

OPTIMIZATION OF A SOLID STATE FERMENTATION BASED ON RADIAL BASIS FUNCTION NEURAL NETWORK AND PARTICLE SWARM OPTIMIZATION ALGORITHM

Badia Dandach-Bouaoudat, Farouk Yalaoui, Lionel Amodeo
University of Technology of Troyes, 12 rue Marie Curie, Troyes, France

Françoise Entzmann
Ets J.Soufflet, Quai Sarrail, Nogent sur Seine, France

Keywords: Solid state fermentation, Enzyme production, Optimization, Response surface methodology, Neural network, Particles swarm optimization algorithm.

Abstract: Radial basis function neural network (RBF) and particle swarm optimization (PSO) are used to model and optimize a solid state fermentation (SSF) for production of the enzyme. Experimental data reported in the literature are used to investigate this approach. The response surface methodology (RSM) is applied to optimize PSO parameters. Using this procedure, two artificial intelligence techniques (RBF-PSO) have been effectively integrated to create a powerful tool for bioprocess modelling and optimization. This paper describes the applications of this approach for the first time in the solid state fermentation optimization.

1 INTRODUCTION

Fermentation processes are used to produce various substances in the pharmaceutical, chemical and food industries. The performance of fermentation processes depends on many factors, including pH, temperature, ionic strength, agitation speed, and aeration rate in the aerobic fermentation (Kennedy and Krouse, 1999). To achieve the best performance of fermentation processes, various process optimization strategies were developed by scientists. The most frequently used optimization is response surface methodology (RSM), which includes factorial design and regression analysis, seeks to identify and optimize significant factors to maximize the response (cell density, high yields of the desired metabolic products or enzyme levels in the microbial system). RSM yields a model, which describes the relationship between the independent and dependent variables of the processes. The most widely used simulating models are second order polynomials (Ceylan et al., 2008) and (Chang et al., 2008), and now RSM has been widely applied in the bioprocess optimization (Kunamneni and Singh, 2005) and (Ustok et al., 2007).

In recent years, a limited number of studies have

investigated the possibility of using non-statistical techniques, such as artificial intelligence (AI), for developing non-linear empirical models. The most commonly used AI are artificial neural networks (ANNs). ANNs are superior and more accurate modelling techniques when compared to RSM and represent the non-linearities in a much better way (Dutta et al., 2004). The most frequently used ANNs is a radial basis function (RBF) neural network. As a universal function approximator under certain general conditions (Wilson et al., 1999).

One type of evolutionary technique used in computer science is particle swarm optimization (PSO). This technique allows one approximate optimization and search problem solutions. Kennedy and Eberhart were the first to propose the PSO algorithm (Kennedy et al., 1995). This algorithm is based on the premise that evolution of a species is advanced by the social sharing of information among members of the said species. PSO is a perfect candidate of optimization tasks based on a number of advantages compared to other algorithms. The PSO algorithm is strong enough to handle complex situations such as non-linear and non-convex design spaces with discontinuities. Also continuous, discrete and integer variable

types can be handled easily. Also, the PSO optimization method is more efficient than other equally robust designs. There are fewer function evaluations yet the results are of equal or better quality than competitors (Hu et al., 2003) and (Hassan et al., 2005).

In this work, RBF neural network coupling PSO algorithm (RBF-PSO) was used to model and optimize the mixing performance of solid state fermentation. Three independent process variables including temperature, agitation, and inoculums size were considered for optimization. The optimization objective was to achieve the maximum alpha-amylase activity at an optimal combination of three independent variables. The RBF neural network was used to develop the mathematical function of alpha-amylase activity with temperature, agitation, and inoculums size. Then PSO algorithm was used to execute the optimization task to achieve the maximum alpha-amylase activity. The response surface methodology (RSM) is applied to optimize PSO parameters.

2 MODELIZATION

2.1 RBF Neural Network

RBF neural network is structured by embedding radial basis function with a two-layer feed-forward neural network. Such a network is characterized by a set of input and set of outputs. In between the inputs and outputs there is a layer of processing units called hidden units. Each of them implements a radial basis function. The architecture of RBF network is shown in Figure 1.

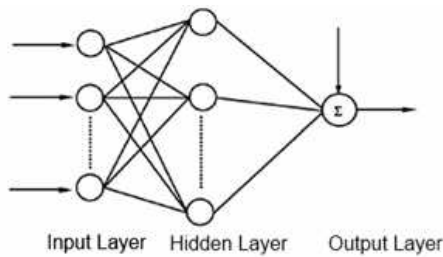


Figure 1: structure of RBF neural network for process modeling.

Mathematically RBF neural network can be formulated as:

$$g(x) = \sum_{k=1}^m \lambda_k \phi_k(\|x_i - c_k\|) \quad (1)$$

Where m is the neuron number of hidden layer, which is equal to cluster number of training set. $\|x_i - c_k\|$

represents the distance between the data point x_i and the RBF center c_k . λ_k is the weight related with RBF center c_k . Therefore, the output of RBF neural network is a weighted sum of the hidden layer's activation functions. Various functions have been tested as activation functions for RBF networks. Here we adopt the most commonly used Gaussian RB functions as basis functions shown in equation (2):

$$\phi_k(x_i) = \frac{R_k(x_i)}{\sum_{i=1}^m R_i(x_i)} \quad (2)$$

$$R_k(x_i) = \exp\left(-\frac{\|x_i - c_k\|^2}{2\sigma_k^2}\right) \quad (3)$$

In equation (3), σ_k indicates the width of the k th Gaussian RB functions. One of the σ_k selection methods is shown as follows.

$$\sigma_k^2 = \frac{1}{M_k} \sum_{x \in \theta_k} \|x_i - c_k\|^2 \quad (4)$$

here θ_k the k th cluster of training set, and M_k is the number of sample data in the k th cluster.

3 OPTIMIZATION

3.1 PSO Algorithm

Stochastic in nature, the particle swarm process updates the position of each particle in the swarm using the velocity vector. This vector is updated using the memory of each particle and the entire swarm. This allows the position of each individual particle to be updated based on the entire swarm. As the swarm adapts to its environment each particle can return to regions of space that are promising and also search for better positions.

Numerically, the position x of a particle i at iteration $k + 1$ is updated as shown in equation (5) and illustrated in figure 2.

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (5)$$

Where v_{k+1}^i is the corresponding updated velocity vector, the velocity vector of each particles is calculated as shown in equation (6),

$$v_{k+1}^i = wv_k^i + w_1r_1(p_k^i - x_k^i) + w_2r_2(p_k^g - x_k^i) \quad (6)$$

Where v_k^i is the velocity vector at iteration k , r_1 and r_2 represents random numbers between 0 and 1; p_k^i represents the best ever particle position of particle i , and p_k^g corresponds to the global best position in the swarm up to iteration k .

Terms that remain are parameters that depend on the problem. These include w_1 , w_2 and w , which stand for the particles confidence in itself (cognitive parameter) and in the swarm (social parameter) and also inertial weight, respectively. The PSO convergence behavior, which is used to control the exploration capabilities of the swarm, relies largely on the inertial weight of the particle. The inertial weight impacts the current velocity, which is based on previous velocities. Therefore, as the inertial weight changes, so does the ability of the particle to roam. If inertial weight is large, the particle can explore the design space in a broader manner, while small inertial weights limit the updates of velocity to local regions within the design space.

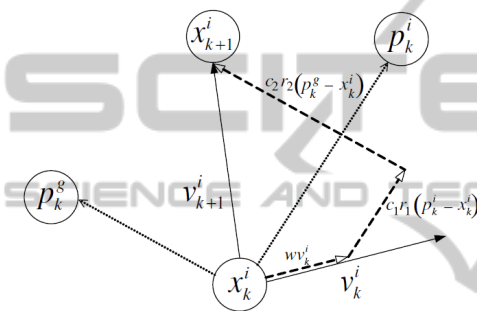


Figure 2: PSO position and velocity update.

3.2 Response Surface Methodology (RSM)

Response surface methodology combines statistical experimental designs and empirical model building by regression for the purpose of process or product optimization. Statistical experimental design is a powerful method for accumulating informations about a process efficiently and rapidly from a small number of experiments, there by minimizing experimental costs. An empirical model is then used to relate the response of the process to some independent variables. This usually entails fitting a quadratic polynomial to the available data by regression analysis. The general form of the quadratic polynomial is:

$$Y = b_0 + \sum b_i X_i + \sum b_i X_i^2 + \sum b_{ij} X_i X_j + e \quad (7)$$

Where Y is the predicted response, X_i and X_j stand for the independent variables, b_0 is the intercept, b_i and $b_{i,j}$ terms are regression coefficients, and e is a random error component. A near-optimum solution can then be deduced by calculating the derivatives of equation (7) or by mapping the response of the model into a surface contour plot.

4 RESULTS AND DISCUSSION

4.1 RBF Modeling

In the RBF estimator for the fermentation example examined in this work, there are three nodes (x_1 : temperature ($^{\circ}\text{C}$), x_2 : agitation (rpm) and x_3 : Log_{10} of inoculums size (spore/ml) in the input layer and an output node for alpha-amylase activity (Y). with the given structure, the RBF neural network is trained by the sampling data imported from (Kammoun et al., 2008) to obtain a desired estimator.

Here the R^2 of 99.8% indicates that the experimental and predicted values are in a good agreement.

4.2 Optimization by PSO Algorithm

In the present work, we used the RBF neural network to produce the fitness function. We developed a (RBF-PSO) algorithm with Matlab software, this algorithm has 2 steps: modeling and optimization, various steps involved in this study were represented schematically in Figure 3.

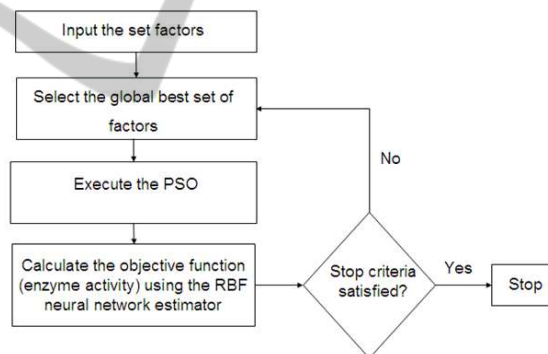


Figure 3: Schematic representation of RBF-PSO algorithm for bioprocess optimization.

Based on the particle and velocity updates as explained in equation (6), the PSO algorithm is constructed as follows:

1. Initialize a set of particles positions x_0^i and velocities v_0^i randomly distributed throughout the design space bounded by specified limits, and set $p_k^i = x_k^i$.
2. Evaluate the objective function values $f(x_k^i)$ using the design space positions x_k^i . That is, employ the particle's position vector as the the input of RBF estimator.
3. Update the optimum particle position p_k^i at current iteration (k) and global optimum particle position p_k^g .

4. Update the position of each particle using its previous position and update velocity vector as specified in equation (5) and equation (6).
5. Repeat steps 2-4 until the stopping criteria is met. For the current implementation the stopping criteria is defined based on the number of iterations reached.

4.3 Tuning PSO Parameters

The role of the inertia weight w , plays a critical role in the PSO's convergence behavior in equation (6). The inertia weight should consist of a trade-off between global and local exploration abilities of the swarm to therefore control the impact of previous velocities on the current velocity. Large inertia weight results in particles searching new areas, while a small inertia weight results in particles exploring locally. By finding a suitable value of inertia weight, a balance between wide-range and local exploration can be achieved. By doing this, the optimum solution can be found the most efficiently and with the fewest iterations. Through experimentation, it was found that setting the inertia weight to a high level results in global exploration, which in turn produces more results (Shi and Eberhart, 1998). This value can then be lowered, refining the solutions. Therefore, a value of w starting at 1.2 and gradually declining towards 0 can be considered as a good choice for w .

The algorithm can be further improved by fine-tuning the parameters w_1 and w_2 , in equation (6). the results could be converged on faster and the local explorat alleviated. In (Kennedy, 1998) an extensive study of the acceleration parameter in PSO's first version can be found. Recent work has also come up with further suggestions for choosing w_1 and w_2 . This work suggests that it may be in one's best interest to choose w_1 , the cognitive parameter, to be larger than w_2 , the social parameter, with the limitation that $w_1 + w_2 \leq 4$ (Carlisle and Dozier, 2001).

The parameters r_1 and r_2 are used to maintain the diversity of the population, and they are uniformly distributed in the range [0, 1].

In this work, optimization of the parameters of the PSO algorithm:

- N : swarm size ,
- w_1 : cognitive parameter,
- w_2 : social parameter,

was done by Box-Behnken design especially made to require three levels coded as (-), (0) and (+) ($N = 13$) (13 experiments and three factors at three levels) under the response surface methodology

(RSM). Table 1 shows the different levels of each of the parameters.

Table 1: Parameters values for Box-Behnken design.

Factors	Basical level	Variation interval	Value of the factor
w_1 : Cognitive parameter	2	2	0 2 4
w_2 : Swarm parameter	2	2	0 2 4
N : Swarm size	65	35	30 65 100

Table 2: Box-Behnken experimental design used to optimize parameters of PSO algorithm.

PSO code	w_1	w_2	N	Alpha-amylase activity (U/ml)
PSO 1	0	2	30	40.3668
PSO 2	0	2	100	40.3478
PSO 3	4	2	30	39.0155
PSO 4	4	2	100	40.3694
PSO 5	2	0	30	24.0529
PSO 6	2	0	100	29.7599
PSO 7	2	4	30	40.2980
PSO 8	2	4	100	40.0989
PSO 9	0	0	65	27.9151
PSO 10	4	0	65	27.4070
PSO 11	0	4	65	40.3685
PSO 12	4	4	65	35.0228
PSO 13	2	2	65	40.3731

Box-Behnken design is a fractional factorial design obtained by combining two-level factorial designs with incomplete block designs. The response surface methodology was used to analyses the experimental design data. In order to be correlated to the independent variables, the response variable was fitted by a second order model.

Table 2 illustrates the Box-Behnken experimental design of the three independent variables together with the experimental result. By applying multiple regression analysis on the experimental data, the following second-order polynomial equation was developed to clarify the relationship of the optimal solution (Y), swarm size (N), cognitive parameter (w_1) and social parameter (w_2):

$$Y = 40.374 + 0.980N - 0.898w_1 + 5.707w_2 + 0.137N^2 - 0.487w_1^2 - 7.209w_2^2 + 0.343Nw_1 - 1.227Nw_2 - 1.209w_1w_2 \quad (8)$$

The R^2 of 98% indicates that the experimental and predicted values are in a good agreement.

The response surface curves described by regression model are shown in figure 4.

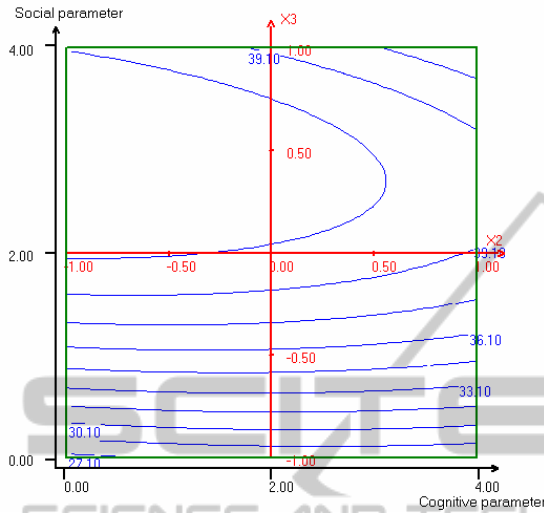


Figure 4: Responses surface plot of cognitive parameter (w_1), social parameter (w_2) and their mutual interaction on alpha-amylase activity (Y), with an optimum level of swarm size (N) of 65.

For equation derived from the differentiation of equation (8), the optimal values of N , w_1 and w_2 in coded units were found to be 0, -0.980 and 0.509 respectively. Correspondingly, we can obtain the optimal combination of three parameters 65, 0.04 and 3.02 respectively.

In order to determine the maximum number of generations, we have plotted the response curve in function of number iteration, and we have presented the results in figure 5. From figure, we deduce that the maximum number of generations to reach the better solution ($Y = 40.3739U/ml$) is 18 generations.

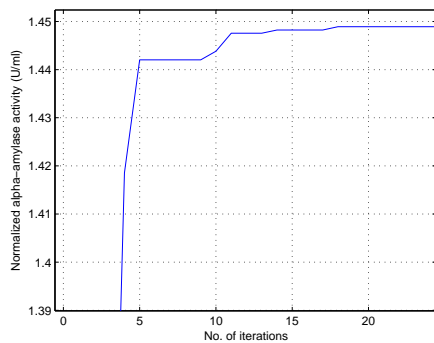


Figure 5: Convergence process of solving optimization task with PSO algorithm.

Table 3: Results after tuning parameters of PSO algorithm.

w : inertia parameter	starting at 1.2 and gradually declining towards 0
N : Swarm size	65
w_1 :cognitive parameter	0.04
w_2 : Social parameter	3.02
G : Number of generation	18

4.4 Optimization Bioprocess Example

The result obtained are shown in the first row of table 4. As it is shown in the last column of the table, the maximum value for alpha-amylase activity by RBF-PSO algorithm is $40.3739U/ml$ which is much better than the value obtained by SM-GA algorithm reported in (Dandach-Bouaoudat et al., 2010).

Table 4: Results after tuning parameters of PSO algorithm.

Approaches	x_1 ($^{\circ}C$)	x_2 <i>rpm</i>	x_3 <i>spore/ml</i>	Y <i>U/ml</i>
RBF-PSO	21.88	303.22	7	40.3739
SM-GA	25.14	249.36	7.23	31.7

5 CONCLUSIONS

This work found that RBF neural network provided good fits to experimental data. The hybrid RBF-PSO approach described in this work serves as a viable alternative for the modelling and optimization of fermentation process. Alpha-amylase activity rises up to $40.37U/ml$ under the optimal culture conditions obtained by RBF-PSO approach. This work indicates that the coupling of RBF neural networks with PSO algorithm has good predictability and accuracy in optimizing the multi-factor, non-linear, and time-variant bioprocess. The knowledge and information obtained here may be also helpful to the other industrial bioprocess to improve productivity.

Perspectives on work is conducting more experiments, especially for those data that have more parameters need to be optimized. We will also test other artificial intelligence techniques for modelling and optimization bioprocess and compared with the proposed method.

REFERENCES

- Carlisle, A. and Dozier, G. (2001). An off-the-shelf PSO. In *Proceedings of the workshop on particle swarm optimization*. Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI.
- Ceylan, H., Kubilay, S., Aktas, N., and Sahiner, N. (2008). An approach for prediction of optimum reaction conditions for laccase-catalyzed bio-transformation of 1-naphthol by response surface methodology (RSM). *Bioresource technology*.
- Chang, S., Shaw, J., Yang, K., Chang, S., and Shieh, C. (2008). Studies of optimum conditions for covalent immobilization of *Candida rugosa* lipase on poly (γ -glutamic acid) by RSM. *Bioresource technology*.
- Dandach-Bouaoudat, B., Yalaoui, F., Amodeo, L., and Entzmann, F. (2010). Optimization of a solid state fermentation using Genetic Algorithm. In *5th International Symposium on Hydrocarbons and Chemistry-Sidi Fredj, Algiers*.
- Dutta, J., Dutta, P., and Banerjee, R. (2004). Optimization of culture parameters for extracellular protease production from a newly isolated *Pseudomonas* sp. using response surface and artificial neural network models. *Process biochemistry*.
- Hassan, R., Cohanin, B., De Weck, O., and Venter, G. (2005). A comparison of particle swarm optimization and the genetic algorithm. In *Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference*.
- Hu, X., Eberhart, R., and Shi, Y. (2003). Engineering optimization with particle swarm. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003. SIS'03*.
- Kammoun, R., Naili, B., and Bejar, S. (2008). Application of a statistical design to the optimization of parameters and culture medium for α -amylase production by *Aspergillus oryzae* CBS 819.72 grown on gruel (wheat grinding by-product). *Bioresource technology*.
- Kennedy, J. (1998). The behavior of particles. In *Evolutionary Programming VII*. Springer.
- Kennedy, J., Eberhart, R., et al. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks*. Piscataway, NJ: IEEE.
- Kennedy, M. and Krouse, D. (1999). Strategies for improving fermentation medium performance: a review. *Journal of Industrial Microbiology and Biotechnology*.
- Kunamneni, A. and Singh, S. (2005). Response surface optimization of enzymatic hydrolysis of maize starch for higher glucose production. *Biochemical Engineering Journal*.
- Shi, Y. and Eberhart, R. (1998). Parameter selection in particle swarm optimization. In *Evolutionary Programming VII*. Springer.
- Ustok, F., Tari, C., and Gogus, N. (2007). Solid-state production of polygalacturonase by *Aspergillus sojae* ATCC 20235. *Journal of biotechnology*.
- Wilson, D., Irwin, G., and Lightbody, G. (1999). RBF principal manifolds for process monitoring. *IEEE Transactions on Neural Networks*.