0273-2289/06/129–132/969–984/$30.00

Ethyl Alcohol Production Optimization by Coupling Genetic Algorithm and Multilayer Perceptron Neural Network

ELMER CCOPA RIVERA,* ALINE C. DA COSTA, MARIA REGINA WOLF MACIEL, AND RUBENS MACIEL FILHO

DPQ/FEQ/UNICAMP, Campinas, SP, Brasil Cx. Postal 6066, 13081-970;
E-mail: elmer@feq.unicamp.br

Abstract

In this present article, genetic algorithms and multilayer perceptron neural network (MLPNN) have been integrated in order to reduce the complexity of an optimization problem. A data-driven identification method based on MLPNN and optimal design of experiments is described in detail. The nonlinear model of an extractive ethanol process, represented by a MLPNN, is optimized using real-coded and binary-coded genetic algorithms to determine the optimal operational conditions. In order to check the validity of the computational modeling, the results were compared with the optimization of a deterministic model, whose kinetic parameters were experimentally determined as functions of the temperature.

Index Entries: Alcoholic fermentation process; artificial intelligence; design of experiments; modeling; penalty function.

Introduction

Demand for ethyl alcohol as a renewable fuel for automotive industries continues to be high as countries are pushing for greater energy self-sufficiency. As a large tropical country, Brazil has a high potential for the use of biomass *(1)*. Sugarcane products are today the most economically important biomass source for energy cogeneration. One of the main cost contributive parameters for ethanol production from biomass is the cost of the raw material. Such cost can be reduced if the conversion efficiency of the raw material is maximized. This would also avoid the additional energy required to remove water from ethanol, which is one of the largest costs involved in the production of ethanol to be used as a fuel additive. The productivity, conversion and yield can be optimized both at the biochemical level and at the level of process operation *(2)*. Some studies have used mathematical models based on fundamental mass balances and kinetic equations with experimentally determined parameters to investigate

*Author to whom all correspondence and reprint requests should be addressed.

the influence of operating variables on the conversion, productivity, and yield (3,4).

Artificial intelligence (AI), such as artificial neural networks and genetic algorithms (GA), covers a wide range of techniques and tools that facilitate decision making. These methods have already been successfully applied in the optimization and control of bioprocesses for more than 20 yr (5). Moreover, recent studies in chemical and biochemical research showed that these tools can often be favorably combined especially for modeling, optimization (6,7), and control (8).

This work focuses on the process operation aspects using model-based optimization of an extractive alcoholic fermentation process. Silva et al. (9) have shown that a scheme combining a fermentor with a vacuum flash vessel presents several positive features and better performance than a conventional industrial process (10). The objective in optimizing this process is to maximize productivity, although maintaining a high conversion of substrates to ethanol in the fermentor. A comparison is made between the performances of two models when the process is optimized using a real-coded genetic algorithm and a binary-coded genetic algorithm. In real coded genetic algorithms, the decision variables are coded in real numbers, unlike the binary numbers used in the binary coded genetic algorithms. The first model used is a deterministic model, whose kinetics parameters were experimentally determined as functions of the temperature, the second is a nonlinear model represented by a multilayer perceptron neural network (MLPNN).

A methodology based on AI was used to determine the best conditions from the interactive effect of four variables: inlet substrate concentration, cells recycle rate, residence time, and flash recycle.

Methods

A Neural Network Mathematical Model for the Ethanol Production Process

The process to be optimized is shown in Fig. 1 (9). The process consists of four interlinked units: the fermentor (ethanol production unit), the centrifuge (cell separation unit), the cell treatment unit, and the vacuum flash vessel (ethanol–water separation unit). The mathematical model is built up of five ordinary differential equations derived from mass and energy balances on the fermentor; algebraic equations for the mass balance on the centrifugue, cell treatment unit, and the purge, as well as the mass and energy balances on the flash tank. Details of the model can be found in Costa et al. (11). The objective of the optimization is to maximize productivity and conversion, which are strongly influenced by the inlet substrate concentration, S_0; cell recycle rate, R; residence time, t_r; and flash recycle rate, r (11).

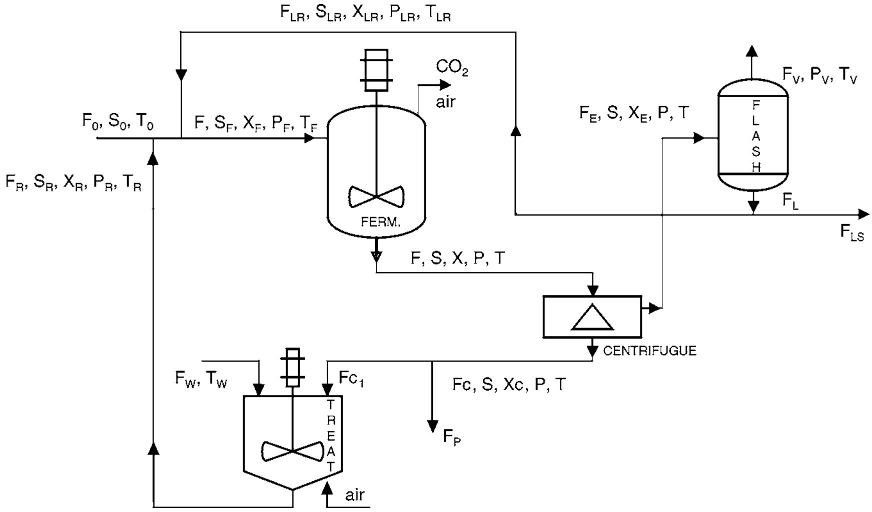

Fig. 1. Extractive alcoholic fermentation scheme.

Table 1
Coded Factor Levels and Equations to Determine the Real Values for the Input Variables Used in the Training of the Neural Networks

Coded factor level	S_0	t_r (h)	R	r
Level + 2	$(+2) \cdot \Delta S_0(i) + 180$	$(+2) \cdot \Delta t_r(i) + 1.75$	$(+2) \cdot \Delta R(i) + 0.35$	$(+2) \cdot \Delta r(i) + 0.4$
Level + 1	$(+1) \cdot \Delta S_0(i) + 180$	$(+1) \cdot \Delta t_r(i) + 1.75$	$(+1) \cdot \Delta R(i) + 0.35$	$(+1) \cdot \Delta r(i) + 0.4$
Central point (0)	180	1.75	0.35	0.4
Level-1	$(-1) \cdot \Delta S_0(i) + 180$	$(-1) \cdot \Delta t_r(i) + 1.75$	$(-1) \cdot \Delta R(i) + 0.35$	$(-1) \cdot \Delta r(i) + 0.4$
Level-2	$(-2) \cdot \Delta S_0(i) + 180$	$(-2) \cdot \Delta t_r(i) + 1.75$	$(-2) \cdot \Delta R(i) + 0.35$	$(-2) \cdot \Delta r(i) + 0.4$

In order to obtain simulation data for the training of the neural network, the inputs (S_0, R, t_r, and r) are distributed according to a design of experiment within the operational intervals (80, 280), (0.2, 0.5), (1.0, 2.5), and (0.2, 0.6), respectively. These intervals were defined based on prior knowledge of the process *(11)*.

A detailed description of the design of experiments theory can be found in *(12)*. In the present work a full factorial design 2^{4+} star configuration with a central point was used (Table 1). In order to have a large amount of training data, eight different values of ΔS_0, Δt_r, ΔR, and Δr were used, as shown in Table 2, so that eight factorial designs 2^{4+} star configuration were simulated. The data of productivity and conversion are shown in Fig. 2 as functions of S_0 and R.

Table 2
Values of ΔS_0, Δt_r, ΔR, and Δr to Simulate the Data Set, Used to Generate Table 1

i	ΔS_0(i) (kg/m^3)	Δt_r(i) (h)	ΔR(i)	Δr(i)
1	50	0.075	0.375	0.1
2	45	0.0675	0.3375	0.09
3	40	0.06	0.3	0.08
4	35	0.0525	0.2625	0.07
5	30	0.045	0.225	0.06
6	25	0.0375	0.1875	0.05
7	20	0.03	0.15	0.04
8	10	0.015	0.075	0.02

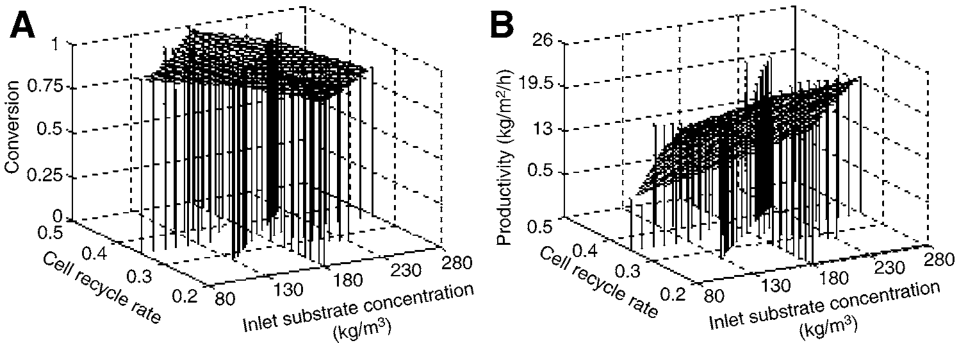

Fig. 2. Spatial distribution of the data set for identification (the partially collected data points would be located only on the mesh surface if only one full factorial 2^4+ star were done): **(A)** Conversion and **(B)** Productivity as functions of cells recycle rate and inlet substrate concentration.

Building the MLPNN Model

In the past years many works *(6,13)* have been proposed that use artificial neural networks as a substitute for deterministic models. The substitution of the deterministic model by an equivalent MLPNN at the optimization step presents the advantage of high speed processing.

One of the main current problems related to the alcoholic fermentation process is the lack of robustness of the fermentation in the presence of fluctuations in the quality of the raw material, which leads to changes in the kinetic behavior and influences yield, productivity, and conversion. These changes make the prediction of the process dynamic behavior with a single mathematical model difficult. In order to take the kinetic changes into account, the kinetic parameters of the model should be re-estimated. However, frequent re-estimation of these parameters in deterministic models is usually difficult, mainly owing to nonlinearities, great number of parameters, and interactions among them. The use of neural network

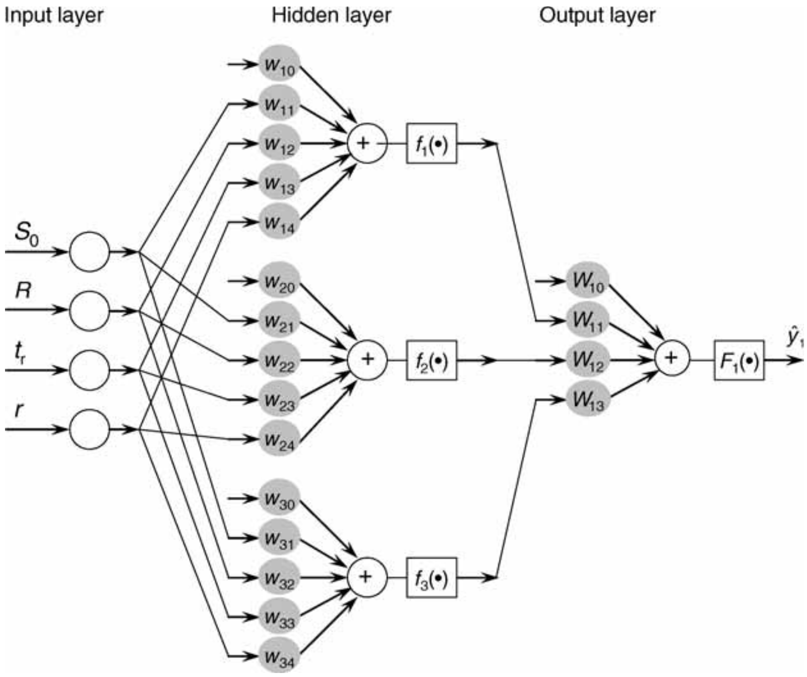

Fig. 3. Representation of a MLPNN with one hidden layer.

models is a good option in this case, because the changes in the kinetic behavior can be assessed by updating the network weights.

A MLPNN consists of three types of layers: an input layer, an output layer, and one or more hidden layers, as shown in Fig. 3 and described mathematically by Eq. 1. Each layer may have a different number of neurons, and even a different transfer function. The most used transfer functions are the sigmoidal, the hyperbolic tangent and the linear transfer functions. More details about neural networks can be found in the works of Chen et al. *(6)* and Chang and Hung *(13)*. An appropriate architecture would help to achieve higher model accuracy.

$$\hat{y}_1 = g_1[\delta,\theta] = F_1\left[\sum_{j=1}^{3} W_{1j} f_j\left(\sum_{l=1}^{4} w_{jl}\delta_l + w_{j0}\right) + W_{10}\right] \qquad (1)$$

In Eq. 1, θ specifies the parameters vector, which contains all the adjustable parameters of the network; i.e., the weights and biases $\{w_{j,l}\ W_{1,j}\}$.

In this work, conversion and productivity were each one modeled with a MLPNN with four inputs (inlet substrate concentration, S_0, cell recycle rate, R, residence time, t_r, and flash recycle rate, r) and one hidden layer.

The appropriate number of nodes to include in the hidden layer was addressed with the cross-validation technique in order to avoid model over-fitting. This technique splits the sample data into a training sample and a validation sample. Then MLPNN with different numbers of hidden nodes are trained with the training sample, and their performance monitored with the validation sample in terms of the lowest normalized mean square error *(14)*, given by Eq. 2. In addition, the quality of the prediction of the neural models can be also characterized using the correlation coefficient *(15)*, given by Eq. 3.

The Normalized Mean Square Error (NMSE):

$$\mathrm{NMSE} = \frac{1}{\sigma^2 np} \sum_{n=1}^{np} (d_p - x_p)^2 \tag{2}$$

where σ^2 denoted the sample variance of the desired outputs in the test set, x_p and d_p are, respectively, the network and desired outputs and np is the number of patterns tested.

The Correlation Coefficient (COR):

$$\mathrm{COR} = \left(1 - \frac{\mathrm{SEE}}{\mathrm{S}\tau\tau}\right) 100 \tag{3}$$

where and $\mathrm{SEE} = \sum_{p=1}^{np} (d_p - x_p)^2$ and $\mathrm{S}\tau\tau = \sum_{p=1}^{np} (d_p - \bar{d}_p)^2$

In the present work, one half of the data sample was used for training and the other half was intended for validation. The parameters were adjusted using the Levenberg–Marquardt algorithm in Matlab's neural network toolbox.

After the neural networks for conversion and productivity were trained and validated, their parameters were fixed, and the neural network models operated in parallel (Fig. 4) to simulate a function, *f*(*conv*, *prod*), that was used as the fitness criteria in the ensuing genetic algorithm optimization. In order to accomplish this task, the neural model was written in Fortran to be used by the optimization algorithm, also written in Fortran.

Optimization by Genetic Algorithms

GA is used in this work to optimize a given objective function *f* over a given search space. A population of individuals undergoes some artificial Darwinian evolution (genetic inheritance and Darwinian strife for survival) based on the fitness of each individual. The fitness of an individual is directly related to the value of the objective function of this individual. The manipulation is done by the genetic operators that work on the chromosomes, in which the parameters of possible solutions are encoded. In each generation of the GA, the new solutions replace the solutions in the population that are selected for deletion. The GA considered in the present article is based on the freeware versions written in Fortran of a real-coded genetic algorithm (RGA) developed by Yedder *(16)* and a binary-coded genetic

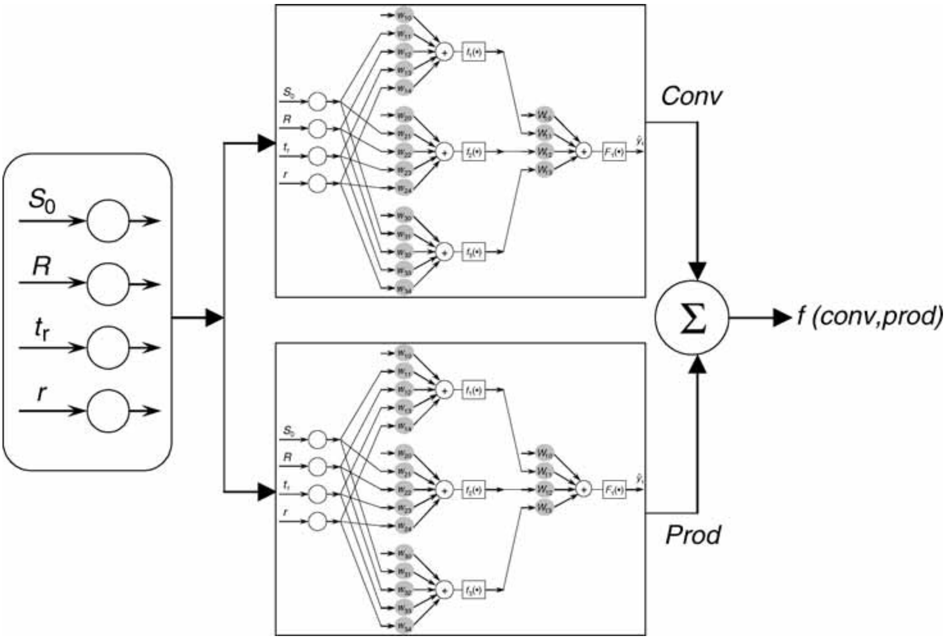

Fig. 4. Combination of the neural network models (hybrid model) for the optimization of the fermentation process.

algorithm (BGA) developed by Carroll (17), adapted to the specific needs of the process optimization. These algorithms are well suited to large-scale problems because they do not demand the computation of any Hessian matrix or its inverse, thus having relatively small memory and processing requirements. Also, it ensures the convergence of the optimization procedure (with a second-order rate).

The formulation of objective functions is one of the crucial steps in the application of optimization to a practical problem. In this work, the objective function was formulated as a nonlinear programming of the following type:

$$\begin{aligned} &\text{Optimize } f(\vec{x}) \\ &\text{subject to } g_i(\vec{x}) \leq 0, i = 1, \ldots, p, \\ &\qquad h_j(\vec{x}) = 0, j = p + 1, \ldots, m, \\ &\qquad l_k \leq x_k \leq u_k, k = 1, .., n, \end{aligned} \tag{4}$$

where x is a vector of n decision variables $(x_1, \ldots, x_n)$, $f(x)$is the objective function, $g_i(\vec{x})$ is the ith inequality constraints, and $h_j(\vec{x})$ is the jth equality constraints. The l_k and u_k are specified lower and upper bounds on the variables with $l_k \leq u_k$.

A general way to deal with constraints, whatever the optimization method, is by penalty function method (18). This technique is the most common approach in the genetic algorithms community, but there are no general guidelines on designing. Some suggestions for genetic algorithms are given in *(19)*. The penalty method of nonlinear programming transforms a constrained problem into a sequence of unconstrained problem. The method used in this work, belong to the second category of constrained handling methods described by *(19)*. In this approach, for handling inequality constraints, the penalty function $P(\vec{x})$ is defined as the sum of the objective function $f(x)$ and a penalty term which depends on the constraints violation $g_i(x)$ and $h_j(x)$:

$$\left.\begin{array}{ll} \text{Optimize} & f(\overset{r}{x}) \\ \text{Subject to} & g_i(\overset{r}{x}) \leq 0, i = 1, \ldots, p, \\ & h_j(\overset{r}{x}) = 0, j = p + 1, \ldots, m, \\ & l_k \leq x_k \leq u_k, k = 1, \ldots, n, \end{array}\right\} \text{Optimize} : P(f, g_i, h_j, C) \qquad (5)$$

where $P(f, g_i, h_j, C)$ is a penalty function, and C is a positive penalty parameter. After the penalty function is formulated, it is minimized for a series of values of increasing C-values, which force the sequence to approach the optimum of the constrained problem.

The most significant advantages of the GA are that it avoids the initial guess selection problem and provides a systematic scanning of the whole population and several acceptable local solutions. Thus, it is important to properly define constraints on the possible parameter values. First, constraints limit the search space and thus speed up the optimization. Second, physically impossible values for the parameters are avoided, which improves the reliability of the approach. Additional constraints depend on the type of model that is used. In this work the parameters satisfy such conditions as: $80 < S_0 < 280$ kg/m^3, $0.2 < R < 0.5$, $1 < t_r < 2.5$ h and $0.2 < r < 0.6$. Costa et al. *(4)* already used these values successfully in a previous study.

Results

Choice of the Neural Network Architecture

The number of hidden nodes was varied from 2 to 8 and, using the cross-validation criterion, the network with three hidden nodes was found to present the lowest normalized mean square error (Eq. 2) for the validation sample. The results can be seen in Fig. 5.

Productivity and conversion were then modeled both with a MLPNN with 19 scalar parameter, which is calculated using Eq. 6.

$$\text{Number of parameters} = (n_i + 1)n_h + (n_h + 1)n_o \qquad (6)$$

where $n_i = 4$, $n_h = 3$ and $n_o = 1$ are the number of neurons in the input, hidden, and output layers, respectively.

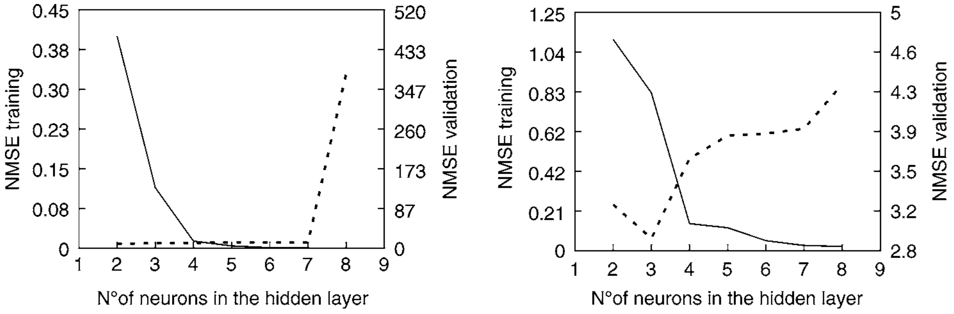

Fig. 5. Training (solid line) and validation (dashed line) error as functions of the number of neurons in the hidden layer: **(A)** Productivity and **(B)** conversion.

Table 3 shows the quality of the prediction for productivity and conversion for training and validation. As can be seen, in both cases by increasing the number of hidden neurons, the training accuracy improves, as indicated by the smaller NMSE and COR values approaching 1. However, using the cross-validation criterion, three neurons in the hidden layer appeared to be the optimal architecture to prevent overfitting.

Computation of an Optimal Solution Using Genetic Algorithms and Deterministic Model

The optimization was conducted with the deterministic steady state model of the process, which can be found in Costa et al. *(11)*. It consists of the steady state mass and energy balances for the fermentor and all the other process units (*see* Fig. 1). The equations that describe productivity and conversion can also be found in Costa et al. *(11)*.

The optimization problem is to maximize productivity. Thus, the objective function can be formulated as follows:

$$\text{Maximize } prod \tag{7}$$

subject to the equality constraints described by the steady state mass and energy balances for the fermentor and to the inequality constrains:

$$Conv > 0.99 \tag{8}$$

$$28 < T < 40^{\circ}\text{C} \tag{9}$$

$$80 < S_0 < 280 \text{ kg/m}^3 \tag{10}$$

$$0.2 < R < 0.5 \tag{11}$$

$$1.0 < t_r < 2.5 \text{ h} \tag{12}$$

$$0.2 < r < 0.6 \tag{13}$$

Because the objective in Yedder's real-coded genetic algorithm driver *(16)* is to minimize the objective function, it is necessary to map the

Table 3
Normalized Mean Square Error (NMSE) and Correlation Coefficient (COR) for Productivity and Conversion

	NMSE				COR			
	Productivity (kg/m^3 · h)		Conversion		Productivity (kg/m^3 · h)		Conversion	
Neurons	Training	Validation	Training	Validation	Training	Validation	Training	Validation
2	0.4	10	1.1	3.2	99.06	97.59	98.72	95.99
3	0.1	11	0.8	2.9	99.33	97.11	99.13	92.53
4	0.01	11	0.1	3.6	99.96	96.76	99.68	92.44
5	0.004	13.2	0.1	3.9	99.97	96.48	99.87	80.31
6	0.001	12.5	0.05	3.9	99.99	96.18	99.88	91.98
7	0.0005	12	0.03	3.9	99.99	96.16	99.95	73.19
8	0.0003	380.1	0.02	4.3	99.99	16.26	99.97	82.82

formulated problem (Eqs. 7–13) into unconstrained objective function given by Eq. 14, following the penalty function method for handling inequality constraints.

The equality constraints were handled by converting them as inequality constraints as $g_i(\vec{x}) = \lambda - |h_j(\vec{x})| \geq 0$ for all j, where λ (a small positive value) is set to 10^{-3}, in order to allow some room for the search algorithm to work on.

$$\min P(\vec{x},R) = f(\vec{x}) + R_1 |g_1(\vec{x})|^2 + R_2 \left(\sum_{j=1}^{5} (10^{-3} - |h_j(\vec{x})|)^2 \right) \tag{14}$$

where $f(x) = 1/|prod|$, $g_1 = 0.99 - |conv|$ and $h_j(\vec{x})$ are the steady state mass and energy balances for the fermentor. The parameters $(\bar{x})$ satisfy such conditions as: 80 kg/m^3 < S_0 < 280 kg/m^3, 0.2 < R < 0.5, 1 h < t_r < 2.5 h, and 0.2 < r < 0.6.

The optimization variables are: concentration of viable cells, X_v; dead cells, X_d; substrate, S; product, P; and Temperature, T; as well as the variables used by Costa et al. *(11)* (S_0, R, t_r, and r).

With the best penalty parameters, $R_1 = 0.02$ and $R_2 = 1250$ for constraints, the calculated values were $S_0 = 216.29$ kg/m^3, $R = 0.41$, $t_r = 1.77$ h, and $r = 0.48$. The optimal solution to this problem, using RGA, is: productivity = 13.1 kg/(m^3 · h) and conversion = 0.99.

Carroll's binary-coded genetic algorithm *(17)* maximizes the objective function. In consequence, the problem (Eqs. 7–13) was formulated into the penalty function given by Eq. 15.

$$\max P(\vec{x},R) = f(\vec{x}) + R_1 \left(|g_1(\vec{x})|^2 \right)^{-1} + R_2 \left(\sum_{j=1}^{5} (10^{-3} - |h_j(\vec{x})|)^2 \right)^{-1} \tag{15}$$

where $f(x) = |prod|$, $g_1(\vec{x}) = 0.99 - |conv|$ and $h_j(\vec{x})$ are the steady state mass and energy balances for the fermentor. Here, the parameters (x) also satisfy conditions given by Eqs. 9–13.

Using BGA the following values are obtained: $S_0 = 135.86$ kg/m^3, $R = 0.28$, $t_r = 1.96$ h, and $r = 0.40$. The optimal solution to this problem with the best penalty parameter, $R_1 = 97$ and $R_2 = 1$ for constraints, are: productivity = 13.1 kg/(m^3 · h) and conversion = 0.99.

Kalil et al. *(3)* optimized the industrial conventional process designed by Andrietta and Maugeri *(10)* and obtained productivity of 12 kg/(m^3 · h), conversion of 0.99 and yield of 0.86. Although in the formulated problem only one inequality restriction was considered (conv > 0.99), the values obtained for yield: 0.87 and 0.9 for RGA and BGA, respectively, were better than the optimized conventional process of Kalil et al. *(3)*. The optimization using genetic algorithms led to productivity a little higher than when the extractive process was optimized using successive quadratic programming by Costa et al. *(4)*.

Computation of an Optimal Solution Using Genetic Algorithms and Neural Network Model

Because the objective of the optimization study was to maximize productivity, although maintaining a high conversion, the neural network models were structured (*see* Fig. 4) to explicitly estimate these output variables from the inlet variables of interest (S_0, R, t_r, and r). This structure models the mass and energy balances for the fermentor and all the other process units. After the solution of the optimization problem, the optimal values of S_0, R, t_r, and r were used in the deterministic model to determined if the MLPNN predictions for optimal productivity and conversion present deviations from the values calculated by the deterministic model.

The optimization problem is the function *f(prod, conv)* transformed in the penalty function form given by Eqs. 16 and 17 for RGA and BGA, respectively:

$$\min P(\overset{r}{x},C) = f(\overset{r}{x}) + C_1\left|g_1(\overset{r}{x})\right|^2 \tag{16}$$

where $f(\bar{x}) = 1/\,|\,prod\,|$ and $g_1(\vec{x}) = 0.99 - conv\,|$.

In this problem, such as in the previous case, a constraint in conversion is included ($conv > 0.99$). With the best penalty parameter, $C_1 = 260$ for constraint, the calculated values were $S_0 = 131.93$ kg/m^3, $R = 0.28$, $t_r = 1.20$ h, and $r = 0.54$.

The optimal solution to this problem, using RGA, is: productivity = 13 kg/(m^3 · h) and conversion = 0.99. Using the values calculated for S_0, R, t_r, and r in the deterministic model, productivity obtained was of 13.2 kg/(m^3 · h) and conversion of 0.99.

$$\max P(\overset{r}{x},C) = f(\overset{r}{x}) + C_1\left(\left|g_1(\overset{r}{x})\right|^2\right)^{-1} \tag{17}$$

where $f(\bar{x}) = \,|\,prod\,|$ and $g_1(\vec{x}) = 0.99 - \,|\,conv\,|$.

Using BGA the following values are calculated: $S_0 = 161.4$ kg/m^3, $R = 0.44$, $t_r = 1.60$ h, and $r = 0.37$. The optimal solution to this problem with the best penalty parameter, $C_1 = 97$ for constraint, is: productivity = 13 kg/(m^3 . h) and conversion = 0.99. Using the values calculated for S_0, R, t_r, and r in the deterministic model it was found productivity of 13.0 kg/(m^3 · h) and conversion of 0.99.

It is possible to notice that, when the optimization problem is solved using genetic algorithms and the neural network models, the values of temperature and concentrations in the fermentor do not have to be considered as optimization variables and so, the number of the optimization variables is smaller than using deterministic model.

Table 4 shows the optimization results of productivity and conversion calculated by the RGA and BGA on the deterministic and MLPNN

Table 4
Optimization Variable Values Obtained Through RGA and BGA on the Deterministic and MLPNN Models

	Neural-RGA	Neural-BGA	Deterministic-RGA	Deterministic-BGA
S_0 (kg/m^3)	131.93	161.4	216.29	135.86
R	0.28	0.44	0.41	0.28
t_r (h)	1.20	1.6	1.77	1.96
R	0.54	0.37	0.48	0.4
X_v (kg/m^3)	–	–	36.66	24.26
X_d (kg/m^3)	–	–	3.18	2.69
S (kg/m^3)	–	–	2.39	2.06
P (kg/m^3)	–	–	39.31	37.99
T (°C)	–	–	32.98	35.18
Prod (kg/m^3•h)	13.2	13	13.1	13.1
Conv	0.99	0.99	0.99	0.99

models. It also shows the comparison among the optimization variables. The results of the optimization methods are expected to lead to different parameters values because they are numerical procedures with expected imprecision in finding the global optimum. Different mathematical models may lead to different predictions, however, a well-trained neural network is able to capture the process dynamic behavior. It is worthwhile mentioning that all the parameters values in Table 4 are within the accepted range for industrial process operation and this was taken into account in the definition of the constraints for the optimization problem.

Discussion

This work focused on modeling and optimization on the level of process operation of an extractive alcoholic fermentation. The computational intelligence techniques are sought to efficiently combine all available knowledge and to direct the development toward an improved process operation strategy. The genetic algorithms have shown a good capability to determine the best conditions of conversion and productivity using both the deterministic and the neural network models.

The NN model was developed using a data set generated from the deterministic model, which means the data are not noisy. However, this may not be considered a limitation, because the procedure may be used when experimental data are available. It can be observed that the values for productivity and conversion obtained using the MLPNN models are similar to that obtained using the deterministic model. MLPNN modeling is a powerful and flexible tool and its use in the alcoholic fermentation process is advantageous because updating the neural network parameters is a simpler procedure than re-estimating kinetic parameters of the deterministic models. Frequent re-estimation of kinetic parameters (or in the case of this work updating the neural network parameters) is necessary owing to changes in the dynamic behavior caused by fluctuations in the quality of the raw material, changes in microorganism metabolism, and variation of the dominant yeast strains present in the fermentation process, among other factors.

Normally the optimization methods based on GA require a large computer burden compared with the conventional optimization methods. Neural network models led to a mathematical representation of lower order than the deterministic approach, which intuitively makes the optimization procedure based on hybrid algorithms (NN-RGA and NN-BGA) to be significantly quicker. In fact, the neural network model requires only four optimization variables; whereas the deterministic model needs nine.

Experimental design is used to obtain data for the training of the neural network in order to guarantee that the region of interest is covered by the training data. This is a very important issue to be addressed as neural network models have no (or very limited) extrapolation properties. The penalty function optimization using RGA and BGA was done in terms of constraint violation to check the feasibility of the solution. In the case of infeasible solution, the one having smaller constraint violation is preferred. In real coded genetic algorithm, the decision variables are coded in real numbers unlike the binary numbers as in the case of binary coded genetic algorithms. The most important feature of the RGAs is their capacity to exploit local continuities, and the corresponding one of the BGA is their capacity to exploit the discrete similarities *(20)*. According to Goldberg *(21)*, BGA are less efficient when applied to multi-dimensional, high precision or continuous problems. The bit strings can become very long and the search space blows up. In RGA the chromosome/individual is simply a vector in which each real-valued element (gene) stands for an unknown parameter of the model.

Although the best result was the one obtained using the rigorous model, the values for productivity and conversion obtained using the MLPNN model are acceptable and can partly eliminate the difficulties of having to specify completely the structure of an alcoholic fermentation model. It is especially promising in situations in which rigorous model is excessively difficult computationally, time-consuming or costly.

Acknowledgment

The authors acknowledge FAPESP (process number 03/03630-3) for financial support.

Nomenclature

F	Feed stream flow rate (m^3/h)
F_c	Cell suspension flow from centrifugue (m^3/h)
F_{c1}	Cell suspension flow to treatment tank (m^3/h)
F_E	Light phase flow rate to flash tank (m^3/h)
F_L	Liquid outflow from the vacuum flash tank (m^3/h)
F_{LR}	Liquid phase recycling flow rate (m^3/h)
F_{RS}	Liquid phase flow to rectification column (m^3/h)
F_O	Fresh medium flow rate (m^3/h)
F_R	Cell recycling flow rate (m^3/h)
F_V	Vapor outflow from the vacuum flash tank (m^3/h)
F_W	Water flow rate (m^3/h)
P	Product concentration into the fermentor (kg/m^3)
P_F	Feed product concentration (kg/m^3)
P_{LR}	Product concentration in the light phase from centrifuge (kg/m^3)
P_R	Product concentration in the cells recycle (kg/m^3)
P_V	Product concentration in the vapor phase from the flash tank (kg/m^3)
$R = F_R/F$	Cells recycle rate
$r = F_{LR}/F_L$	Flash recycle rate
S	Substrate concentration into the fermentor (kg/m^3)
S_F	Feed substrate concentration (kg/m^3)
S_{LR}	Substrate concentration in the light phase from centrifugue (kg/m^3)
S_O	Inlet substrate concentration (kg/m^3)
S_R	Substrate concentration in the cells recycle (kg/ m^3)
T	Temperature into the fermentor (°C)
T_F	Feed temperature (°C)
T_{LR}	Light phase temperature (°C)
T_O	Inlet temperature of the fresh medium (°C)
T_R	Cells recycle temperature (°C)
T_V	Temperature of vapor from the flash tank (°C)
T_W	Water temperature (°C)
X_c	Biomass concentration in the heavy phase from centrifugue (kg/m^3)
X_d	Dead biomass concentration into the fermentor (kg/m^3)
X_E	Biomass concentration in the light phase flow rate to flash tank (kg/m^3)
X_F	Feed biomass concentration (kg/m^3)
X_{LR}	Biomass concentration in the light phase from centrifugue (kg/m^3)
X_R	Cell recycling concentration (kg/m^3)
$X_t = X_v + X_d$	Total biomass concentration into the fermentor (kg/m^3)
X_r	Viable biomass concentration into the fermentor (kg/m^3)

References

1 Goldemberg, J., Coelho, S. T., Nastari, P. N., and Lucon, O. (2004), *Biomass Bioenerg.* **26,** 301–304.

2 Parekh, S., Vinci, V. A., and Strobel, R. J. (2000), *Appl. Microbiol. Biol.* **54,** 287–301.

3 Kalil, S. J., Maugeri, F., and Rodrigues, M. I. (2000), *Process Biochem.* **35,** 539–550.

4 Costa, A. C. and Maciel Filho, R. (2004), *App. Biochem. Biotech.* **114/1–3,** 485–496.

5 Schügerl, K. (2001), *J. Biotechnol.* **85,** 149–173.

6 Chen, L., Nguang, S. K., Chen, X. D., and Li, X. M. (2004), *Biochem. Eng. J.* **22,** 51–61.

7 Nandi, S., Badhe, Y., Lonari, J., et al. (2004), *Chem. Eng. J.* **97,** 115–129.

8 Ahmad, A. L., Azid, I. A., Yusof, A. R., and Seetharamu, K. N. (2004), *Comput. Chem. Eng.* **28,** 755–766.

9 Silva, F. L. H., Rodrigues, M. I., and Maugeri, F. (1999), *J. Chem. Tech. Biotechnol.* **74,** 176–182.

10 Andrietta, S. R. and Maugeri, F. (1994), In *Adv. Bioprocess Engng.* 47–52.

11 Costa, A. C., Atala, D. I. P., Maciel Filho, R., and Maugeri Filho, F. (2001), *Process Biochem.* **37-2,** 125–137.

12 Montgomery, Douglas C. (1997), *Design and Analysis of Experiments*, 4th ed., John Wiley and Sons, New York.

13 Chang, J. and Hung, B. (2002), *Ind. Eng. Chem. Res.* **41,** 2716–2727.

14 Geva, A. B. (1998), *IEEE Trans. neural net.* **95,** 1471–1482.

15 Milton, J. S. and Arnold, J. C. (1990), *Introduction to probability and statistics*, McGraw Hill, New York.

16 Yedder, R. B. (2002), Ph.D. thesis, ENPC, Paris, France.

17 Carroll, D. L. (1998), FORTRAN Genetic Algorithm Driver, last accessed date 04-04-2005. University of Illinois, USA (carroll@cuaerospace.com).

18 Edgar, T. F. and Himmelblau, D. M. (1998), *Optimization of Chemical Processes*, 2nd ed., McGraw-Hill, New York.

19 Michalewicz, Z. and Schoenauer, M. (1996), *Evol. Comp.* **4,** 1–32.

20 Herrera, F., Lozano, M., and Verdegay, J. L. (1998), *A. I. Rev.* **12,** 265–319.

21 Goldberg, D. E. (1989), *Genetic algorithms in search, optimization and machine learning.* Addisson-Wesley Reading, Massachussets.