ARTIFICIAL NEURAL NETWORK-GENETIC AND PSO ALGORITHM FOR OPTIMIZATION OF MULTIVARIATE FUNCTION: AN APPLICATION TO LACTIC ACID PRODUCTION

Asim Mahat[1], Unique Karki[1], Utsav Darlami[1], Samir Shrestha[1]

Abstract

In this study, an Artificial Neural Network (ANN) was developed to model the relationship between factors affecting a process and its output. The Genetic Algorithm(GA) and Particle Swarm Optimization(PSO) were used as a meta-heuristic approach to find the optimal values of the factors that maximize the output, where the ANN was used as the fitness function for the Genetic Algorithm. A known multivariate function was constructed to validate the ANN model, and the GA/PSO algorithm was applied to estimate its optimal value. Finally, the proposed approach was applied to optimize the experimental conditions for the production of lactic acid.

1. Introduction

Lactic acid is a versatile organic acid with numerous applications, including its use as a food preservative, flavor enhancer, solvent, cleaning agent, and antioxidant. Traditionally, lactic acid has been produced in the lab using a composition of multiple chemicals through the Response Surface Method (RSM). However, one major limitation of RSM is the requirement for a large number of experiments. In this research paper, we propose an alternative approach for producing lactic acid efficiently and effectively. We utilize a dataset collected from the lab and train a neural network to predict the optimal set of values for the production of lactic acid. To further optimize the production process, we employ Genetics and Particle Swarm optimization algorithms. Our goal is to identify a practical method that can be implemented in the industry to produce lactic acid with high efficiency and quality.

2. Literature Review

Artificial neural networks have been widely used for process modeling and optimization in various industries [1]. In the field of bioengineering, neural networks have been utilized for the optimization of bioprocesses such as the production of ethanol [2], citric acid [3], and scleroglucan [4]. The optimization of bioprocesses using neural networks has shown improved efficiency and cost-effectiveness compared to traditional methods such as Response Surface Methodology [5]. Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are meta-heuristic algorithms that have been used in combination with neural networks for the optimization of various industrial processes [6]. In addition, the combination of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) has also been proven effective in enhancing optimization outcomes by outperforming individual implementations of these algorithms. For instance, a study achieved 28% to 48% improvements in electricity demand forecasting using a hybrid ANN-GA-PSO model over standalone models [7].

After reviewing the literature, we developed a strategy to optimize lactic acid production using an ANN model as the fitness function. This approach aims to improve production efficiency and validate the effectiveness of both individual and hybrid optimization techniques in bioprocess applications.

3. Artifical Neural Network

A neural network can be represented mathematically as a function $f(x;\theta)$, where x is the input to the network, θ represents the parameters of the network, and f is a composition of multiple functions [1].

Let $z^{(l)}$ be the input to layer l, and $a^{(l)}$ be the output of layer l after applying a non-linear activation function g:

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}, \quad a^{(l)} = g(z^{(l)})$$

where $W^{(l)}$ is the weight matrix and $b^{(l)}$ is the bias vector for layer l [2].

The input to the first layer is simply the input x: $a^{(0)} = x$.

The output of the neural network is given by the output of the last layer:

$$f(x;\theta) = a^{(L)} = g(z^{(L)})$$

where L is the index of the last layer.

The parameters θ of the neural network consist of all the weight matrices and bias vectors:

$$\theta = W^{(1)}, b^{(1)}, \ldots, W^{(L)}, b^{(L)}$$

The objective of training the neural network is to minimize a loss function $J(\theta)$, which measures the difference between the predicted output of the network and the actual output. This is typically done using backpropagation, which involves calculating the gradient of the loss function with respect to the parameters, and then updating the parameters using an optimization algorithm such as stochastic gradient descent:

$$\theta \leftarrow \theta - \alpha\frac{\partial J(\theta)}{\partial\theta}$$

where α is the learning rate.

4. Genetic Algorithms

A genetic algorithm is a search algorithm that works by maintaining a population of candidate solutions and evolving the population over time using operations inspired by biological evolution, such as selection, crossover, and mutation. [8] Let P be the population of candidate solutions, each represented as a string of n genes or variables, and let $f(x)$ be the fitness function that evaluates the quality of a solution x.

The genetic algorithm works by iteratively performing the following steps:

- Selection: Choose a subset of the population with high fitness values to be the parents of the next generation. The probability of selection is proportional to the fitness value of each individual. [9]
- Crossover: Create new individuals by combining the genes of two parents using a crossover operator. This creates offspring that inherit traits from both parents. [10]
- Mutation: Introduce random changes to the genes of the offspring using a mutation operator. This introduces genetic diversity and prevents the population from converging to a suboptimal solution. [11]
- Replacement: Replace some members of the current population with the offspring generated by crossover and mutation. [12]
- Termination: Stop the algorithm when a termination condition is met, such as a maximum number of generations or a satisfactory level of fitness.

The mathematical background of genetic algorithms involves defining the fitness function $f(x)$, the selection method, the crossover operator, and the mutation operator. These operations can be defined in a way that optimizes the performance of the algorithm for a specific problem. The algorithm also requires parameters such as population size, crossover rate, and mutation rate, which can be tuned to optimize its performance. The goal of the genetic algorithm is to find a solution that maximizes or minimizes the fitness function, depending on the problem. The algorithm converges towards an optimal solution over time, but the quality of the solution depends on the problem, the parameters, and the design of the genetic algorithm. The flowchart for GA in shown in figure 1.

5. Particle Swarm Optimization

PSO is a population-based optimization algorithm that was introduced by Kennedy and Eberhart in 1995 [13]. The algorithm is inspired by the collective behavior of social organisms, such as bird flocks and fish schools, which exhibit coordinated movements that enable them to find food, avoid predators, and migrate. In PSO, a population of particles moves through the search space to find the optimal solution to an optimization problem.

The basic PSO algorithm can be described as follows:

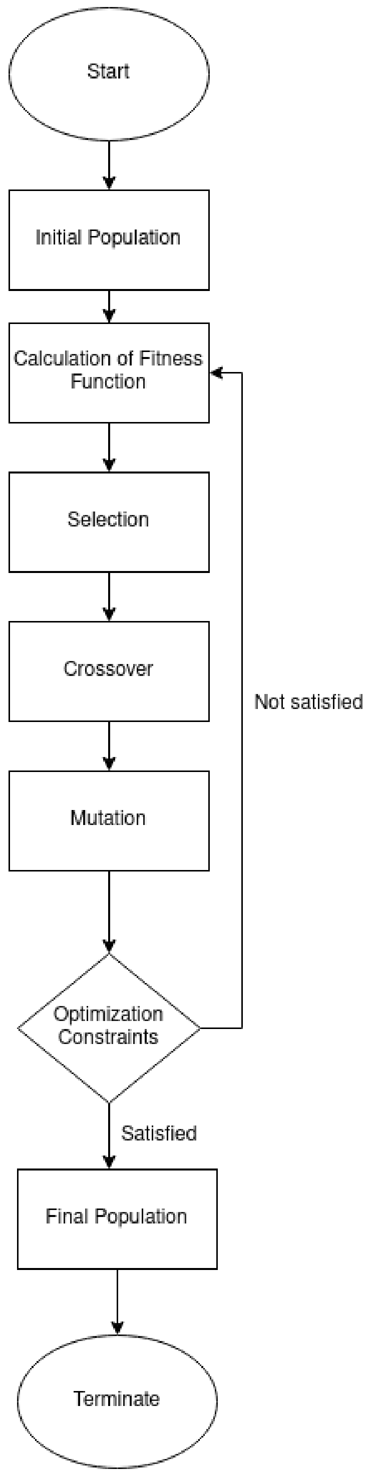


Figure 1: Genetic Algorithm Flowchart

- Initialize the population of particles with random positions and velocities.
- Evaluate the fitness of each particle using a fitness function.
- Update the particle's velocity using the current position, the particle's best position, and the global best position of the swarm.
- Update the particle's position using the new velocity.
- If the stopping criterion is met, stop. Otherwise, return to step 2.

The velocity of each particle is updated based on its own best position, called the personal best (Pbest), and the global best position of the swarm, called the global best (Gbest). The velocity update equation can be written as follows:

$$v_i(t+1) = wv_i(t) + c1r1(Pbest_i - x_i(t)) + c2r2(Gbest - x_i(t))$$

where $v_i(t)$ is the velocity of particle i at time t, w is the inertia weight, c1 and c2 are the cognitive and social parameters, r1 and r2 are random values between 0 and 1, $Pbest_i$ is the personal best position of particle i, $x_i(t)$ is the current position of particle i at time t, and $Gbest$ is the global best position of the swarm.

The position update equation can be written as follows:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

where $x_i(t)$ is the current position of particle i at time t.

The inertia weight w controls the balance between exploration and exploitation in the algorithm. A high inertia weight can lead to better exploration, while a low inertia weight can lead to better exploitation. The inertia weight is typically decreased over time to gradually shift the algorithm from exploration to exploitation. The most commonly used formula for the inertia weight is:

$$w(t) = w_max - ((w_max - w_min)/max_iter) * t$$

where w_max and w_min are the maximum and minimum inertia weights, respectively, max_iter is the maximum number of iterations, and t is the current iteration.

The cognitive and social parameters, c1 and c2, control the influence of the particle's own best position and the global best position on the particle's velocity, respectively. The values of c1 and c2 are typically set to constant values between 0 and 2.

PSO has been successfully applied to a wide range of optimization problems, including engineering design, feature selection, and neural network training. One of the advantages of PSO is that it is relatively easy to implement and requires few tuning parameters compared to other optimization algorithms.

Table 1: Results

Eqn	Model	Meta-Heuristic	True Maxima	True Optimal Points	Predicted Maxima	Predicted Optimal Points	RMSE for ANN
1	ANN	GA	8	$(-2, -2)$	7.598332	(-1.97, -1.99)	0.5179
1	ANN	PSO	8	$(-2, -2)$	7.438	(-2.32, -1.66)	0.5179
2	ANN	GA	45	(-5, 3)	46.5717	(-4.89, 3.11)	3.3096
2	ANN	PSO	45	(-5, 3)	44.89	(-5.12, 2.98)	3.3096

6. Classical Method for Lactic Acid Production

The Response Surface method(RSM) is used in lactic acid production in industry. It is a widely accepted mathematical approach to the fermentation process. A large number of experiments are required to find the optimal solution for lactic acid production.

7. Methods and Results

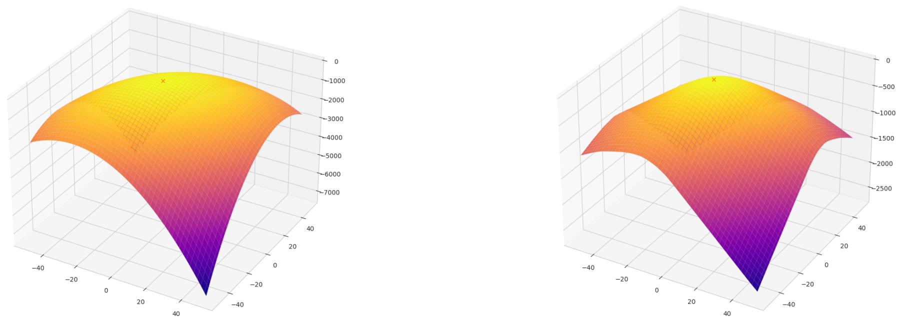
7.1. *On Multivariate Functions*

Before training the neural network using a lactic acid dataset, we first trained our model using a dataset consisting of multivariate polynomial functions. The purpose of this step was to demonstrate the ability of a simple artificial neural network model to represent multivariate functions. A multivariate function is a mathematical function with multiple inputs and a definite output. To ensure the robustness of our results, we utilized various forms of multivariate functions in our experiment. We generated the necessary data from the multivariate functions and split it into 80% for training and 20% for testing the model. This resulted in a dataset that was suitable for training the neural network. Taking Following Equations

$$f(x, y) = xy - x^2 - y^2 - 2x - 2y + 4 \tag{1}$$

$$f(x, y) = (4x - x^2) * \cos(y) \tag{2}$$

Visualization of the above equation and the visualization obtained from modeling those equations using ANN is shown in Figure 2 and 4. Similarly, the actual optimal values for the above equation and the optimal values obtained from applying GA and PSO using ANN as a fitness function are shown in table 1



(a) Original

(b) ANN Prediction

Figure 2: Visualization of Equation 1

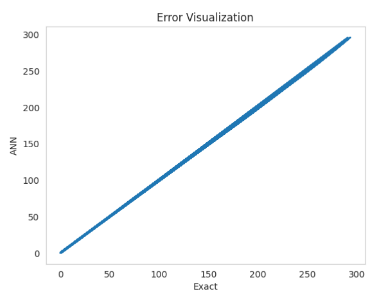
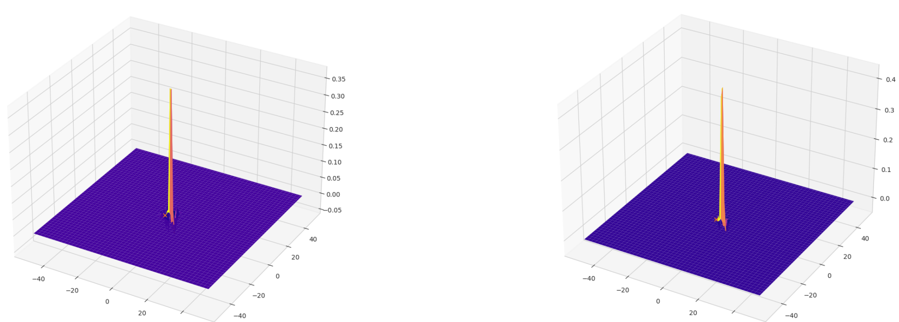


Figure 3: Equation 1 Modeling Error



(a) Original

(b) ANN Prediction

Figure 4: Visualization of Equation 2

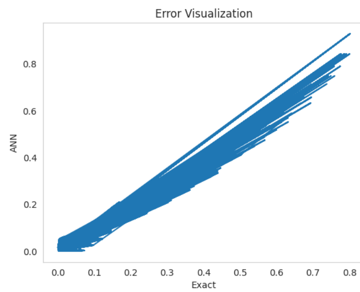


Figure 5: Equation 2 Modeling Error

	pH	Temperature	MgSO4	MnSO4	K2HPO4	CaCO3	Tween80	Glycerol	Yeast Extract	(NH4)2SO4	Response
0	4.0	40.0	0.1	1.0	5.0	4.0	1.0	0.1	1.0	30.0	1.753890
1	7.0	40.0	1.0	0.1	0.2	4.0	1.0	0.1	10.0	30.0	0.961810
2	4.0	40.0	1.0	1.0	5.0	0.0	0.1	4.0	10.0	5.0	1.852900
3	4.0	30.0	1.0	1.0	0.2	4.0	1.0	0.1	1.0	5.0	1.074965
4	7.0	30.0	0.1	1.0	5.0	0.0	1.0	4.0	1.0	5.0	1.859972

Figure 6: Sample of the dataset

7.2. Application to Lactic Acid Production

Our research mainly aims to apply the techniques and methodologies used above to lactic acid production. Generally, lactic acid is produced in the lab with multiple hit-and-trial experiments to maximize the output. Till now, we have used a dataset generated from a multivariate function to train an Artificial neural network model. Now we will use the real dataset from the laboratory to train the ANN model and optimize the output using GA and PSO. Figure 6 is sample taken from the dataset we used.

The input for our ANN is (PH, Temperature, MgSO4, MnSO4, K2HPO4, CaCO3, Tween80, Glycerol, Yeast Extract, (NH4)2SO4) and the output is Response. We trained the neural network model using the above dataset and the predicted set of values is optimized using the genetics algorithm(GA) and particle swarm optimization (PSO) algorithm.

Results From training ANN model with the above dataset

Root mean square error(RMSE) = 0.378

Based on the optimized medium value shown on table 2 we got a yield (Optimal Yield) of **3.08** from GA and yield of **3.1** from PSO.

Table 2: Optimal Medium Value

Medium	Predicted Optimal Value	
	From GA	**From PSO**
pH	5.22	5.36
Temperature	45.12	46.01
MgSO4	0.73	0.71
MnSO4	0.27	0.29
K2HPO4	2.70	2.68
CaCO3	2.46	2.53
Tween80	0.05	0.05
Glycerol	2.89	2.76
Yeast Extract	13.58	13.39
(NH4)2SO4	7.67	7.78

8. Conclusion and Discussion

From the above experiments, we concluded that a multivariate mathematical function can be represented by a simple ANN model if trained properly with enough datasets. Meta-heuristic algorithms such as genetics algorithm(GA) and particle swarm optimization(PSO) algorithm can be used to find the optimal solution to an equation. The main finding of this research work is that we can use the real-time dataset generated from the lab to train an ANN model and find the optimal solution for maximizing lactic acid production. Here we have used the lactic acid production dataset from the biotech lab, trained the ANN model using the dataset, and found the optimal requirements of various chemicals for maximizing the lactic acid production. This method helps to save a lot of time and energy used in multiple hit-and-trial experiments in the lab.

References

[1] L. J. Lancashire, C. Lemetre, G. R. Ball, An introduction to artificial neural networks in bioinformatics–application to complex microarray and mass spectrometry datasets in cancer studies, Briefings in Bioinformatics 10 (3) (2008) 315–329. doi:10.1093/bib/bbp012.
URL http://dx.doi.org/10.1093/bib/bbp012

[2] W. A. Owusu, S. A. Marfo, Artificial intelligence application in bioethanol production, International Journal of Energy Research 2023 (2023) 1–8. doi:10.1155/2023/7844835.
URL http://dx.doi.org/10.1155/2023/7844835

[3] R. Nasaruddin, M. Jami, M. Alam, The potential of artificial neural network (ann) in optimizing media constituents of citric acid production by solid state bioconversion, International Food Research Journal 19 (2012) 491–497.

[4] K. M. Desai, S. A. Survase, P. S. Saudagar, S. Lele, R. S. Singhal, Comparison of artificial neural network (ann) and response surface methodology (rsm) in fermentation media optimization: Case study of fermentative production of scleroglucan, Biochemical Engineering Journal 41 (3) (2008) 266–273. doi:10.1016/j.bej.2008.05.009.

[5] F. Machado Cavalcanti, C. Emilia Kozonoe, K. André Pacheco, R. Maria de Brito Alves, Application of Artificial Neural Networks to Chemical and Process Engineering, IntechOpen, 2021. doi:10.5772/intechopen.96641. URL http://dx.doi.org/10.5772/intechopen.96641

[6] D. Devikanniga, K. Vetrivel, N. Badrinath, Review of meta-heuristic optimization based artificial neural networks and its applications, Journal of Physics: Conference Series 1362 (1) (2019) 012074. doi:10.1088/1742-6596/1362/1/012074.
URL http://dx.doi.org/10.1088/1742-6596/1362/1/012074

[7] A. Anand, L. Suganthi, Hybrid ga-pso optimization of artificial neural network for forecasting electricity demand, Energies 11 (4) (2018) 728. doi:10.3390/en11040728.
URL http://dx.doi.org/10.3390/en11040728

[8] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, The MIT Press, 1992. doi:10.7551/mitpress/1090.001.0001.
URL https://doi.org/10.7551/mitpress/1090.001.0001

[9] D. E. Goldberg, J. H. Holland, Genetic algorithms and machine learning, Machine Learning 3 (1988) 95–99.
URL https://api.semanticscholar.org/CorpusID:2043246

[10] L. Davis, Applying adaptive algorithms to epistatic domains, in: Proceedings of the 9th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'85, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1985, p. 162–164.

[11] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, USA, 1998.

[12] L. D. Whitley, T. Starkweather, D. Fuquay, Scheduling problems and traveling salesmen: The genetic edge recombination operator, in: International Conference on Genetic Algorithms, 1989.
URL https://api.semanticscholar.org/CorpusID:12884640

[13] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, ICNN-95, IEEE. doi:10.1109/icnn.1995.488968.