5-phosphate ($Ru5P$), Xylulose 5-phosphate ($Xu5P$), 2-Keto-D-gluconate ($2KG$), and Sedoheptulose 7-phosphate ($S7P$).

As shown in Figure 2, 3, and 4, the GRO and PSO algorithms showed perfect result.

Thus, Figure 2, 3, & 4 displays the estimated concentration patterns of key variables using both our proposed model and the model developed by [5], [9], and [20], as well as the observed data. We use the optimal parameter values from Table 1 to estimate the metabolite concentrations with our proposed model, and for the original model, we use the optimal values for kinetic parameters from [5], [9], and [20]. By comparing the simulated profiles in Figure 2, 3, & 4 it becomes obvious that our proposed model, together with the parameter estimation method we introduced in this work, leads to improved estimations of dynamic responses. Our GRO algorithm, is thus a useful methodology for solving kinetic parameter estimation problems because it leverages information on objective function and differential algebraic constraints gradients relative to optimization variables (the parameters to be estimated) evaluated throughout the time horizon in each integration of the DAE system (corresponding to each iteration of the outer optimization problem).

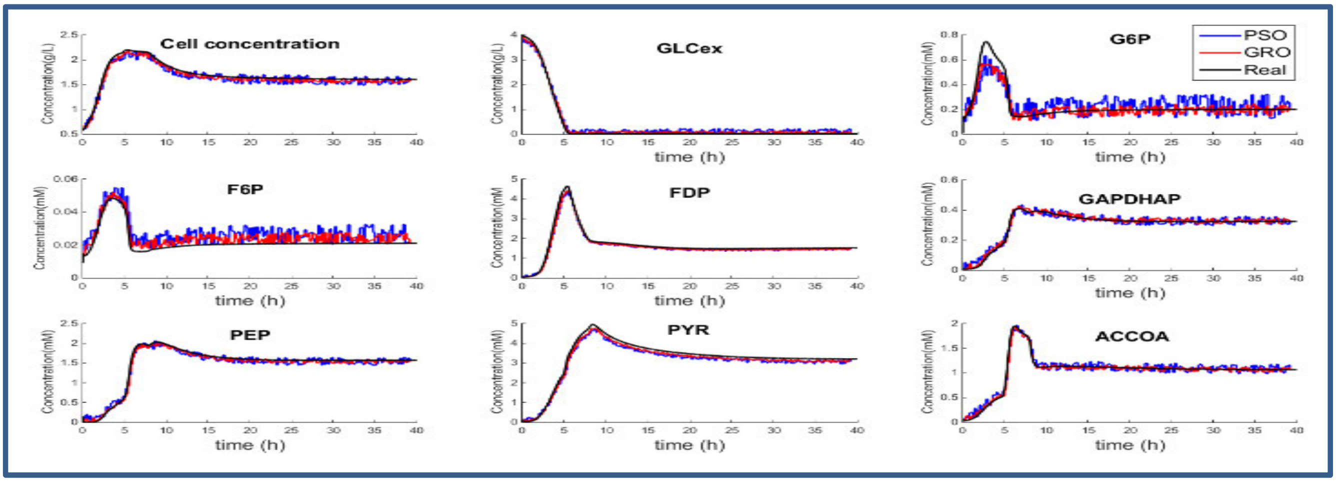

FIGURE 2. The Glycolysis, Anapleorotic and Acetate formation pathways simulation by GRO and PSO.

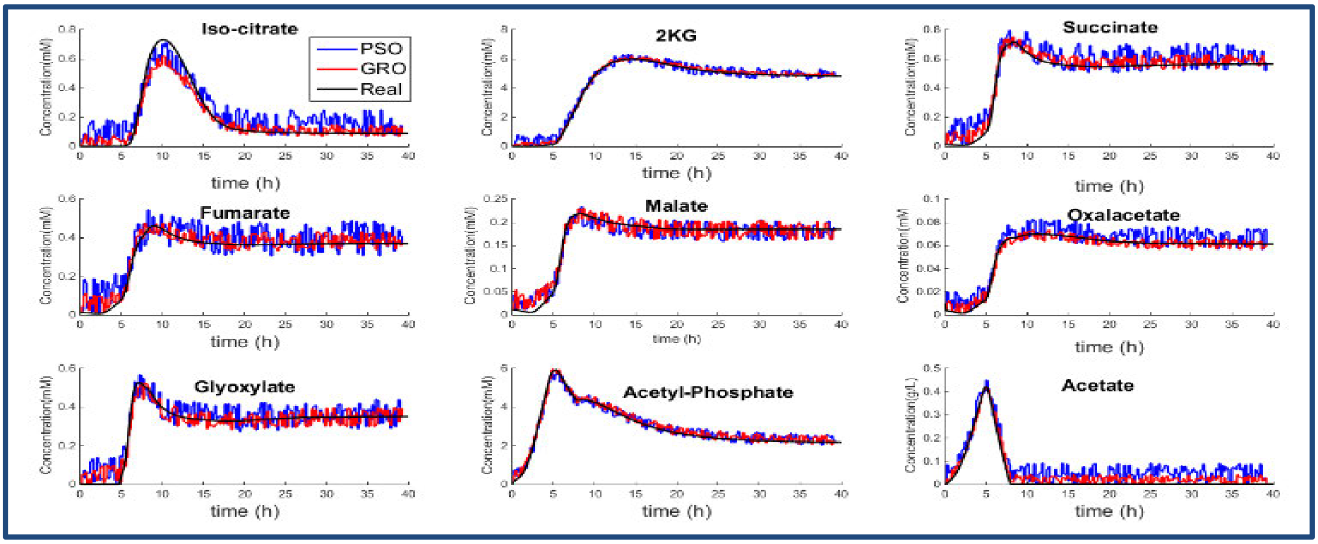

FIGURE 3. The TCA cycle, Anapleorotic, Glyoxylate, and Acetate formation pathways simulation by GRO and PSO.

Table 2 shows the relative error (RE) and the mean error (ME) from parameter estimation results for the model using GRO and PSO algorithms. Focusing on the results obtained for the kinetic estimation in this work by GRO, it becomes clear that the mean error values show a good fit between the observed data and model outputs for state variables. The ME, in the case of F6P, takes the highest value equal to −0.437 (Table 2).

Considering the relative error values, the best estimations are encountered for extracellular glucose Glc^{ex} (RE=2.31%), FDP (RE=3.46%), $6PG$ (RE=4.3%), $G6P$ (RE=7.63%), $Ru5P$ (RE=11.5%), PEP (RE=15.52%), $Xu5P$ (RE=19.6%), $2KGDH$ (RE=24.3%) show a reasonable fit. The largest discrepancy between the observed data and model prediction occurs for $S7P$ (RE=34.57%) with $F6P$ (RE=41.25%).

Furthermore, PSO algorithm shows that the mean error values have a good fit between the observed data and model outputs for state variables. The ME, in the case of F6P, takes the highest value equal to −0.524 (Table 2).

Taking into account the relative error values, the best estimations are encountered for FDP (RE=7.81%), $6PG$ (RE=12.47%), extracellular glucose Glc^{ex} (RE=14.29%), PEP (RE=18.14%), $G6P$ (RE=23.5%), $Ru5P$ (RE=27.72%), $Xu5P$ (RE=32.24%), $2KGDH$ (RE=35.5%) show a reasonable fit. The largest discrepancy between the observed data and model prediction occurs for $S7P$ (RE=41.51%) with $F6P$ (RE=54.32%).

In order to guarantee a fair comparison, we conscientiously selected effective parameter values for the PSO and GRO [31], [33] techniques, as detailed in Table 3. The parameters that were chosen for inclusion in this table are

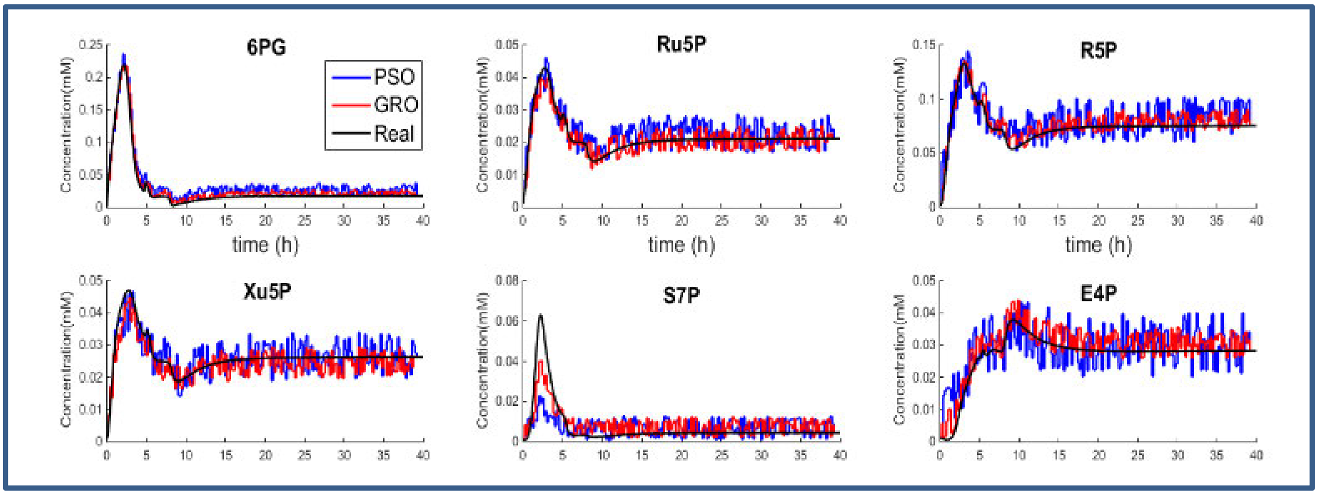

FIGURE 4. The Pentose Phosphate (PP) pathway simulation by GRO and PSO.

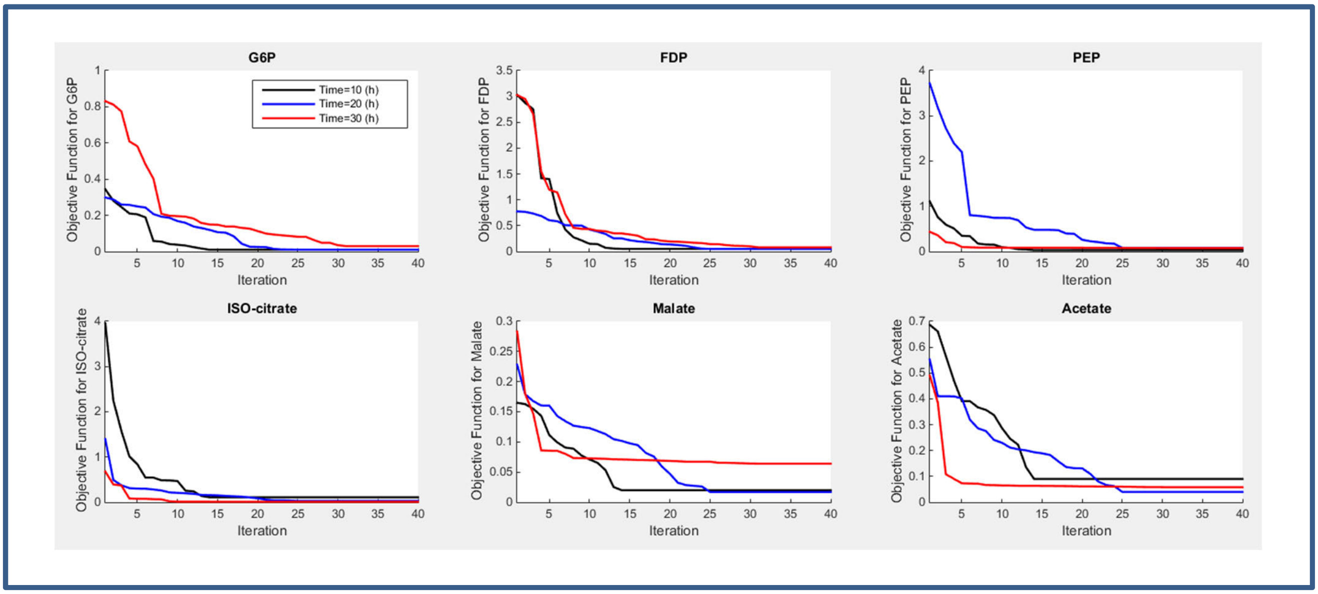

FIGURE 5. The GRO performance.

TABLE 3. PSO and GRO algorithm parameters.

parameter	ω	c1 and c2	Iteration	£	Particles numbers	TOC(sec)
PSO	0.7	1.9	40	---	25	783.8039
GRO	0.7	1.9	40	10%	25	807.0879

accompanied by information regarding the execution time (TOC) of each algorithm, which was precisely calculated at the 10-hour mark.

The convergence performance of the GRO algorithm in six different scenarios involving the *E. coli* model is depicted in Figure 5. This exhaustive analysis encompasses three distinct time intervals: precisely, 10, 20, and 30 hours. The visual depicted in Figure 5 illustrates the performance of the convergent-to-optimal point method using the GRO algorithm throughout various temporal phases of the *E. coli* model.

V. CONCLUSION

Our study introduces an innovative method for estimating kinetic parameters in a complex metabolic network that includes the phosphotransferase system, glycolysis, pentose phosphate routes, and fermentation pathways. To address the difficulty of optimizing 172 kinetic parameters at once, we performed a thorough sensitivity analysis to determine the seven most important parameters for estimation.

The Garra Rufa Optimization (GRO) algorithm, which draws inspiration from the skin-cleaning behavior of the

TABLE 4. Model mass balance equations.

Metabolites	Mass balance description
Cell	$\frac{d[X]}{dt} = \mu[X]$
Extra Glucose ($\boldsymbol{GLC^{ex}}$)	$\frac{d[GLC^{ex}]}{dt} = -v_{PTS}[X]$
Glucose-6-phosphate ($\boldsymbol{G6P}$)	$\frac{d[G6P]}{dt} = v_{PTS} - v_{PGI} - v_{G6PDH} - \mu[G6P]$
Fructose 6-phosphate ($\boldsymbol{F6P}$)	$\frac{d[F6P]}{dt} = v_{PGI} - v_{PFK} + v_{TKTB} + v_{TAL} - \mu[F6P]$
Fructose 1,6-Phosphate ($\boldsymbol{FDP}$)	$\frac{d[FDP]}{dt} = v_{PFK} - v_{ALDO} - \mu[FDP]$
Glyceraldehyde 3-phosphate ($\boldsymbol{GAPDH}$)	$\frac{d[GAPDH]}{dt} = 2v_{ALDO} - v_{GAPDH} + v_{TKTA} + v_{TKTB} - v_{TAL} - \mu[GAPDH]$
Phosphoenol-pyruvate ($\boldsymbol{PEP}$)	$\frac{d[PEP]}{dt} = v_{GAPDH} + v_{PCK} - v_{PTS} - v_{PYK} - v_{PPC} - \mu[PEP]$
Pyruvate ($\boldsymbol{PYR}$)	$\frac{d[PYR]}{dt} = v_{PYK} + v_{PTS} + v_{MEZ} - v_{PDH} - \mu[PYR]$
Acetyl-CoA ($\boldsymbol{AcCoA}$)	$\frac{d[AcCoA]}{dt} = v_{PDH} + v_{ACS} + v_{CS} - v_{PTA} - \mu[AcCoA]$
Isocitrate ($\boldsymbol{ICIT}$)	$\frac{d[ICIT]}{dt} = v_{CS} - v_{ICDH} - v_{ICL} - \mu[ICIT]$
2-Keto-D-gluconate ($\boldsymbol{2KG}$)	$\frac{d[2KG]}{dt} = v_{ICDH} - v_{2KGDH} - \mu[2KG]$
Succinate ($\boldsymbol{SUC}$)	$\frac{d[SUC]}{dt} = v_{2KGDH} + v_{ICL} - v_{SDH} - \mu[SUC]$
Fumarate ($\boldsymbol{FUM}$)	$\frac{d[FUM]}{dt} = v_{SDH} - v_{FUM} - \mu[FUM]$
Malate ($\boldsymbol{MAL}$)	$\frac{d[MAL]}{dt} = v_{FUM} + v_{MS} - v_{MDH} - v_{MEZ} - \mu[MAL]$
Oxaloacetate ($\boldsymbol{OAA}$)	$\frac{d[OAA]}{dt} = v_{MDH} + v_{PPC} - v_{CS} - v_{PCK} - \mu[OAA]$
Glyoxylate ($\boldsymbol{GOX}$)	$\frac{d[GOX]}{dt} = v_{ICL} - v_{MS} - \mu[GOX]$
Acetyl phosphate ($\boldsymbol{ACP}$)	$\frac{d[ACP]}{dt} = v_{PTA} - v_{ACK} - \mu[ACP]$
Acetate ($\boldsymbol{ACE^{ex}}$)	$\frac{d[ACE^{ex}]}{dt} = (v_{ACK} - v_{ACS})[X]$
6-Phosphogluconolactone ($\boldsymbol{6PG}$)	$\frac{d[6PG]}{dt} = v_{G6PDH} - v_{6PGDH} - \mu[6PG]$
Ribose 5-phosphate ($\boldsymbol{Ru5P}$)	$\frac{d[Ru5P]}{dt} = v_{6PGDH} - v_{RPE} - v_{RPI} - \mu[Ru5P]$
Ribulose 5-phosphoenolpyruvate ($\boldsymbol{R5P}$)	$\frac{d[R5P]}{dt} = v_{RPI} - v_{TKTA} - \mu[R5P]$
Xylulose 5-phosphate ($\boldsymbol{Xu5P}$)	$\frac{d[Xu5P]}{dt} = v_{RPE} - v_{TKTA} - v_{TKTB} - \mu[Xu5P]$
Sedoheptulose 7-phosphate ($\boldsymbol{S7P}$)	$\frac{d[S7P]}{dt} = v_{TKTA} - v_{TAL} - \mu[S7P]$
Erythrose 4-phosphate ($\boldsymbol{E4P}$)	$\frac{d[E4P]}{dt} = v_{TAL} - v_{TKTB} - \mu[E4P]$

TABLE 5. Kinetic rate equations.

Reaction's	Kinetic equation
Cell growth	$\begin{cases} \mu_m\left(1-\frac{[X]}{X_m}\right)\left(\frac{[GLc^{ex}]}{K_s+[GLc^{ex}]}\right)k_{ATP}v_{ATP}(.),([GLc^{ex}]>0) \\ \frac{\mu_{mA}[Ace^{ex}]}{K_{sA}+[Ace^{ex}]}\ k_{ATP}v_{ATP}(.),([GLc^{ex}]\leq 1\ and[Ace^{ex}]>0) \end{cases}$
Phosphotransferase system (PTS)	$\frac{v_{PTS}^{max}[GLc^{ex}]\frac{[PEP]}{PYR}}{\left(K_{a1}+K_{a2}\frac{[PEP]}{[PYR]}+K_{a3}[GLc^{ex}]+[GLc^{ex}]\frac{[PEP]}{[PYR]}\right)\left(1+\frac{[G6P]^nG6P}{K_{G6P}}\right)}$
Phosphoglucose isomerase (PGI)	$\frac{v_{PGI}^{max}\left([G6P]-\frac{[F6P]}{K_{eq}}\right)}{K_{G6P}\left(1+\frac{[F6P]}{K_{F6P}\left(1+\frac{[F6P]}{K_{6pginh}^{F6P}}\right)}+\frac{[6PG]}{K_{6pginh}^{G6P}}\right)+G6P}$
Phosphofructokinase (PFK)	$\frac{v_{PFK}^{max}K_{ATP}\,[F6P]}{K_{(ATP\,,ADP)}\left([F6P]+K_z^{F6P}\frac{K_{b(ADP,AMP)}+\frac{[PEP]}{K_{PEP}}}{K_{a(ADP,AMP)}}\right)\left(1+\frac{L_{pfk}}{\left(1+[F6P]\left(\frac{K_{a(ADP,AMP)}}{K_s^{F6P}\left(K_{b(ADP,AMP)}+\frac{[PEP]}{K_{PEP}}\right)}\right)\right)^{n_{PFK}}}\right)}$
Aldolase (Aldo)	$\frac{v_{ALDO}^{max}\left([FDP]-\frac{[DHAP][GAP]}{K_{eq}}\right)}{\left(K_{FDP}+[FDP]+\frac{K_{GAP}[DAHP]}{[K_{eq}V_{bif}]}+\frac{K_{DHAP}[GAP]}{[K_{eq}V_{bif}]}+\frac{[FDP][GAP]}{K_{inh}^{PEP}}+\frac{[DHAP][GAP]}{K_{eq}V_{bif}}\right)}$
Glyceraldehydes-3-phosphate dehydrogenase (GAPDH)	$\frac{v_{GAPDH}^{max}\left([GAP]-\frac{[PEP][NADH]}{K_{eq}[NAD]}\right)}{\left(K_{GAP}\left(1+\frac{[PEP]}{K_{PGP}}\right)+[GAP]\right)\left(\frac{K_{NAD}}{NAD}\left(1+\frac{[NADH]}{K_{NADH}}\right)+1\right)}$
Pyruvate kinase (PYK)	$\frac{v_{PYK}^{max}\,[PEP]\left(\frac{PEP}{K_{PEP}}+1\right)^{n_{pyk^{-1}}}[ADP]}{K_{PEP}\left(L_{PYK}\left(\frac{1+\frac{[ATP]}{K_{ATP}}}{\frac{[FDP]}{K_{FDP}}+\frac{[AMP]}{K_{AMP}}+1}\right)^{n_{pyk}}+\left(\frac{[PEP]}{K_{PEP}}+1\right)^{n_{pyk}}\right)([ADP]+K_{ADP})}$
Phosphoenolpyruvate carboxylase (Ppc)	$\frac{K_1+K_2[AcCOA]+K_3[FDP]+K_4[AcCOA][FDP]}{1+K_5[AcCOA]+K_6[FDP]}\left(\frac{[PEP]}{K_m+[PEP]}\right)$
Glucose-6-phosphate dehydrogenaseG(6PDH)	$\frac{v_{G6PDH}^{max}[G6P][NADP]}{([G6P]+K_{g6p})\left(1+\frac{[NADPH]}{K_{ndph}}\right)\left(K_{nadp}\left(1+\frac{[NADPH]}{K_{nadph}}\right)+NADP\right)}$
6-Phosphogluconate dehydrogenase (PGDH)	$\frac{v_{PGDH}^{max}[6PG][NADP]}{([6PG]+K_{6pg})\left([NADP]+K_{nadp}\left(1+\frac{[NADPH]}{K_{nadph}}\right)\left(1+\frac{[ATP]}{K_{atp}}\right)\right)}$
Ribose-phosphate epimerase (Rpe)	$v_{Rpe}^{max}\left([Ru5P]-\frac{[R5P]}{K_{eq}^{Rpe}}\right)$
Ribose-phosphate isomerase (Rpi)	$v_{Rpi}^{max}\left([Ru5P]-\frac{[R5P]}{K_{eq}^{Rpi}}\right)$
Transketolase 1 (TktA)	$v_{TKtA}^{max}\left([R5P][Xu5P]-\frac{[S7P][GAP]}{K_{eq}^{TKtA}}\right)$

TABLE 5. ***(Continued.)*** **Kinetic rate equations.**

Transketolase 2 (TktB)	$v_{TKtB}^{max}\left([Xu5P][E4]-\frac{[F6P][GAP]}{K_{eq}^{TKtB}}\right)$
Tyrosine ammonia lyase (Tal)	$v_{TaL}^{max}\left([GAP][S7P]-\frac{[E4P][F6P]}{K_{eq}^{TKtB}}\right)$
Phosphoenolpyruvate carboxykinase (PcK)	$v_{PcK}^{max}\left(\frac{[OAA]\frac{[ATP]}{[ADP]}}{K_m^{OAA}\frac{[ATP]}{[ADP]}+[OAA]\frac{[ATP]}{[ADP]}+\frac{K_i^{ATP}K_m^{OAA}}{K_i^{ADP}}+\frac{K_i^{ATP}K_m^{OAA}}{K_m^{PEP}K_i^{ADP}}[PEP]+\frac{K_i^{ATP}K_m^{OAA}}{K_i^{PEP}K_l^{ATP}}\frac{[ATP][PEP]}{[ADP]}+\frac{K_i^{ATP}K_m^{OAA}}{K_i^{ADP}K_l^{OAA}}[OAA]}\right)$
Pyruvate dehydrogenase (PDH)	$\frac{\frac{v_{PDH}^{max}}{[NAD]}\left(\frac{1}{1+K_i\frac{[NADH]}{[NAD]}}\right)\left(\frac{[PYR]}{K_m^{PYR}}\right)\left(\frac{1}{K_m^{NAD}}\right)\left(\frac{[COA]}{K_m^{COA}}\right)}{\left(1+\frac{[PYR]}{K_m^{PYR}}\right)\left(\frac{1}{NAD}+\frac{1}{K_m^{NAD}}+\frac{[NADH]}{K_m^{NADH}[NAD]}\right)\left(1+\frac{[COA]}{K_m^{COA}}+\frac{[AcCOA]}{K_m^{AcCOA}}\right)}$
Phosphotransacetylase (Pta)	$\frac{v_{Pta}^{max}\left(\frac{1}{K_i^{AcCOA}K_m^{P}}\right)\left([AcCoA][P]-\frac{[AcP][CoA]}{K_{eq}}\right)}{\left(1+\frac{[AcCoA]}{K_i^{AcCoA}}+\frac{[P]}{K_i^{P}}+\frac{[ACP]}{K_i^{ACP}}+\frac{[CoA]}{K_i^{CoA}}+\left(\frac{[AcCoA][P]}{K_i^{AcCoA}K_m^{P}}\right)+\left(\frac{[AcP][CoA]}{K_m^{ACP}K_i^{CoA}}\right)\right)}$
Acetate Kinase (Ack)	$\frac{v_{Ack}^{max}\left(\frac{1}{K_m^{ADP}K_m^{ACP}}\right)\left([AcP][ADP]-\frac{[ACE][ATP]}{K_{eq}}\right)}{\left(1+\frac{[Acp]}{K_m^{AcP}}+\frac{[ACE]}{K_m^{ACE}}\right)\left(1+\frac{[ADP]}{K_m^{ADP}}+\frac{[ATP]}{K_m^{ATP}}\right)}$
Acetyl-CoA synthetase (Acs)	$\frac{v_{Acs}^{max}[ACE][NADP]}{(K_m+[ACE])(K_{eq}+[NADP])}$
Citrate synthase (Cs)	$\frac{v_{CS}^{max}[AcCoA][OAA]}{\left(K_d^{AcCoA}K_m^{OAA}+K_m^{AcCoA}[OAA]\right)+\left([AcCoA]K_m^{OAA}\left(1+\frac{[NADH]}{K_{i1}^{NADH}}\right)\right)+\left([AcCoA][OAA]\left(1+\frac{[NADH]}{K_{i2}^{NDAH}}\right)\right)}$
Isocytrate dehydrogenase (ICDH)	$\frac{[ICDH]\frac{K_f}{K_m^{ICIT}}K_d^{NADP}\left([ICIT]-\frac{[NADH][2KG]}{K_{eq}^{ICDH}[NADP]}\right)}{\left(\frac{1}{[NADP]}+\frac{[ICIT]K_m^{NADP}}{K_m^{ICIT}K_d^{NADP}[NADP]}+\frac{1}{K_d^{NADP}}+\frac{[ICIT]}{K_m^{ICIT}K_d^{NADP}}+\frac{[ICIT]}{K_d^{ICIT}[NADP]}\frac{[NADPH]K_m^{NADP}}{K_m^{ICIT}K_m^{NADP}K_{einh}^{NADPH}}+\frac{[NADPH]K_{eknh}^{2KG}}{K_m^{2KG}K_{enhe}^{NADPH}[NADP]}+\frac{[2KG]K_m^{NADPH}}{K_m^{2KG}K_{enhe}^{NADPH}[NADP]}+\frac{[2KG]}{K_m^{2KG}}\frac{[NADPH]}{K_{enhe}^{NADPH}[NADP]}+\frac{[2KG]K_m^{NADPH}}{K_m^{2KG}K_m^{NADPH}}\frac{[NADPH]}{K_{ekn}^{NADP}[NADP]}\right)}$
Isocitrate lyase (IcL)	$\frac{v_{lcl-f}^{max}\frac{[ICIT]}{K_m^{ICIT}}}{\left(1+\frac{[ICIT]}{K_m^{ICIT}}+\frac{[SUC]}{K_m^{SUC}}+\frac{[PEP]}{K_m^{PEP}}+\frac{[2KG]}{K_m^{2KG}}+\frac{1}{K_l}\right)}$
Isocitrate lyase (MS)	$\frac{v_{MS}^{max}\frac{[GOX]}{K_m^{GOX}}\frac{[AcCoA]}{K_m^{AcCoA}}-v_{MS}^{max}\frac{[MAL]}{K_m^{MAL}}}{\left(1+\frac{[GOX]}{K_m^{GOX}}+\frac{[MAL]}{K_m^{MAL}}+\left(1+\frac{[AcCoA]}{K_m^{AcCoA}}\right)\right)}$
2-Ketoglutarate (**2KGDH**)	$\frac{v_{2KGDH}^{max}[aKG][CoA]}{\left(\frac{K_m^{NAD}[aKG][CoA]}{[NAD]}+K_m^{CoA}[aKG]+K_m^{2KG}[CoA]+[aKG][CoA]+\frac{K_m^{2KG}K_z[aKG][SUC][NADH]}{K_1^{2KG}K_1^{SUC}[NAD]}+\frac{K_m^{2KG}K_z[SUC][NADH]}{K_1^{SUC}[NAD]}+\frac{K_m^{NAD}[aKG][CoA][NADH]}{K_1^{NADH}[NAD]}+\frac{K_m^{CoA}[aK][SUC]}{K_1^{SUC}}\right)}$
Succinate dehydrogenase (SDH)	$\frac{v_{SDH1}v_{SDH2}\left([SUC]-\frac{[FUM]}{K_{eq}}\right)}{K_m^{SUC}v_{SDH2}+v_{SDH2}[SUC]+\frac{v_{SDH1}[FUM]}{K_{eq}}}$
Fumarate hydratase (Fum)	$\frac{v_{Fum1}v_{Fum2}\left([FUM]-\frac{[MAL]}{k_{Fum\,eq}}\right)}{K_m^{Fum}v_{Fum1}+v_{Fum2}[FUM]+\frac{V_{Fum1}[MAL]}{K_{eq}}}$
Malic enzyme (Mez)	$\frac{v_{Mez}^{max}[MAL][NADP]}{(K_{MAL}+[MAL])(K_{eq}+[NADP])}$
Malate dehydrogenase (MDH)	$\frac{v_{MDH1}v_{MDH2}\left([MAL]-\frac{[OAA]}{K_{eq}}\right)}{\left(\frac{K_1^{NAD}K_m^{MAL}v_{MDH2}}{[NAD]}+K_m^{MAL}v_{MDH2}+\frac{K_m^{NAD}v_{MDH2}[MAL]}{[NAD]}+v_{MDH2}[MAL]+\frac{K_m^{OAA}v_{MDH1}[NADH]}{K_{eq}[NAD]}+\frac{K_m^{NADH}v_{MDH1}[OAA]}{K_{eq}[NAD]}+\frac{v_{MDH1}[NADH][OAA]}{K_{eq}[NAD]}+\frac{v_{MDH1}K_m^{OAA}[NADH]}{K_{eq}K_1^{NAD}}+\frac{v_{MDH2}K_m^{NAD}[MAL][OAA]}{K_1^{OAA}[NAD]}+\frac{v_{MDH2}[MAL][NADH]}{K_1^{NAD}}+\frac{v_{MDH1}[MAL][NADH][OAA]}{K_{eq}K_1^{MAL}[NAD]}+\frac{v_{MDH2}[MAL][OAA]}{K_{II}^{OAA}}+\frac{v_{MDH1}[NADH][OAA]}{K_{II}^{NAD}K_{eq}}+\frac{K_1^{NAD}v_{MDH2}[MAL][NADH][OAA]}{K_{II}^{NAD}K_m^{OAA}K_1^{NADH}}\right)}$

Garra Rufa fish, has demonstrated its efficacy as a potent optimization tool. In addition to GRO, we utilized the Particle Swarm Optimization (PSO) algorithm for the purpose of comparison. The results of our study unequivocally establish the GRO algorithm as superior in accurately and effectively estimating the specific kinetic parameters of interest. Furthermore, our findings highlight its ability to handle models of different levels of complexity with ease.

By using experimental data obtained from a continuous culture of *E. coli* that was exposed to a sudden increase in glucose, our simulations produced time-dependent profiles for various metabolites. The visual depictions in Figures 2, 3, & 4 demonstrate the improved accuracy of our proposed GRO-driven model in comparison to models developed by other researchers.

The enhanced precision in capturing dynamic reactions highlights the potential of the GRO algorithm in advancing the modelling of metabolic networks.

Evaluating the model's performance can be done by assessing the relative error (RE) and mean error (ME) values, which offer a nuanced evaluation. Although there are some small differences for certain metabolites, the ME values, particularly for state variables such as F6P, confirm a strong agreement between the observed data and the predictions of the model. The relative error values provide additional evidence of a satisfactory fit for most variables, thereby strengthening the credibility of our proposed GRO-driven approach.

Building upon these discoveries, it is important to mention that the GRO algorithm not only performed exceptionally well in terms of efficiency and accuracy, but also showcased computational benefits, making it especially suitable for models of a large scale. The algorithm consistently demonstrated its capacity to efficiently navigate the parameter space and converge on optimal solutions.

Our proposed GRO-driven methodology provides a practical and transformative tool for systems biology, offering the potential for advancements in comprehending and manipulating complex biological systems. Our approach offers a strong framework for precise parameter estimation in the study of intricate metabolic networks, making a valuable contribution to the field of computational biology research.

CONFLICTS OF INTEREST

"The authors declare no conflict of interest."

APPENDIX

The appendix is showing the model mass balance and the kinetic rate equations Tables 4&5.

REFERENCES

[1] J. C. Mason and M. W. Covert, "An energetic reformulation of kinetic rate laws enables scalable parameter estimation for biochemical networks," *J. Theor. Biol.*, vol. 461, pp. 145–156, Jan. 2019.

[2] M. A. Kunna, T. A. A. Kadir, M. A. Remli, N. M. Ali, K. Moorthy, and N. Muhammad, "An enhanced segment particle swarm optimization algorithm for kinetic parameters estimation of the main metabolic model of *Escherichia coli*," *Processes*, vol. 8, no. 8, p. 963, Aug. 2020.

[3] A. F. Villaverde, D. Henriques, K. Smallbone, S. Bongard, J. Schmid, D. Cicin-Sain, A. Crombach, J. Saez-Rodriguez, K. Mauch, E. Balsa-Canto, P. Mendes, J. Jaeger, and J. R. Banga, "BioPreDyn-bench: A suite of benchmark problems for dynamic modelling in systems biology," *BMC Syst. Biol.*, vol. 9, no. 1, pp. 1–5, Dec. 2015.

[4] M. A. Kunna, T. A. A. Kadir, and A. S. Jaber, "Sensitivity analysis in large-scale of metabolic network of *E. coli*," in *Proc. Int. Conf. Adv. Comput. Sci. Appl. Technol.*, Dec. 2013, pp. 346–351.

[5] C. Chassagnole, N. Noisommit-Rizzi, J. W. Schmid, K. Mauch, and M. Reuss, "Dynamic modeling of the central carbon metabolism of *Escherichia coli*," *Biotechnol. Bioeng.*, vol. 79, no. 1, pp. 53–73, Jul. 2002.

[6] Y. Usuda, Y. Nishio, S. Iwatani, S. J. Van Dien, A. Imaizumi, K. Shimbo, N. Kageyama, D. Iwahata, H. Miyano, and K. Matsui, "Dynamic modeling of *Escherichia coli* metabolic and regulatory systems for amino-acid production," *J. Biotechnol.*, vol. 147, no. 1, pp. 17–30, May 2010.

[7] Y. Matsuoka and K. Shimizu, "Catabolite regulation analysis of *Escherichia coli* for acetate overflow mechanism and co-consumption of multiple sugars based on systems biology approach using computer simulation," *J. Biotechnol.*, vol. 168, no. 2, pp. 155–173, Oct. 2013.

[8] O. Kotte, J. B. Zaugg, and M. Heinemann, "Bacterial adaptation through distributed sensing of metabolic fluxes," *Mol. Syst. Biol.*, vol. 6, no. 1, p. 355, 2010.

[9] T. A. A. Kadir, A. A. Mannan, A. M. Kierzek, J. McFadden, and K. Shimizu, "Modeling and simulation of the main metabolism in *Escherichia coli* and its several single-gene knockout mutants with experimental verification," *Microbial Cell Factories*, vol. 9, no. 1, pp. 1–21, Dec. 2010.

[10] M. A. K. Azrag, T. A. A. Kadir, and A. S. Jaber, "Segment particle swarm optimization adoption for large-scale kinetic parameter identification of *Escherichia coli* metabolic network model," *IEEE Access*, vol. 6, pp. 78622–78639, 2018.

[11] M. A. Kunna, T. A. A. Kadir, A. S. Jaber, and J. B. Odili, "Large-scale kinetic parameter identification of metabolic network model of *E. coli* using PSO," *Adv. Bioscience Biotechnol.*, vol. 6, no. 2, p. 120, 2015.

[12] S. Ceric, Z. Kurtanjek, S. Ceric, and Z. Kurtanjek, "Model identification, parameter estimation, and dynamic flux analysis of *E. coli* central metabolism," *Chem. Biochem. Eng. Quart.*, vol. 20, no. 3, pp. 243–253, 2006.

[13] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965.

[14] W. Won, C. Park, C. Park, S. Y. Lee, K. S. Lee, and J. Lee, "Parameter estimation and dynamic control analysis of central carbon metabolism in *Escherichia coli*," *Biotechnol. Bioprocess Eng.*, vol. 16, no. 2, pp. 216–228, Apr. 2011.

[15] H. Qin, X. Ma, T. Herawan, and J. M. Zain, "MGR: An information theory based hierarchical divisive clustering algorithm for categorical data," *Knowl.-Based Syst.*, vol. 67, pp. 401–411, Sep. 2014.

[16] Y. Tohsato, K. Ikuta, A. Shionoya, Y. Mazaki, and M. Ito, "Parameter optimization and sensitivity analysis for large kinetic models using a real-coded genetic algorithm," *Gene*, vol. 518, no. 1, pp. 84–90, Apr. 2013.

[17] J. Di Maggio, C. Paulo, V. Estrada, N. Perotti, J. C. D. Ricci, and M. S. Diaz, "Parameter estimation in kinetic models for large scale biotechnological systems with advanced mathematical programming techniques," *Biochem. Eng. J.*, vol. 83, pp. 104–115, Feb. 2014.

[18] A. F. Villaverde, F. Fröhlich, D. Weindl, J. Hasenauer, and J. R. Banga, "Benchmarking optimization methods for parameter estimation in large kinetic models," *Bioinformatics*, vol. 35, no. 5, pp. 830–838, Mar. 2019.

[19] A. Sagar, R. LeCover, C. Shoemaker, and J. Varner, "Dynamic optimization with particle swarms (DOPS): A meta-heuristic for parameter estimation in biochemical models," *BMC Syst. Biol.*, vol. 12, no. 1, pp. 1–5, Dec. 2018.

[20] M. A. Hoque, H. Ushiyama, M. Tomita, and K. Shimizu, "Dynamic responses of the intracellular metabolite concentrations of the wild type and pykA mutant *Escherichia coli* against pulse addition of glucose or NH3 under those limiting continuous cultures," *Biochem. Eng. J.*, vol. 26, no. 1, pp. 38–49, Nov. 2005.

[21] M. A. K. Azrag, T. A. A. Kadir, and M. A. Ismail, "A review of large-scale kinetic parameters in metabolic network model of *Escherichia coli*," *Adv. Sci. Lett.*, vol. 24, no. 10, pp. 7512–7518, Oct. 2018.

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

[23] S. M. Baker, K. Schallau, and B. H. Junker, "Comparison of different algorithms for simultaneous estimation of multiple parameters in kinetic metabolic models," *J. Integrative Bioinf.*, vol. 7, no. 3, pp. 254–262, Dec. 2010.

[24] Z. Chen, H. Liu, Y. Tian, R. Wang, P. Xiong, and G. Wu, "A particle swarm optimization algorithm based on time-space weight for helicopter maritime search and rescue decision-making," *IEEE Access*, vol. 8, pp. 81526–81541, 2020.

[25] S. N. Ghorpade, M. Zennaro, B. S. Chaudhari, R. A. Saeed, H. Alhumyani, and S. Abdel-Khalek, "Enhanced differential crossover and quantum particle swarm optimization for IoT applications," *IEEE Access*, vol. 9, pp. 93831–93846, 2021.

[26] J. J. Jamian, M. N. Abdullah, H. Mokhlis, M. W. Mustafa, and A. H. A. Bakar, "Global particle swarm optimization for high dimension numerical functions analysis," *J. Appl. Math.*, vol. 2014, pp. 1–14, Feb. 2014.

[27] J. B. Odili, M. N. M. Kahar, S. Anwar, and M. A. K. Azrag, "A comparative study of African buffalo optimization and randomized insertion algorithm for asymmetric travelling Salesman's problem," in *Proc. 4th Int. Conf. Softw. Eng. Comput. Syst. (ICSECS)*, Aug. 2015, pp. 90–95.

[28] M. A. K. Azrag, J. M. Zain, T. A. A. Kadir, M. Yusoff, A. S. Jaber, H. S. M. Abdlrhman, Y. H. Z. Ahmed, and M. S. B. Husain, "Estimation of small-scale kinetic parameters of *Escherichia coli* (*E. coli*) model by enhanced segment particle swarm optimization algorithm ESe-PSO," *Processes*, vol. 11, no. 1, p. 126, Jan. 2023.

[29] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, Oct. 1995, pp. 39–43.

[30] A. S. Jaber, A. Z. Ahmad, and A. N. Abdalla, "A new parameters identification of single area power system based LFC using segmentation particle swarm optimization (SePSO) algorithm," in *Proc. IEEE PES Asia–Pacific Power Energy Eng. Conf. (APPEEC)*, Dec. 2013, pp. 1–6.

[31] A. S. Jaber, H. A. Abdulbari, N. A. Shalash, and A. N. Abdalla, "Garra Rufa-inspired optimization technique," *Int. J. Intell. Syst.*, vol. 35, no. 11, pp. 1831–1856, Nov. 2020.

[32] V. A. Krishnan and N. S. Kumar, "Robust soft computing control algorithm for sustainable enhancement of renewable energy sources based microgrid: A hybrid Garra Rufa fish optimization—Isolation forest approach," *Sustain. Comput., Informat. Syst.*, vol. 35, Sep. 2022, Art. no. 100764.

[33] R. K. Chillab, A. S. Jaber, M. B. Smida, and A. Sakly, "Optimal DG location and sizing to minimize losses and improve voltage profile using garra rufa optimization," *Sustainability*, vol. 15, no. 2, p. 1156, Jan. 2023.

[34] K. Elango, A. Prakash, and L. Umasankar, "Multiobjective optimization model for renewable energy sources and load demands uncertainty consideration for optimal design of hybrid combined cooling, heating, and power systems," *Int. J. Energy Res.*, vol. 46, no. 6, pp. 7840–7860, May 2022.

[35] A. Prakash, A. Shyam Joseph, R. Shanmugasundaram, and C. S. Ravichandran, "A machine learning approach-based power theft detection using GRF optimization," *J. Eng., Design Technol.*, vol. 21, no. 5, pp. 1373–1388, Nov. 2023.

[36] C. S. R. Reddy, B. V. Prasanth, and B. M. Chandra, "Active power management of grid-connected PV-PEV using a hybrid GRFO-ITSA technique," *Sci. Technol. Energy Transition*, vol. 78, p. 7, Jan. 2023.

[37] M. A. K. Azrag and T. Asmawaty, "Empirical study of segment particle swarm optimization and particle swarm optimization algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 8, pp. 1–6, 2019.

[38] N. Jahan, K. Maeda, Y. Matsuoka, Y. Sugimoto, and H. Kurata, "Development of an accurate kinetic model for the central carbon metabolism of *Escherichia coli*," *Microbial Cell Factories*, vol. 15, no. 1, pp. 1–9, Dec. 2016.

[39] M. A. Remli, S. Deris, M. S. Mohamad, S. Omatu, and J. M. Corchado, "An enhanced scatter search with combined opposition-based learning for parameter estimation in large-scale kinetic models of biochemical systems," *Eng. Appl. Artif. Intell.*, vol. 62, pp. 164–180, Jun. 2017.

[40] M. A. K. Azrag, T. A. A. Kadir, M. N. Kabir, and A. S. Jaber, "Large-scale kinetic parameters estimation of metabolic model of *Escherichia coli*," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 2, pp. 160–167, Apr. 2019.

[41] M. Rodriguez-Fernandez, J. A. Egea, and J. R. Banga, "Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems," *BMC Bioinf.*, vol. 7, no. 1, pp. 1–8, Dec. 2006.

[42] A. F. Villaverde, J. A. Egea, and J. R. Banga, "A cooperative strategy for parameter estimation in large scale systems biology models," *BMC Syst. Biol.*, vol. 6, no. 1, pp. 1–7, Dec. 2012.

[43] M. A. K. Azrag, T. A. A. Kadir, and N. M. Ali, "A comparison of particle swarm optimization and global African buffalo optimization," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 769, no. 1, Feb. 2020, Art. no. 012034.

JASNI MOHAMAD ZAIN (Senior Member, IEEE) received the bachelor's degree in computer science from the University of Liverpool, England, U.K., in 1989, and the Ph.D. degree from Brunel University, West London, U.K., in 2005.

She started her career as a Tutor with the University of Technology Malaysia (UTM), in 1997. She was the Deputy Director of Cybertechnology with Research Nexus UiTM and the Head of the Advanced Analytics and Engineering Centre (AAEC). She was the Dean of the Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, for eight years. She is currently the Director of the Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), UiTM, Shah Alam, Malaysia. She has been actively presenting papers and keynote addresses at national and international conferences. She has graduated with 15 Ph.D. and six master's students by research under her supervision and published more than 100 refereed articles. She has a patent file for digital watermarking. Her research interests include data mining, digital watermarking, image processing, and network security.

MOHAMMED ADAM KUNNA AZRAG received the B.Sc. degree (Hons.) in computer studies from National Ribat University, in 2010, and the M.Sc. and Ph.D. degrees in computer science from Universiti Malaysia Pahang, in 2015 and 2021, respectively. He is currently with the Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), UiTM, Shah Alam, Malaysia. His research interests include optimization algorithms, data analysis, bioinformatics, and databases.

SAIFUL FARIK MAT YATIN received the B.Sc. degree in information studies (information system management) and the master's degree in information management from UiTM and the Ph.D. degree in information management (electronic information and records management). He has held various academic or administrative posts, such as the Coordinator of the Master's Program, the Coordinator of Strategic Planning for the Faculty, and the Head of Strategic Planning (Information Management) with UiTM. He is currently an Associate Professor with the Faculty of Information Management, UiTM. He is also appointed as a Research Fellow with the Institute of Big Data Analytics and Artificial Intelligence (IBDAAI). Before joining UiTM, he was a Bank Officer, attached to the Human Resource Information System and Data Management, Recruitment, Placement Unit, and the Human Resource Division. He has published over 100 peer-reviewed journals, conference publications, and book chapters. His research interests include digitization, information management, knowledge management, management information systems, records, and archive management.

GHADAH ALDEHIM received the B.S. degree in computer science from King Saud University, the M.S. degree in information systems, and the Ph.D. degree in data mining from the University of East Anglia, U.K. She is currently an Assistant Professor with the Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Saudi Arabia. Her main research interests include data science, machine learning, data mining, and big data.

ZUHAIRA MUHAMMAD ZAIN received the bachelor's degree in information science and the master's degree in computer science from Universiti Kebangsaan Malaysia, in 2000 and 2007, respectively, and the Ph.D. degree in software engineering from Universiti Putra Malaysia, in 2013. She was a Programmer with Fujitsu System Business Malaysia Bhd, from 2000 to 2007. She is currently an Assistant Professor with the College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia. Her research interests include software engineering, data mining, machine learning, data analytics, software quality, and software evaluation.

HADIL SHAIBA received the B.S. degree in information technology from King Saud University, Riyadh, Saudi Arabia, in 2008, and the M.S. and Ph.D. degrees from Southern Methodist University, Dallas, TX, USA, in 2016. In 2012, she was a Research Assistant with the Intelligent Data Analysis Laboratory, Southern Methodist University. She is currently an Assistant Professor with the College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh. Her research interests include image retrieval, data mining, and applying machine learning in meteorology, health sciences, education, and accident analysis. Her awards include the Outstanding Graduate Student Award (Southern Methodist University, Bobby B. Lyle School of Engineering) and the Research Day Dean's Departmental Award (Southern Methodist University).

NAZIK ALTURKI received the Ph.D. degree in information systems from The University of Melbourne. She is currently an Assistant Professor with the Information Systems Department, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia. Her research interests include health informatics, big data, data analytics, and mining.

SAPIAH SAKRI (Member, IEEE) received the B.Sc. degree in computer science from Universiti Kebangsaan Malaysia, Malaysia, in 1985, the M.Sc. degree in information science from the University of Malaya, Malaysia, in 1997, and the Ph.D. degree in information security from Universiti Kebangsaan Malaysia, in 2007. She served the Malaysian public sector for 30 years. Her latest position (before retiring) was the Director of the Information Communication Technology Policy and Strategic Division, Prime Minister's Department. She is currently an Assistant Professor with the Department of Information Systems, Princess Nourah Bint Abdulrahman University, Saudi Arabia. She has published articles mainly in the domain of data analytics. Her research interests include information security management, cloud computing security, data analytics, and data science. She received the Highest Award from the Prime Minister of Malaysia, in 2012, for her achievement in initiating the SMART city known as eKL.

AZLINAH MOHAMED received the M.Sc. degree in artificial intelligence from the University of Bristol and the Ph.D. degree in decision support systems from the National University of Malaysia. Before this, she was a Tutor with the University of Bristol and a Research Fellow with the National University of Malaysia. She is currently a Professor with the Information Systems Center of Studies, Universiti Teknologi MARA, Malaysia, and the Institute of Big Data Analytics and Artificial Intelligence (IBDAAI), and also appointed as the Managing Director of YSL Strategic Alliances Sdn Bhd, where one of the major focus is on research and development for Green Sustainable Environmental Product in Digital Age. In addition, she also held an administrative post about academic development at the university level as the Head of Academic Development and a Special Officer on Academic Affairs and Development to the Vice-Chancellor and the Dean of the Faculty of Computer and Mathematical Sciences for seven years. She has published well over 150 peer-reviewed journals, conference publications, and book chapters internationally and locally. Besides that, she has also contributed as a keynote speaker, an examiner, and a reviewer to many conferences, journals, and university academic activities. Her research interests include big data, soft computing, artificial intelligence, and web-based decision support systems using intelligent agents in electronic government applications. She received the Outstanding Research Award, in 2020, and the Best Paper Award, in 2021, 2019, and 2018, for various conferences.

AQEEL S. JABER was born in Iraq, in 1977. He received the B.E. and M.E. degrees from the University of Technology, Bagdad, Iraq, in 2001 and 2007, respectively, and the Ph.D. degree from Universiti Malaysia Pahang, Pahang, Malaysia, in 2015. He has been holding lecturing positions with the Department of Electrical Power Engineering Techniques, Al-Ma'moon University College, Baghdad, since 2009. Until October 2021, he was an Associate Professor with the Al-Ma'moon University College. He is currently an Independent Researcher in Helsinki, Finland.