https://revistas.ucr.ac.cr/index.php/ingenieria/index
www.ucr.ac.cr / ISSN: 2215-2652

# Ingeniería

Revista de la Universidad de Costa Rica

ENERO/JUNIO 2026 - VOLUMEN 36 (1)

EDITORIAL UCR

UNIVERSIDAD DE COSTA RICA

Ingeniería. Revista de la Universidad de Costa Rica

Vol. 36. No. 1: 48-57, Enero-Junio, 2026. ISSN: 2215-2652. San José, Costa Rica

Ingeniería Química

# Developing a Computational Framework to Harness Genetic Algorithms for Enhanced Optimization of Bioprocesses

## Desarrollo de un marco computacional para la optimización mejorada de bioprocesos mediante algoritmos genéticos

*Leonardo Garro Mena*

*Associate Professor, Chemical Engineering Department, Universidad de Costa Rica, San José, Costa Rica.*
*correo: leonardo.garromena@ucr.ac.cr*

***Keywords:***

Bioprocess optimization, genetic algorithms, hybrid optimization, xylitol production, xylose fermentation.

**Abstract**

The optimization of bioprocesses frequently involves highly nonlinear dynamic models that challenge derivative-based methods. In this study, a kinetic model of xylose fermentation for xylitol production was optimized with a hierarchical genetic-algorithm (GA) framework. First, a secondary GA minimized an economic objective function that penalized residual substrate and rewarded product formation; the Runge Kutta Fehlberg (RKF45) scheme solved the model under a constant local-error tolerance. Subsequently, a primary GA tuned the secondary GA's population size and crossover fraction. The secondary GA alone revealed a "stability valley" at crossover fractions: $0.60 \leq cf \leq 0.90$, where convergence proved steady and standard deviations remained low. Benchmarks with the two-dimensional Ackley function confirmed that solution surface ruggedness, rather than dimensionality, governed GA performance. The hierarchical approach identified an optimal configuration: $cf = 0.53$ and population = 260, which lay outside the stability valley and elevated the objective function by 3.1 % relative to the best value within the valley ($0.6 \leq cf \leq 0.9$). No modifications were introduced to the kinetic model or to the economic criterion; the gain therefore arose exclusively from metaparameter tuning. However, computation time increased, indicating that hybrid strategies, such as variable-fidelity models or adaptive-mutation schemes, may be required for larger systems. Even so, the demonstrated improvement translates into meaningful economic benefits and underscores the value of systematic GA meta-optimization for industrial bioprocesses. The results provide a reproducible benchmark and a foundation for extending this framework to more complex models, hybrid metaheuristics, and machine-learning-guided parameter adjustment.

Recibido: *25/04/2025*
Aceptado: 0*3/11/2025*

***Palabras clave:***

Algoritmos genéticos, fermentación de xilosa, optimización de bioprocesos, optimización híbrida, producción de xilitol.

**Resumen**

La optimización de bioprocesos frecuentemente implica modelos dinámicos no lineales que desafían los métodos basados en derivadas. En este estudio, se optimizó un modelo cinético para producir xilitol, mediante un esquema jerárquico de algoritmos genéticos (GA). Primero, un GA secundario minimizó una función objetivo económica. Posteriormente, un GA primario ajustó el tamaño de población y la fracción de entrecruzamiento del GA secundario. El GA secundario reveló un "valle de estabilidad" en el intervalo de fracción de entrecruzamiento: $0.60 \leq cf \leq 0.90$, donde la convergencia fue estable y las desviaciones estándar se mantuvieron bajas. Las pruebas con la función de Ackley confirmaron que la rugosidad de la superficie de solución rige el desempeño del GA. El enfoque jerárquico identificó una configuración óptima: $cf = 0.53$ y población = 260, fuera del valle, que incrementó la función objetivo en un 3.1 % respecto al mejor valor dentro del valle. No se introdujeron modificaciones al modelo cinético ni al criterio económico, la ganancia se atribuye exclusivamente a la sintonización de los metaparámetros. Sin embargo, el tiempo de cómputo se incrementó, por tanto, sistemas de mayor tamaño podrían requerir estrategias híbridas, como modelos de fidelidad variable o esquemas de mutación adaptativa. Aun así, la mejora demostrada se traduce en beneficios económicos significativos y subraya el valor de la metaoptimización sistemática de los GA para bioprocesos industriales. Los resultados proporcionan un referente reproducible y una base para extender este marco a modelos más complejos, metaheurísticas híbridas y ajustes de parámetros guiados por aprendizaje automático.

DOI: 10.15517/ri.v36i1.64867

NOMENCLATURE

| | |
|---|---|
| $A_t$ | Cross-sectional area of the reactor ($m^2$). |
| $C_{gO2}$ | Oxygen concentration in the gas phase (g/L). |
| $C_{O2}$ | Dissolved oxygen concentration (g/L). |
| $C_P$ | Product concentration (g/L). |
| $C_S$ | Substrate concentration (g/L). |
| $C_X$ | Cell concentration (g/L). |
| $\boldsymbol{d}$ | Dependent or predetermined state variables. |
| $e$ | Estimated error between fourth and fifth-order solutions. |
| $F$ | Feed flow (L/h). |
| $f$ | Function. |
| $GA$ | Genetic algorithm. |
| $\boldsymbol{g}$ | Design variable vector. |
| $h_{initial}$ | Initial step size. |
| $K_{eq}$ | Equilibrium constant. |
| $K_La$ | Volumetric oxygen transfer coefficient ($h^{-1}$). |
| $k_1$, $k_2$, $k_3$, $k_4$, $k_5$, $k_6$ | Slopes used in the RKF45 method. |
| $N$ | Agitation speed (rpm). |
| $N_{CD}$ | Critical agitation speed (rpm). |
| $OF(x, d)$ | Objective function. |
| *Options_primary* | Options for the primary GA function. |
| *Options_secondary* | Options for the secondary GA function. |
| $\boldsymbol{p}$ | Constant parameters affecting the system. |
| $T$ | Temperature (°C). |
| $t$ | Time (h). |
| $t_0$ | Initial time (h). |
| $t_{final}$ | Final time (h). |
| $tol$ | Error control tolerance. |
| $U_g$ | Superficial gas velocity (m/s). |
| $V$ | Reactor volume (L). |
| $vvm$ | Average volumetric velocity ($h^{-1}$). |
| $\boldsymbol{x}$ | State vector of the system. |
| $\boldsymbol{x}_0$ | Initial value of the state vector x. |
| $x_4$, $x_5$ | Fourth and fifth-order solutions, respectively. |
| $Y_{P/S}$ | Product yield per substrate (g/g). |
| $Y_{X/S}$ | Biomass yield per substrate (g/g). |
| $\delta$ | Additional production parameters. |
| $\mu$ | Specific growth rate ($h^{-1}$). |
| $v_g$ | Volumetric gas flow (L/h). |

## I. INTRODUCTION

Many processes in the field of engineering involve complex phenomena that are represented by systems of equations of the form $d\boldsymbol{x}/dt = \boldsymbol{f}(t, \boldsymbol{x}; \boldsymbol{p}, \boldsymbol{d})$, where $\boldsymbol{x}$ is the state vector of the system, $\boldsymbol{d}$ are variables that can be functions of $\boldsymbol{x}$ or be predetermined, t is the time coordinate and $\boldsymbol{p}$ are parameters. The performance of the system is measured by an objective function $OF(\boldsymbol{x}, \boldsymbol{d})$ [1]. To achieve the best possible performance, $OF(\boldsymbol{x}, \boldsymbol{d})$ must be optimized (maximized or minimized) by changing the vector of decision variables ($\boldsymbol{g}$). In our notation, $\boldsymbol{g} \subset \boldsymbol{d}$ collects those exogenous inputs the optimizer is allowed to change. A variety of methods are applied to carry out this optimization, which can be broadly classified into gradient-based and gradient-free (derivative-free) methods [2], [3]. One of the main advantages of the latter is that they do not require evaluating derivatives, whereas gradient methods do. Metaheuristic methods, in general, can be classified into evolutionary algorithms, swarm intelligence-based algorithms, physics-based algorithms, human-related algorithms, and hybrid metaheuristic algorithms [4]. Evolutionary algorithms, and specifically genetic algorithms (GAs), have proven to be particularly suitable for solving certain engineering problems [5], which has made them an active line of research in the field of optimization.

GAs emulate the principles of natural selection. They were introduced in 1970 by John Holland [6]. Their exploration of the solution space begins with a population of candidate solutions, which are ranked based on their suitability according to the objective function. Successive generations of solutions are then produced, each of which is more fit in terms of its ability to optimize the objective function. Less fit solutions are disfavored (they have a lower probability of contributing to future generations). The process concludes when certain criteria suggest that no significant improvement is being made between generations [7]. This algorithmic structure allows GAs to be versatile and robust, which is why they have been used to solve a variety of engineering problems, such as design problems [8], reactor synthesis and process optimization [9], bioprocess optimization [10], supply chain networks [11], among others. However, the versatility of GAs comes with a computational cost. Therefore, algorithms have been developed to reduce this cost, such as the use of low-fidelity models [12], coupled optimizers [13], dynamic mutation rates [14], or more general concepts like hybrid models [15].

On the other hand, GAs are particularly suitable for problems with many nonlinear degrees of freedom. For example, the study of heat exchanger networks [16] or the synthesis and optimization of chemical reactors [17]. Also, the design of materials [18] or bioprocesses, where they have been used to optimize fermentation processes [19], [20] and enhance the production of antimicrobial peptides and other bioproducts by optimizing cultivation conditions and selecting microbial strains [21], among other uses. A case of particular interest is model-assisted biofuel production [22], which illustrates the value of model-based approaches. These frameworks exploit mechanistic (first-principles) or hybrid dynamic models as oracles to evaluate objectives and constraints, typically through time-domain simulations. Although GAs are not included in this paper [22] its approach requires algorithms for solving the models presented, and given the nature of these models, GAs appear promising.

For the proper application of GAs through model-based approaches, it is necessary to develop these models; in this case, GAs can be used to optimize kinetic models, which in turn provide

optimal feeding strategies, set-points, or dynamic profiles. This has been applied in various fields, such as the chemical [23], [24] or biomedical industries [25]. In the field of bioprocesses, kinetic models (together with hydrodynamic models) are becoming increasingly important due to their global growth, which is also driving an increase in their scale [26].

The aim of this article is to examine the application of GAs in the optimization of bioprocesses, with a specific focus on a particular bioprocess model as a case study, capitalizing on the ability of GAs to handle problems with multiple degrees of freedom and complex nonlinearities. This approach seeks to demonstrate the effectiveness and versatility of GAs in the bioprocess field, providing a solid foundation for their implementation in industrial applications and their potential to tackle current challenges in large-scale bioproduct production. Furthermore, the behavior of genetic algorithms in bioprocess is analyzed and compared to a model function, such as the Ackley function, to assess whether there are significant differences in their behavior and to establish a framework for applying genetic algorithms to these types of systems. We hypothesize that hyperparameter tuning via a primary GA (crossover fraction and population size) can outperform a direct GA. This hypothesis is first tested on a two-dimensional benchmark with known properties (Ackley) and then on the bioprocess model, thereby isolating surface ruggedness effects from system dimensionality.

## II. CASE OF STUDY

As a case study, a model for xylose fermentation aimed at producing xylitol will be used [27]. This model was modified to allow for semi-continuous operation and the estimation of gas retention [28], among other adaptations not present in the original study. The model $\boldsymbol{dx/dt = f(d; x; p)}$ is presented in (1) to (6). Additionally, (7) to (14) present the algebraic equations required to complete the vector $\boldsymbol{d} = (U_g, vvm, v_g, K_L a, \delta, \mu, \epsilon)^T$. In this case, the state vector is defined as $\boldsymbol{x}(t)=(V(t), C_X(t), C_S(t), C_P(t), C_{O2}(t), C_{gO2}(t))^T$.

$$\frac{dV}{dt} = F \tag{1}$$

$$\frac{dC_X}{dt} = \mu C_X - \frac{F}{V}C_X \tag{2}$$

$$\frac{dC_S}{dt} = \frac{F}{V}\left(C_{Sf} - C_S\right) - \frac{1}{Y_{XS}}\frac{dC_X}{dt} - \frac{1}{Y_{PS}}P \tag{3}$$

$$\frac{dC_P}{dt} = P + \delta - \frac{F}{V}C_P \tag{4}$$

$$\frac{dC_{O2}}{dt} = K_L a\left(K_{eq}C_{g,O2} - C_{O2}\right) - Y_{O2}C_X\left(\frac{C_{O2}}{K_{O2}+C_{O2}}\right) + \frac{F}{V}\left(C_{O2f} - C_{O2}\right) \tag{5}$$

$$\frac{dC_{gO2}}{dt} = \frac{\upsilon_g}{\epsilon V}\left(C_{g,O2in} - C_{g,O2}\right) - K_L a\left(K_{eq}C_{g,O2} - C_{O2}\right) \tag{6}$$

$$U_g = \frac{Vg}{At} \tag{7}$$

$$vvm = \frac{Vg}{1000V} \tag{8}$$

$$\upsilon_g = 60 \times V \times vvm \tag{9}$$

$$K_L a = 5724 U_g^{0.93}\left(\frac{N}{N_{CD}}\right)^{1.342}\left(\frac{T}{D}\right)^{0.415} \tag{10}$$

$$\delta = \delta_0 C_X \frac{K_\delta}{K_\delta + C_S} \tag{11}$$

$$P = AC_X\left(\frac{C_S}{C_S + K_{li,P}}\right)\left(\frac{K_{in,P}^2}{K_{in,P}^2 + C_S^2}\right)\left[1 - \left(\frac{C_{O2}}{K_{in,O2}}\right)^2\right] \tag{12}$$

$$\mu = \mu_{max}\left(\frac{C_S}{C_S + K_{li,S}}\right)\left(\frac{K_{in,S}}{K_{in,S} + C_S}\right)\left(\frac{C_{O2}}{C_{O2} + K_{li,O2}}\right) \tag{13}$$

$$\epsilon = 0.104\left(\frac{N}{N_{CD}}\right)^{0.62} vvm^{0.64} \tag{14}$$

Fig. 1 shows a schematic representation of the bioreactor operating in semicontinuous mode. Initially, the bioreactor starts with $\boldsymbol{x_0}=(V_0, C_{X0}, C_{S0}, C_{P0}, C_{O_20}, C_{gO_20})^T$. During operation, a feed stream $F$ of substrate with concentration $C_{Sf}$ is supplied.

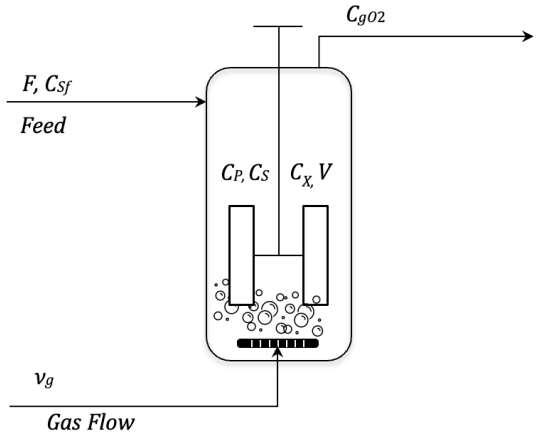


Fig. 1. Schematic diagram of the bioreactor under semi-continuous operation. In batch mode, $F$=0.

## III. METHODOLOGY

We cast the study as a bilevel optimization. The primary (outer) problem tunes the genetic algorithm's hyperparameters (crossover fraction $cf \in (0,1)$ and population size $S \in \mathbb{N}$) to improve the performance attained by a secondary (inner) GA that solves the domain problem.

First, the Ackley function (benchmark, inner problem) is given in (15).

$$\min_{z \in \mathbb{R}^2} f_{Ackley}(z),\ with\ z = (x, y),\ f_{Ackley}\ as\ in\ (28) \tag{15}$$

Second, the bioprocess (inner problem) is presented in (16).

$$\min_{\mathbf{D} \in \Omega_{\mathbf{D}}} \boldsymbol{OF}(\mathbf{D}) \text{ s. t. } \dot{\mathbf{x}} = f(t, \mathbf{x}; p, \mathbf{d}),\ \mathbf{x}(t_0) = \mathbf{x}_0(\mathbf{D}) \tag{16}$$

In (16) $\mathbf{D} = [t_{\text{final}}, C_{X0}, S_0, C_{Sf}, F]^{\mathrm{T}}$ are the decision variables and **d** collects exogenous inputs and algebraic quantities, while ***OF*** is defined in (19).

Third, the primary (outer) problem, in both cases, is formulated in (17).

$$\min_{cf,S} \Phi(cf, S) = J\left((inner\ problem; cf, S)\right) \tag{17}$$

In (17) GA(·;*cf*,*S*)denotes the result returned by a GA configured with crossover fraction *cf* and population size *S*, and *J*(·)is the performance metric (for Ackley, the attained value of $f_{\text{Ackley}}$, for the bioprocess, the attained *OF*). Fig. 2 provides a schematic overview of the methodology described above.

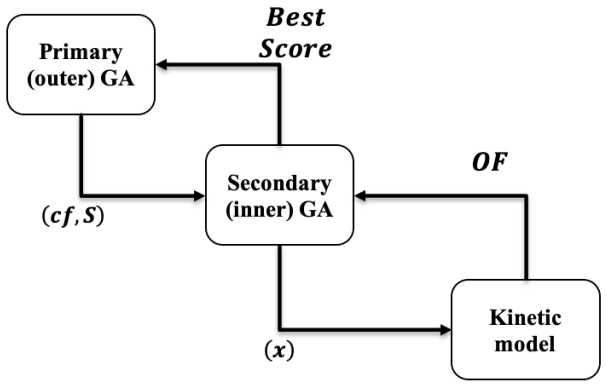


Fig. 2. Bilevel GA framework: primary GA optimizing a secondary GA coupled with a kinetic model.

The optimization is carried out using the *ga* function from MATLAB [29], which is given in (18).

$$[\boldsymbol{A}] = ga[ff, \boldsymbol{B}] \tag{18}$$

In (18) ***A*** is the result matrix returned by the function, *ff* is the fitness function, and ***B*** is the input matrix specifying the required details for ga to operate in the desired manner. For all the cases studied, the mutation function used is @mutationadaptfeasible and the default function is applied for crossover. Although implemented in MATLAB [29], the optimization algorithms are presented as a language-independent hierarchical GA framework.

As the objective function for the kinetic model (*OF*), the following is used in this analysis:

$$OF = \frac{RC_pV - \left(S_0V_0 + C_{sf}Ft_f\right)Y_{PS}}{tf} \tag{19}$$

where

$$R = \frac{\$/kg\ producto}{\$/kg\ sustrato} = 2$$

This objective function compares product equivalents adjusted by price with product equivalents adjusted by yield. Additionally, it is divided by the reaction time to convert it into a production rate.

To solve the model ***dx/dt*** = *f(t,* ***x****;* ***p****,* ***d****)*, an explicit RKF45 integrator was implemented to enforce a uniform local-error tolerance across all experiments, ensuring fair wall-time comparisons and language-independent reproducibility. Unlike black-box solvers (e.g., MATLAB's ode45), the step-accept/reject policy and the step-size update are fully specified, which prevents version-dependent behavior and facilitates re-use in Python or R with identical numerical policies. The RKF45 method is based on the following equations to calculate the slopes $k_i$: (20) to (25), the fourth-order ($x_4$) and fifth-order ($x_5$) solutions: (26) and (27), and the error ***e***:(28)[30].

$$k_1 = hf\left(t_n, x_n\right) \tag{20}$$

$$k_2 = hf\left(t_n + \frac{1}{4}h, x_n + \frac{1}{4}k_1\right) \tag{21}$$

$$k_3 = hf\left(t_n + \frac{3}{8}h, x_n + \frac{3}{32}k_1 + \frac{9}{32}k_2\right) \tag{22}$$

$$k_4 = hf\left(t_n + \frac{12}{13}h, x_n + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right) \tag{23}$$

$$k_5 = hf\left(t_n + h, x_n + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right) \tag{24}$$

$$k_6 = hf\left(t_n + \frac{1}{2}h, x_n - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\right) \tag{25}$$

$$x_4 = x_n + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5 \tag{26}$$

$$x_5 = x_n + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6 \tag{27}$$

$$e = \|x_5 - x_4\|_\infty \tag{28}$$

The RKF45 method calculation algorithm is described in Algorithm 1.

**Algorithm 1**. Implementation Structure of the RKF45 Method [30]
**Input:** $f, x_0, t_0, tfinal, tol, hinitial$
**Output:** $T, X$
Initialize $(t, x, h) \leftarrow (t_0, x_{0(i)}, h_{initial})$
Initialize $T \leftarrow [t]$, $X \leftarrow [x^T]$
**while** $t < t_{final}$ **do**
  Compute $k_1, k_2, k_3, k_4, k_5, k_6$ (equations (20) a (25))
  Compute $x_4$ y $x_5$ (equations (26) y (27))
  Estimate the error $e = \|x_5 - x_4\|_\infty$ (equation (28))
  **if** $e < tol$ **then**
    Acept the step: $t \leftarrow t + h$, $x \leftarrow x_5$
    Store the results: $T \leftarrow [T;t]$, $X \leftarrow [X;xT]$
    Adjust the step: $h \leftarrow h \cdot \min(4,\max(0.1,0.8(tol/e)^{1/2}))$
  **else**
  Reduce the step: $h \leftarrow h \cdot \max(0.1,0.8(tol/e)^{1/4})$
  **end if**
  **if** $t + h > t_{final}$ **then**
    $h = t_{final} - t$
  **end if**
**end while**
RETURN $T, X$

The implementation of the GA is detailed in Algorithm 2. In this algorithm, the kinetic model is housed in a vector of anonymous functions, and the functions are processed in wrappers to facilitate the execution flow. Additionally, Algorithm 2 outlines how to optimize the crossover fraction and population size by using another genetic algorithm, also through the ga function.

**Algorithm 2.** Computational Structure for Optimization
**Input:**
  ***params*** $\leftarrow [A, delta_0, K_{delta}, K_{O2}, K_{eq}, K_{ind}, K_{inS}, K_{inP}, K_{inO2}, K_{lid}, K_{liP}, K_{liS}, K_{liO2}, N_{CD}, Y_{O2}, Y_{PS}, Y_{XS}, u_{max}, V_0, F, C_{Sf}, TD, N, C_{O2f}, S_0, V_g, T, A_t, D, C_{gO2in}]$ `//Constants required by the kinetic model`
  $\boldsymbol{x_0} \leftarrow [V_0, C_{x0}, C_{s0}, C_{p0}, C_{O20}, C_{gO20}]$ `//Initial conditions`
  $t_0, t_{final}, tol, h\_initial$
**Output:**
  ***xx*** [*CrosoverFraction, Population*]
  $\boldsymbol{x}_{optimal}$

`//Kinetic model of the system defined using anonymous functions`
$\boldsymbol{x} \leftarrow [V, C_x, C_s, C_p, C_{gO2}]$ `//State variables`
dxdt1 $\leftarrow$ @(*t, x, x_algebraic*) *F*;
dxdt2 $\leftarrow$ @(*t, x, x_algebraic*) *x_algebraic(7)*x(2) + F/x(1)*x(2)*;
dxdt3 $\leftarrow$ @(*t, x, x_algebraic*) *F/x(1)*(CSf - x(3)) - (1/YXS)*dxdt2(t, x, x_algebraic)…*
  *-(1/YPS)*x_algebraic(6)*;
dxdt4 $\leftarrow$ @(*t, x, x_algebraic*) *x_algebraic(6) + x_algebraic(5) - F/x(1)*x(4)*;
dxdt5 $\leftarrow$ @(*t, x, x_algebraic*) *x_algebraic(4)*(Keq*x(6) - x(5)) - YO2*x(2)*…*
  *(x(5)/(KO2+x(5)) +…F/x(1)*(CO2f-x(4)))*;
dxdt6 $\leftarrow$ @(*t, x, x_algebraic*) *x_algebraic(3)/(x_algebraic(8)*x(1))*(CgO2in - x(6)) -*
  *x_algebraic(4)*(Keq*x(6) - x(5))*;
**dxdt** $\leftarrow$ [*dxdt1, dxdt2, dxdt3, dxdt4, dxdt5, dxdt6*] `// System of equations vector`

***for*** `each` (*P,cf*):
  a. Configure **`Inner GA`** with population *P* and crossover fraction *cf*.
  b. Inner GA minimizes *OF*(*x*) by repeatedly:
  — proposing *x*, integrating *f* with RKF45 at fixed tolerance

```
        – evaluating FO from terminal values using equation (16).
        c. Record the best inner-GA score.
    Selection/Recombination/Mutation on (P, cf) until outer-GA termination.
    RETURN the best ((P, cf) and the corresponding x from the inner GA.

    FUNCTION [FO] RKF45_SystemL(x, fdxdt, fx0, ft0, ftfinal, ftol, fh_initial, fp0)
        x_algebraic = AlgebraicEquations(x, p);
        for i=1 to number of state variables
                Execute Algorithm 1 for each equation of the model
        end
        FO = -(2*X(length(X),4)*X(length(X),1)-(params(3)*x(1)…
        +params(4)*params(5)*tfinal)*p0(16))/tfinal; // Use the tfinal values to compute the
objective function (FO)
        RETURN FO
    END FUNCTION
    FUNCTION Algebraic_equations (x, params)
        eq ⇠ [eqi, eqi+1… eqi+N]
        eq ⇠ @(x, params) feq // Algebraic equations
        RETURN x_algebraic
    END FUNCTION
```

To carry out comparative optimization tests, the two-dimensional Ackley function is used, as given in (29), whose graph for a range from -5 to +5 on both coordinates is shown in Fig. 3. This function is used because it has a large number of local minima and a known global minimum of 0 at the coordinates (0, 0).

$$f(x,y) = 20 + e - 20e^{0.2\cdot\sqrt{0.5\cdot\Sigma\left(x^2+y^2\right)}} - e^{0.5(cos(2\pi x)+cos(2\pi y))} \quad (29)$$

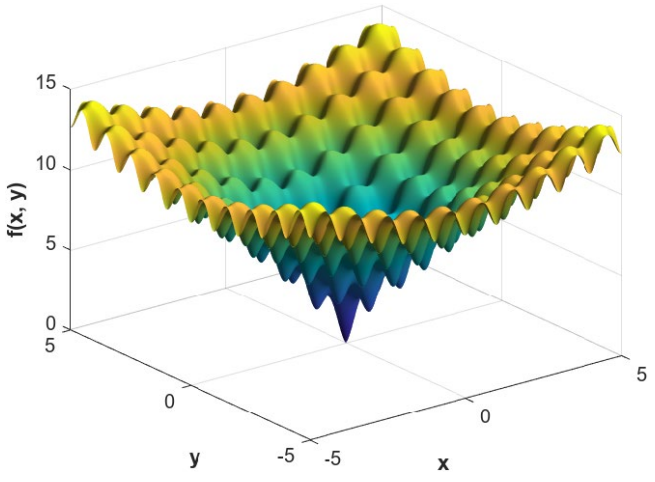


Fig. 3. Visualization of the Ackley function used to compare with the kinetic model on the performance of the genetic algorithm.

## IV. RESULTS

First, the proper behavior of the model is verified, as presented in Fig. 4. This behavior is characteristic of kinetic models for bioprocesses, as it shows a gradual reduction in substrate while biomass increases, followed by the product formation.

In this case study, the state variables are $\boldsymbol{x}=(V, C_X, C_S, C_P, C_{O_2}, C_{gO_2})$. The inputs/decision variables explored by the GA include inoculum ($C_{X0}$), initial substrate ($C_{S0}$), substrate feed ($C_{Sf}$), flow rate ($F$), agitation ($N$), and reaction time ($t$).

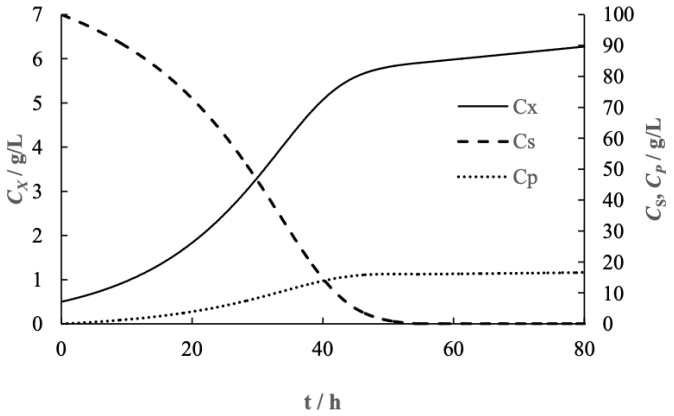


Fig. 4. Behavior of the kinetic model. The continued increase in biomass concentration even after the substrate has been depleted is a consequence of the modified mass balance that includes biomass feed.

Furthermore, Fig. 5 presents the behavior of the mean objective function (*OF*) value for the kinetic model, as well as the Ackley function value after optimization. It is important to note that the GA performs minimization; therefore, the *OF* is equal to the negative of the mean fitness. Additionally, the global optimum of the Ackley function is zero, meaning that the objective function in this case is directly the Ackley function. For the kinetic model, error bars are also shown, representing the standard deviation of the population of individuals for the *OF*. The simulations were performed using crossover fraction values within the range: $0 \leq$ *crossover fraction* $\leq 1$.

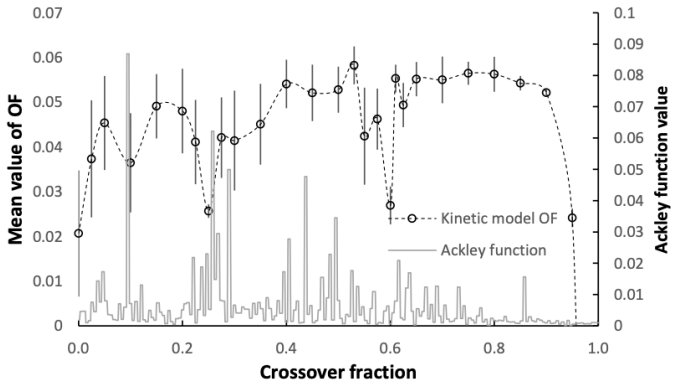


Fig. 5. Behavior of the mean fitness value for the kinetic model and the Ackley function as the crossover fraction varies, while keeping the population constant at 200. Dashed lines indicate the interpolated values of sampled crossover fractions for visualization purposes only.

In addition, Fig. 6 illustrates the behavior of the best fitness value across generations for a population size of 200. Each data series corresponds to a crossover fraction within the range of 0 to 1. The data series in group A represent crossover fractions between 0.6 and 0.9. Group B consists of crossover fractions less than or equal to 0.6, while group C includes crossover fractions greater than 0.9. These results align with expected behavior. In group C, convergence occurs rapidly since each generation is largely composed of the best individuals from the previous one. However, this leads to the disadvantage of reaching local optima. In group B, convergence is also fast, but the optima obtained appear to be closer to the global optimum, making this the most suitable working range. Finally, in group A, mutation plays a dominant role, resulting in slower convergence, but also leading to optima that seem to approach the global solution.

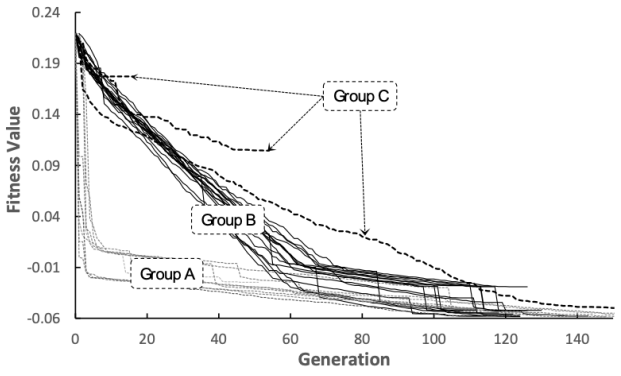


Fig. 6. Behavior of the best fitness value across generations for a population size of 200, using crossover fraction (cf) values between 0 and 1. The data series correspond approximately to the following ranges: A: **0.6≤cf≤0.9;** B: **0≤cf≤0.6;** C: **0.9≤cf≤1.**

On the other hand, Fig. 7 illustrates the standard deviation of the objective function final populations(scores) for each crossover fraction value. As expected, the standard deviation, which represents the average difference between individuals within the same generation, decreases as the crossover fraction increases. This behavior is explained by the fact that lower crossover rates allow for higher mutation rates, leading to greater diversity within the population.

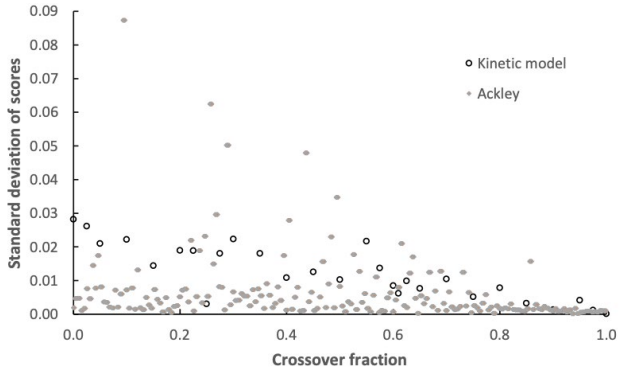


Fig. 7. Standard deviation of scores (-*OF* value) for the population of the last generation as a function of the crossover fraction.

## V. DISCUSSION

In this study, the optimization of the kinetic model for the xylose fermentation process aimed at xylitol production was analyzed in detail. The results demonstrate that the convergence of the genetic algorithm (GA) is critically dependent on the crossover fraction. It was observed that, within the range ($0.6 \leq cf \leq 0.9$), an adequate balance is achieved between exploration of the search space and exploitation of the most promising solutions. The standard deviation of the population (Fig. 5) in this range remains relatively constant, decreases at higher crossover fractions due to rapid convergence, and increases at lower crossover fractions as a result of the effect of mutation. This behavior is consistent with the literature [5], [9], in which it has been reported that moderate crossover rates tend to prevent premature convergence to local optima. This interval of balanced convergence is known as the stability valley.

On the other hand, crossover fraction values within the intervals $0 \leq cf \leq 0.6$ and $0.9 \leq cf \leq 1.0$ (as shown in the populations of Fig. 4) exhibited faster convergence, although with a greater susceptibility to becoming trapped in local optima. When these findings were compared with the behavior of the Ackley function (Fig. 3 and Fig. 5), the robustness of GAs in addressing problems with highly complex solution landscapes was confirmed, an aspect particularly relevant in bioprocesses, where models are typically intricate [4], [7]. Moreover, although the comparison with the Ackley function was restricted to two dimensions, the convergence pattern closely resembled that of the kinetic model. This suggests that the irregular topography of the search space, rather than dimensionality alone, governs the GA's performance. Therefore, it is proposed that the conclusions drawn from this study may be transferable to other bioprocesses. However, this assumption warrants further investigation.

In order to further investigate the GA's sensitivity to its crossover fraction values, a primary GA was employed to optimize the parameters (population size and crossover fraction) of the secondary GA (responsible for the direct resolution of the kinetic model). This strategy identified a crossover fraction configuration of approximately 0.53 outside the stability valley, which increased the objective function by about 3.1 % compared to the best value observed within the valley. This improvement was achieved without introducing any modifications to the kinetic model or altering the economic criterion that penalizes substrate consumption and rewards xylitol production; therefore, the increase is entirely attributable to the tuning of the evolutionary heuristic. This demonstrates that the systematic exploration of the secondary model's metaparameter space can surpass usual empirical recommendations and yield tangible economic benefits.

The benefits, however, come at a cost. Let $P_{in}$,$G_{in}$ be the inner GA population and number of generations, and $P_{out}$,$G_{out}$ their outer counterparts. Let $C_{eval}$ denote the cost of a single fitness evaluation, dominated here by integrating the ODE system with RKF45 at a constant error tolerance. If $N_{steps}$ is the (adaptive) number of steps and $cf$ the cost of one right-hand-side evaluation, then $C_{eval}=O(N_{steps}\cdot cf)$. The direct (single-level) cost scales as $O(P_{in}G_{in}C_{eval})$. The bilevel cost scales as $O(P_{out}G_{out}(P_{in}G_{in}C_{eval}))$, which explains the multiple-fold increase observed in practice and motivates hybrid strategies (e.g., variable-fidelity models [12] or adaptive mutations [14]) to reduce the number of objective evaluations in large systems. Therefore, the acceptability of this additional cost must be carefully assessed.

Regarding the optimum found, whose simulation is shown in Fig. 8, it is notable that the value of $S_0$ was set by the optimization at its lower bound, while tfinal reached its upper bound. This outcome can be explained by the fact that the objective function prioritizes efficient substrate utilization; consequently, the optimization tends to minimize the initial substrate and supply substrate exclusively through feeding.

In other words, the optimal policy drives $S_0$ to its lower bound and extends $t_{final}$, consistent with (19), penalizing residual substrate: feed-only strategies minimize wastage while enabling product formation via $P$ and $\delta$ ((11) and (12)). Productivity is governed less by a large initial charge and more by sustained, moderate substrate availability. Because the objective penalizes residual substrate, the combination of low $S_0$ with controlled feeding minimizes wastage and aligns with the model's production structure (growth-associated plus non-growth-associated terms). This yields an early growth-dominant phase followed by a substrate-limited phase in which $dC_P/dt$ declines as $C_S$ approaches depletion. Under the biomass-feed modification of the mass balance, $C_X$ can continue to increase despite $C_S \simeq 0$ because the positive inflow term offsets dilution, clarifying the trajectories observed.

These patterns imply practical guidelines: schedule feed trajectories to keep $C_S$ within the productive window, tune ($F$,$C_{Sf}$) to balance dilution against reaction rates, and choose tfinal at the point of diminishing marginal product (slope of $C_P$). Alternative objective functions could be used to evaluate other scenarios, in which additional factors are considered or specific goals are pursued.

Future research could focus on the application of GAs to more complex bioprocess models, including variability in operating conditions and the integration of experimental data. Additionally, the implementation of hybrid approaches, combining GAs with other optimization methods, could be explored, as this may provide further benefits in terms of efficiency and accuracy. This is particularly relevant given that one of the main limitations of using a primary GA to optimize the parameters of the secondary optimization is its high computational cost. Finally, the exploration of machine learning techniques to dynamically adjust GA parameters during optimization could represent an interesting research avenue to further improve the performance of genetic algorithms in bioprocess optimization.

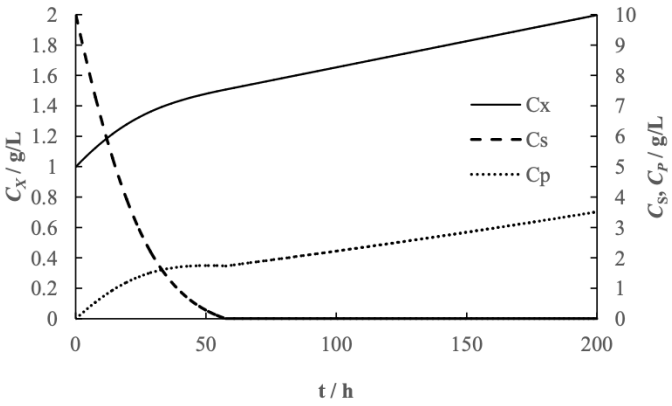


Fig. 8. Time evolution of substrate, biomass, and product concentrations under optimal genetic algorithm parameters for xylose fermentation ($t_{final}$=**200**, $C_{x0}$=**1**, $S_0$=**10**, $C_{Sf}$=**0.005**, $F$ = **0.0011**, $N$=**164**). The observed discontinuity arises due to the feeding process when the substrate is depleted.

## VI. CONCLUSION

This study has demonstrated the effectiveness of genetic algorithms (GAs) in the optimization of bioprocesses, using as a case study a xylose fermentation model for xylitol production. By applying a primary genetic algorithm to fine-tune the metaparameters of the secondary algorithm, a 3.1 % increase in the objective function ($OF$) value was achieved compared to the best value obtained within the stability valley ($0.6 \leq cf \leq 0.9$), with an optimal crossover fraction of 0.53, outside this range. The results indicate that GAs can effectively handle complex and nonlinear systems, offering a robust tool to enhance the performance of bioprocesses.

From a practical perspective, the findings of this study have important implications for the biotechnology industry. The ability of GAs to optimize complex bioprocesses could translate into improvements in the efficiency and profitability of bioproduct manufacturing. Theoretically, this study contributes to the understanding of how GAs can be configured to maximize

their performance in the optimization of nonlinear dynamic systems. The implementation of the RKF45 method for solving the kinetic model ensures unbiased comparisons between different algorithm configurations, as a constant error is maintained. The research presented here provides a solid foundation for future applications of GAs in other bioprocesses and in process engineering in general. Hybrid approaches and the integration of machine learning techniques are promising areas for further enhancing GA performance.

The simulations indicated that reducing the initial substrate and extending the process time maximize conversion, a result consistent with the penalty imposed on residual substrate in the objective function. In this way, critical variables were identified, such as the feed rate, the initial substrate concentration, and the fermentation duration—factors that must be considered for scale-up to industrial applications. Furthermore, the comparison with the Ackley function revealed similar convergence trajectories despite the lower dimensionality of that benchmark, suggesting that the ruggedness of the search surface, rather than model dimensionality, governs GA performance. This finding supports the extrapolation of these guidelines to other bioprocesses with complex topographies.

In summary, this study has demonstrated the effectiveness of GAs in the optimization of bioprocesses, highlighting their potential to improve efficiency and profitability in bioproduct manufacturing. The results obtained provide a solid foundation for future research and applications in the biotechnology industry and in process engineering.

## ROLE OF THE AUTHORS

The author Leonardo Garro Mena is responsible for the entire document.

## REFERENCES

[1] T. A.-N. Nguyen y T.-A. Nguyen, "Genetic Algorithms for Chemical Engineering Optimization Problems," in Genetic Algorithms, S. Ventura, J.M. Luna, and J.M. Moyano, Eds. London, United Kingdom: IntechOpen, 2022, doi: 10.5772/intechopen.104884.

[2] G. Venter, "Review of Optimization Techniques," in Encyclopedia of Aerospace Engineering, R. Blockley and W. Shyy, Eds. Chichester, United Kingdom: John Wiley & Sons, Ltd., 2010, doi: 10.1002/9780470686652.eae495.

[3] N. Brahimi, A. Dolgui, E. Gurevsky, and A. R. Yelles-Chaouche, "A literature review of optimization problems for reconfigurable manufacturing systems," IFAC-PapersOnLine, vol. 52, no. 13, pp. 433-438, Jan. 2019, doi: 10.1016/j.ifacol.2019.11.097.

[4] V. Tomar, M. Bansal, and P. Singh, "Metaheuristic Algorithms for Optimization: A Brief Review," Eng. Proc., vol. 59, no. 1, Art. no. 238, 2024, doi: 10.3390/engproc2023059238.

[5] A. S. Ramadan and E. O. Elgendi, "A review of optimization techniques and algorithms used for FRP applications in civil engineering," J. Eng. and Appl. Sci., vol. 70, no. 1, Art. no. 61, Jun. 2023, doi: 10.1186/s44147-023-00209-5.

[6] John H. Holland, Adaptation in Natural and Artificial Systems. Cambridge, MA, USA: The MIT Press, 1992. [Online]. Available: https://mitpress.mit.edu/9780262581110/adaptation-in-natural-and-artificial-systems/

[7] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," Multimedia tools and applications, vol. 80, no. 5, pp. 8091-8126, 2021, doi: 10.1007/s11042-020-10139-6.

[8] M. A. El-Shorbagy and A. M. El-Refaey, "A hybrid genetic–firefly algorithm for engineering design problems," J. Comput. Des. and Eng., vol. 9, n.o 2, pp. 706-730, Apr. 2022, doi: 10.1093/jcde/qwac013.

[9] S. Zhang, Z. Ge, and Y. Lai, "Application of Genetic Algorithm in Optimizing a Chemical Adsorption Bed with Cacl2/expanded Graphite Adsorbent," Procedia Eng., vol. 205, pp. 1828-1834, Jan. 2017, doi: 10.1016/j.proeng.2017.10.244.

[10] M. Rocha, I. Rocha, and E. Ferreira, "A new representation in evolutionary algorithms for the optimization of bioprocesses," 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2005, pp. 484-490 Vol. 1. doi: 10.1109/CEC.2005.1554722.

[11] J. O. Robles, C. Azzaro-Pantel, and A. Aguilar-Lasserre, "Optimization of a hydrogen supply chain network design under demand uncertainty by multi-objective genetic algorithms," Comput. & Chem. Eng., vol. 140, p. 106853, Sep. 2020, doi: 10.1016/j.compchemeng.2020.106853.

[12] L. Shu, P. Jiang, Q. Zhou, X. Shao, J. Hu, and X. Meng, "An on-line variable fidelity metamodel assisted Multi-objective Genetic Algorithm for engineering design optimization," Appl. Soft Comput., vol. 66, pp. 438-448, May 2018, doi: 10.1016/J.ASOC.2018.02.033.

[13] M. Gobbi, "A k, k-ε optimality selection based multi objective genetic algorithm with applications to vehicle engineering," Optim. Eng., vol. 14, no. 2, pp. 345-360, Jun. 2013, doi: 10.1007/s11081-011-9185-8.

[14] M. S. Krejca and C. Witt, "A Flexible Evolutionary Algorithm With Dynamic Mutation Rate Archive," arXiv.org, Apr. 2024, doi: 10.1145/3638529.3654076.

[15] X. Yan, H. Liu, Z. Zhu, and Q. Wu, "Hybrid genetic algorithm for engineering design problems," Cluster Comput., vol. 20, no. 1, pp. 263-275, Mar. 2017, doi: 10.1007/s10586-016-0680-8.

[16] M. C. Aguitoni, L. V. Pavão, P. H. Siqueira, L. Jiménez, and M. A. da Silva Sá Ravagnani, "Heat exchanger network syn-

thesis using genetic algorithm and differential evolution," Comput. & Chem. Eng., vol. 117, pp. 82-96, Sep. 2018, doi: 10.1016/j.compchemeng.2018.06.005.

[17] F. Sun, W. Du, R. Qi, F. Qian, and W. Zhong, "A Hybrid Improved Genetic Algorithm and Its Application in Dynamic Optimization Problems of Chemical Processes", Chin. J. Chem. Eng., vol. 21, no. 2, pp. 144-154, Feb. 2013, doi: 10.1016/S1004-9541(13)60452-8.

[18] S. K. Gupta and M. Ramteke, "Applications of Genetic Algorithms in Chemical Engineering II: Case Studies," in Applications of Metaheuristics in Process Engineering, J. Valadi and P. Siarry, Eds. Switzerland: Springer International Publishing, 2014, pp. 61-87. doi: 10.1007/978-3-319-06508-3_3.

[19] J. S. Tumuluru and R. McCulloch, "Application of Hybrid Genetic Algorithm Routine in Optimizing Food and Bioengineering Processes," Foods, vol. 5, no. 4, Art. no. 4, Dic. 2016, doi: 10.3390/foods5040076.

[20] M. Rocha, R. Mendes, O. Rocha, I. Rocha, and E. C. Ferreira, "Optimization of fed-batch fermentation processes with bio-inspired algorithms," Expert Syst. Appl., vol. 41, no. 5, pp. 2186-2195, Apr. 2014, doi: 10.1016/j.eswa.2013.09.017.

[21] I. J. Peerzade, S. Mutturi, and P. M. Halami, "Improved production of RNA-inhibiting antimicrobial peptide by Bacillus licheniformis MCC 2514 facilitated by a genetic algorithm optimized medium," Bioprocess Biosyst. Eng., vol. 47, no. 5, pp. 683-695, May 2024, doi: 10.1007/s00449-024-02998-2.

[22] H. Narayanan et al., "Bioprocessing in the Digital Age: The Role of Process Models," Biotechnology J., vol. 15, no. 1, Art. no. 1900172, Jan. 2020, doi: 10.1002/biot.201900172.

[23] D. A. DelVescovo, J. Li, D. A. Splitter, F. D. F. Chuahy, and P. Zhao, "Genetic algorithm optimization of a chemical kinetic mechanism for propane at engine relevant conditions," Fuel, vol. 338, Art. no. 127371, Apr. 2023, doi: 10.1016/j.fuel.2022.127371.

[24] A. Lapene, G. Debenest, M. Quintard, L. M. Castanier, M. G. Gerritsen, and A. R. Kovscek, "Kinetics Oxidation of Heavy Oil. 2. Application of Genetic Algorithm for Evaluation of Kinetic Parameters," Energy Fuels, vol. 29, no. 2, pp. 1119-1129, Feb. 2015, doi: 10.1021/ef501392k.

[25] Y. Wang, J. Luan, K. Luo, J. Fan, and T. Zhu, "Model reduction of coagulation cascade based on genetic algorithm," Int. J. Num. Methods Biomed. Eng., vol. 38, no. 11, Art. no. 3652, Nov. 2022, doi: 10.1002/cnm.3652.

[26] V. K. Singh, I. Jiménez del Val, J. Glassey, and F. Kavousi, "Integration Approaches to Model Bioreactor Hydrodynamics and Cellular Kinetics for Advancing Bioprocess Optimisation," Bioeng., vol. 11, no. 6, Art. no. 6, Jun. 2024, doi: 10.3390/bioengineering11060546.

[27] N. L. Mohamad, S. M. Mustapa Kamal, M. N. Mokhtar, S. A. Husain, and N. Abdullah, "Dynamic mathematical modelling of reaction kinetics for xylitol fermentation using Candida tropicalis," Biochem. Eng. J., vol. 111, pp. 10-17, May 2016, doi: 10.1016/j.bej.2016.02.017.

[28] A. A. Yawalkar, V. G. Pangarkar, and A. A. C. M. Beenackers, "Gas hold-up in stirred tank reactors," Can. J. Chem. Eng., vol. 80, no. 1, pp. 158-166, 2002, doi: 10.1002/cjce.5450800117.

[29] MATLAB. (R2024b). The MathWorks, Inc.

[30] R. L. A. Burden, Análisis numérico, 7th. ed. Mexico City, Mexico: Thomson Learning, 2002.