Drug Development and Industrial Pharmacy, 35:396–407, 2009
Copyright © Informa UK, Ltd.
ISSN: 0363-9045 print / 1520-5762 online
DOI: 10.1080/03639040802422088

informa
healthcare

Application of Artificial Intelligent Tools to Modeling of Glucosamine Preparation from Exoskeleton of Shrimp

Hadi Valizadeh
Department of Pharmaceutics, Faculty of Pharmacy, Tabriz University of Medical Sciences, Tabriz, Iran and Drug Applied Research Center, Tabriz University of Medical Sciences, Tabriz, Iran, and Zahravi Pharmaceutical Co., Tabriz, Iran

Mohammad Pourmahmood
Control Engineering Department, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

Javid Shahbazi Mojarrad
Department of Medicinal Chemistry, Faculty of Pharmacy, Tabriz University of Medical Sciences, Tabriz, Iran and Biotechnology Research Center, Tabriz University of Medical Sciences, Tabriz, Iran

Mahboob Nemati
Department of Pharmacognosy and Food Sciences, Faculty of Pharmacy, Tabriz University of Medical Sciences, Tabriz, Iran and Drug Applied Research Center, Tabriz University of Medical Sciences, Tabriz, Iran

Parvin Zakeri-Milani
Department of Pharmaceutics, Faculty of Pharmacy, Tabriz University of Medical Sciences, Tabriz, Iran and Research Center for Pharmaceutical Nanotechnology, Tabriz University of Medical Sciences, Tabriz, Iran

The objective of this study was to forecast and optimize the glucosamine production yield from chitin (obtained from Persian Gulf shrimp) by means of genetic algorithm (GA), particle swarm optimization (PSO), and artificial neural networks (ANNs) as tools of artificial intelligence methods. Three factors (acid concentration, acid solution to chitin ratio, and reaction time) were used as the input parameters of the models investigated. According to the obtained results, the production yield of glucosamine hydrochloride depends linearly on acid concentration, acid solution to solid ratio, and time and also the cross-product of acid concentration and time and the cross-product of solids to acid solution ratio and time. The production yield significantly increased with an increase of acid concentration, acid solution ratio, and reaction time. The production yield is inversely related to the cross-product of acid concentration and time. It means that at high acid concentrations, the longer reaction times give lower production yields. The results revealed that the average percent error (PE) for prediction of production yield by GA, PSO, and ANN are 6.84, 7.11, and 5.49%, respectively. Considering the low PE, it might be concluded that these models have a good predictive power in the studied range of variables and they have the ability of generalization to unknown cases.

Keywords chitin; glucosamine; production yield; artificial neural networks; genetic algorithm; particle swarm optimization

Address correspondence to Hadi Valizadeh, Department of Pharmaceutics, Faculty of Pharmacy, Tabriz University of Medical Sciences, Daneshgah Street, Tabriz 51664, Iran. E-mail: valizadeh@tbzmed.ac.ir

INTRODUCTION

Glucosamine in the human body participates in the structure of cartilage and works to stimulate joint function and repair. It has been proven effective in numerous scientific trials for easing osteoarthritis pain, aiding in the rehabilitation of cartilage, renewing synovial fluid, and repairing joints that have been damaged from osteoarthritis (Hauselman, 2001; Mankin, Brandt, & Shulman, 1986). The preparation of glucosamine hydrochloride from chitin is a simple hydrolysis reaction. During this reaction, chitin is deacetylated and depolymerized to glucosamine hydrochloride in the presence of hydrochloric acid solution. The variable factors that can influence the yield of this reaction

are acid concentration, acid solution to solid ratio (vol/wt), and time (Mojarrad, Nemati, Valizadeh, Ansarin, & Bourbour, 2007).

Genetic algorithm (GA) and particle swarm optimization (PSO), as tools of artificial intelligence methods, are two efficient methods that, inspiring of nature, can predict and optimize complex process. Another application of intelligent tools is artificial neural networks (AANs), which are known as function approximators. The development of ANNs was inspired by the characteristics and capabilities of the human nervous system. Thus, an ANN, in general, consists of interconnected computing units that are geometrically organized in one, two, or three dimensions. The individual computing units can be called "neurons" after the complex cells in the central nervous system that are the basic computing elements of the human brain. The multilayered perceptron (MLP) neural networks with the error back propagation (EBP) training algorithm are one of the most popular ANN architectures as they are simple in implementation and are capable of performing any arbitrary mapping to a desired accuracy, provided a sufficient number of hidden units are implemented and a set of connection weights that perform the desired mapping exists.

In previous work (Mojarrad et al., 2007), using response surface analysis, a model was introduced to represent the production yield of glucosamine hydrochloride as a function of the acid concentration (%), acid solution to solid ratio (vol/wt), and reaction time. The objective of the present study is to apply artificial intelligence methods (GA, PSO, and MLP) to explore whether these methodologies can help us to approximate and predict the glucosamine production yield from the reactions with different conditions with better results.

METHODOLOGY

Genetic Algorithm

GA was proposed by Holland (1975) and was inspired by Darwinian evolution and Mendelian genetics (Randy & Sue, 2004). GA is a general-purpose stochastic search method simulating natural selection and evaluation in the biological world. Generally, a GA works as follows: A population of individuals is initialized where each individual represents a potential solution to the problem at hand. Individuals in GA are called *chromosomes*. Each chromosome consists of a string of cells called *genes*. The value of each gene is called *allele*. The quality of each solution is evaluated using a *fitness function*. A selection process is applied during each iteration of a GA in order to form a new population. The selection process is biased toward the fitter individuals to ensure that they will be part of the new population. Individuals are altered using unary transformation (mutation) and higher order transformation (crossover). This procedure is repeated until convergence is reached. The best solution found is expected to be a *near-optimum* solution. The major parameters of a GA are discussed as follows.

Solution Representation

Binary representation is often used in GAs where each gene has a value of either 0 or 1. Other presentations are proposed, for example, floating point representations, integer representations, gray-coded representations, and matrix representation. It has been demonstrated that floating point representations are faster, more consistent, and have higher precision than binary representation (Michalewicz & Fogel, 2000).

Fitness Function

A key element in GA is the selection of a fitness function that accurately quantifies the quality of candidate solutions. A good fitness function enables the chromosomes to effectively solve a specific problem. Both the fitness function and solution representation are problem-dependent parameters. A poor selection of these two parameters will drastically affect the performance of GAs. One problem related to fitness function that may occur when GAs are used to optimize combinatorial problems is the existence of points in the search space that do not map to feasible solutions. One solution to this problem is the addition of a *penalty function* term to the original fitness function so that chromosomes representing infeasible solutions will have a low fitness score and, as such, will disappear from the population (Fletcher, 2000).

Selection

Another key element of GAs is the selection operator that is used to select chromosomes (called *parents*) for mating in order to generate new chromosomes (called *offspring*). In addition, the selection operator can be used to select elitist individuals. The selection process is usually biased toward fitter chromosomes. Selection methods are used as mechanisms to focus the search on apparently more profitable regions in the search spaces. Examples of well-known selection approaches are follows:

1. *Rolette wheel selection*. Parent chromosomes are probabilistically selected based on their fitness. The fitter the chromosome, the higher the probability that it may be chosen for mating.
2. *Rank selection.* Rank selection sorts the chromosomes according to their fitness and bases selection on the rank order of the chromosomes, and not on the absolute fitness values.
3. *Tournament selection.* In this more commonly used approach, a set of chromosomes is randomly chosen. The fittest chromosome from the set is then placed in a mating pool. This process is repeated until the mating pool contains a sufficient number of chromosomes to start the mating process.
4. *Elitism*. In this approach, the fittest chromosome, or a user-specified number of best chromosomes, is copied into the new population. The remaining chromosomes are then chosen using any selection operator. Because the best solution is never lost, the performance of GA can significantly be improved.

Crossover

Crossover is the main explorative operator in GAs. Crossover occurs with a user-specified rate X_{rate}. At each iteration, X_{rate}% of worse parents are eliminated to be replaced by new offspring. X_{rate} is problem dependent with typical values in the range from 0.4 to 0.8. The four main crossover operators are as follows:

1. *Single-point crossover*. In this approach, a position is randomly selected at which the parents are divided into two parts. The parts of the two parents are then swapped to generate two new offspring.
2. *Two-point crossover*. In this approach, two positions are randomly selected. The middle parts of the two parents are then swapped to generate two new offspring.
3. *Uniform crossover*. In this approach, alleles are copied from either the first parent or the second parent with some probability, usually set to 0.5.
4. *Arithmetic crossover*. In this approach, which is used for floating point representations, the number of offspring is calculated as the arithmetic mean of the parents, as described below.

$$x_{\text{offspring A}} = \text{rand} \times x_{\text{parent A}} + (1 - \text{rand}) \times x_{\text{parent B}}$$

$$x_{\text{offspring B}} = \text{rand} \times x_{\text{parent B}} + (1 - \text{rand}) \times x_{\text{parent A}}$$

Mutation

In GAs, mutation is considered to be a background operator, mainly used to explore new areas in the search space and to add diversity (contrary to selection and crossover, which reduce diversity) to the population of chromosomes in order to prevent being trapped in a local optimum. Mutation is applied to the offspring chromosomes after crossover is performed. In a binary-coded GA, mutation is done by inverting the value of each gene in the chromosome according to a user-specified probability, which is called the *mutation probability*, *pm*. This probability is problem dependent. Mutation occurs infrequently both in nature and in Gas; hence, a typical value for *pm* is 0.01. However, a better value for *pm* is the inverse of the number of genes in a chromosome (i.e., chromosome size) (Goldberg, 1989). The pseudo code for a GA is given in Appendix A.

Particle Swarm Optimization

A particle swarm optimizer is a population-based stochastic optimization algorithm modeled after the simulation of the social behavior of bird flocks (Rardin, 1998). PSO is similar to GA in the sense that both approaches are population-based and each individual has a fitness function. Furthermore, the adjustments of the individuals in PSO are relatively similar to the arithmetic crossover operator used in GA. However, PSO is influenced by the simulation of social behavior rather than the survival of the fittest. Another major difference is that in PSO each individual benefits from its history, whereas no such mechanism exists in GA. In a PSO system, a swarm of individuals (called *particles*) fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself (i.e., its own experience) and the position of the best particle in its neighborhood. When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best particle, and the resulting algorithm is referred to as a *gbest* PSO. When smaller neighborhoods are used, the algorithm is generally referred to as a *lbest* PSO. The performance of each particle (i.e., how close the particle is from the global optimum) is measured using a fitness function that varies depending on the optimization problem.

The global optimizing model proposed by Shi and Eberhart (1998) is as follows:

$$v_{i+1} = w \times v_i + \text{RAND} \times c_1 \times (P_{\text{best}} - x_i) + \text{rand} \times c_2 \times (G_{\text{best}} - x_i)$$

$$x_{i+1} = x_i + v_{i+1}$$

where v_i is the *velocity* of particle i, x_i is the particle position, w is the inertial weight, c_1 and c_2 are the positive constant parameters, RAND and rand are the random functions in the range [0,1], P_{best} is the best position of the ith particle, and G_{best} is the best position among all particles in the swarm. The inertia weight term, w, serves as a memory of previous velocities. The inertia weight controls the impact of the previous velocity: a large inertia weight favors exploration, whereas a small inertia weight favors exploitation (Kennedy & Eberhart, 1995; Shi & Eberhart, 1998). As such, global search starts with a large weight and then decreases with time to favor local search over global search (Eberhart & Shi, 1998).

It is noted that the second term in equation (1) represents cognition, or the private thinking of the particle when comparing its current position to its own best. The third term in equation (1), on the other hand, represents the social collaboration among the particles, which compares a particle's current position with that of the best particle (Kennedy, 1997). Also, to control the change of particles' velocities, upper and lower bounds for velocity change are limited to a user-specified value of $V_{\max}$. Once the new position of a particle is calculated using equation (2), the particle, then, flies toward it (Shi & Eberhart, 1998). As such, the main parameters used in the PSO technique are the population size (number of birds); number of generation cycles; the maximum change of a particle velocity $V_{\max}$; and w.

The pseudo code for the basic PSO is given in Appendix B. The process is initialized with a group of random particles (solutions). The ith particle is represented by its position as a point in search space. Throughout the process, each particle moves about the cost surface with a velocity. Then the particles

update their velocities and positions based on the best solutions. This process continues until stop condition(s) is satisfied.

For both GA and PSO algorithms, stop conditions could be as follows:

a. Sufficient good solution has been found.
b. The maximum number of iterations has been reached.
c. No improvement has been made for a number of iterations in cost decreasing.

Artificial Neural Networks

As presented schematically in Figure 1, an ANN (Fausett, 1994; Hagan, Demuth, & Beale, 1996; Nelson & Illingworth, 1991; Simpson, 1990) involves the nodes that are known as neurons. The neurons are organized into a sequence of layers and connected to each other by using variable connection weights. The first layer is the input layer with one node for each variable or feature of the data. The last layer is the output layer consisting of one node for each variable to be investigated. A series of one or more hidden layer(s) consisting of a number of nodes is placed between input and output layers, which are responsible for learning process of the network. MLP networks are most often used to analyze nonlinear multivariable data. In these networks, signals are propagated from the input layer. A node receives signals via connections from previous nodes (or the outside in the case of the input layer). The output of a node is determined by the transfer function and net input of the node. A popular transfer function is the sigmoid function:

$$\mathrm{Out}_j = \frac{1}{1+\mathrm{e}^{-\alpha \mathrm{Net}_j}}$$

where Net_j is defined as

$$\mathrm{Net}_j = \sum_{i=1}^{n} (\mathrm{Inp}_i \times W_{ij}) + B_j$$

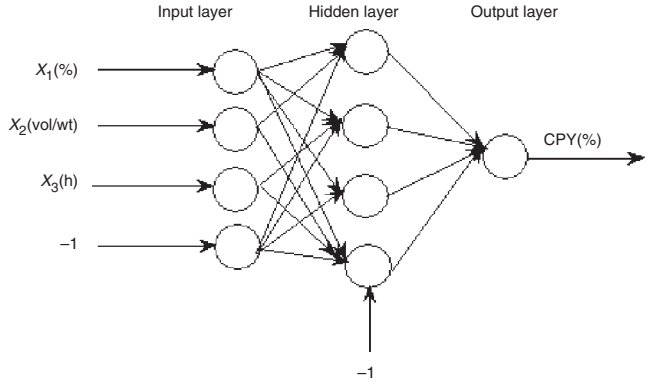

FIGURE 1. Structural organization of the multilayered perceptron network.

where Out_j is the output from jth neuron, Inp_i is the input to jth node from a previous layer with n nodes, and W_{ij} is the respective weights, the B_j is a bias term, and α is the slope of the transfer function.

The ANN learns from data that are presented to the network as training set. The goal of training process is to find the optimum weights, and the process starts with random connection weights. The computed output (O_{pm}) is compared with the target value (T_{pm}), and an error term $(T_{pm} - O_{pm})^2$ is determined. The total sum square error (TSSE) is used as a criterion for finalizing the learning process and computed using the following equation:

$$\mathrm{TSSE} = \frac{1}{2} \sum_{p=1}^{p} \sum_{m=1}^{M} (T_{pm} - O_{pm})^2$$

where M is the number of neurons in output layer and P denotes the number of patterns, that is, the number of experimental solubility data employed in the training process of the network. The number of neurons in the hidden layer and epochs has been optimized by minimizing the TSSE term.

The EBP algorithm is used in layered feed-forward ANNs. This means that the artificial neurons are organized in layers and send their signals forward, and then the errors are propagated backwards. The network receives inputs by neurons in the input layer, and the output of the network is given by the neurons on an output layer. The EBP algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated. The idea of the EBP algorithm is to reduce this error (TSSE) until the ANN learns the training data. The training begins with random weights, and the goal is to adjust them so that the error will be minimal. The training of the network is an iterative process. One of the various learning algorithms (in this article, EBP method) is used to optimally adjust the weights of the connections between the neurons in the network in order to optimally predict (find a best fit) the sample data on which the training is performed.

Overtraining Problem

One of the problems that may occur during neural network training is called overtraining, in such a way that the error on the training set is driven to a very small value, but when new data are presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations. If the number of parameters in the network is much smaller than the total number of points in the training set, then there is little or no chance of overtraining. If more data can be easily collected to increase the size of the training set, then there will be no need to worry about techniques to prevent overtraining. One method for improving network generalization is to use a network that is just large enough to provide an adequate fit. The larger a

network is used, the more complex the functions the network can be created. If a small network is employed, it will not have enough power to overtrain the data. Usually, it is difficult to know beforehand how large a network should be for a specific application. There are two other methods for improving generalization that are implemented in the neural network literature: regularization and early stopping.

The first method for improving generalization is called regularization. This involves modifying the performance function, which is normally chosen to be the sum of squares of the network errors on the training set. In this technique, the performance function can be modified, and a routine can automatically set the optimal performance function to achieve the best generalization. It is possible to improve generalization if the performance function TSSE is modified by adding a term that consists of the sum of squares of the network weights and biases. Using the new modified performance function will cause the network to have smaller weights and biases, and this will force the network response to be smoother and less likely to overtrain.

Another method for improving generalization is called early stopping. In this technique, the available data are divided into three subsets. The first subset is the training set, which is used for computing the gradient (EBP method) and updating the network weights and biases. The second subset is the test set. The third sub set is the validation set. The error on the validation set is monitored during the training process. The validation error will normally decrease during the initial phase of training, as does the training set error. However, when the network begins to overtrain the data, the error on the validation set will typically start to increase. When the validation error increases for a specific number of epochs, the training is stopped, and the weights and biases at the minimum of the validation error are returned.

Both regularization and early stopping can ensure network generalization when properly applied. With both regularization and early stopping, it is a good idea to train the network starting from several initial conditions. It is possible for either method to fail in certain circumstances. By testing several initial conditions, we can verify robust network performance.

When the training process is ended (typically when TSSE be smaller than a specific value or when no improvement in reduction of TSSE is reached for several epochs or when training process is repeated for a specific number of epochs; by early stopping or regularization techniques for preventing overtraining), the trained network can then be used to generate predictions for unknown cases. On the other hand, one of the major advantages of ANNs is that the user need not have any knowledge of the underlying model on which input variables depend.

MATERIALS AND METHODS

Preparation of Chitin

Marine shrimp (*Metapenaeus monoceros*) were obtained from beach of Boshehr port in the Persian Gulf in July 2005. The shells of the shrimp were scraped free of loose tissue, treated with running hot water to remove soluble organics and adherent proteins, and dried in the shadow (25–30°C) for 3 days. This was done to minimize batch dissimilarities because of the adherent proteins of the shell (Mojarrad et al., 2007). The dried shell was ground and sieved with a 35 mesh (0.5 mm) sieve. The deproteinization of shells involved stirring of the shells in dilute NaOH (3.5%) with a solvent to solid ratio of 10:1 (vol/wt) for 2 h at 65°C. The residue was then collected on an 80 mesh (0.177 mm) sieve, washed to neutrality in running tap water, and filtered to remove excess moisture. The deproteinized shells were demineralized with 1 N HCl for 0.5 h at ambient temperature with constant stirring and a solvent to solid ratio of 15:1 (vol/wt). Following demineralization, the decalcified chitin was then washed and filtered as above (Mojarrad et al., 2007). Carotenoids and other pigments were extracted by absolute acetone, and the chitin residue was bleached with sodium hypochlorite solution (0.315%) for 5 min at ambient temperature with a solvent to solid ratio of 10:1 (vol/wt). The white chitin was collected, washed with water, and dried at 60°C in a forced-air oven for 4 h (Mojarrad et al., 2007; No, Meyers, & Lee, 1989; Percot, Viton, & Domard, 2003a, b; Shahidi & Synowiecki, 1991).

Preparation of Glucosamine Hydrochloride

Chitin (0.5 g) was treated with different ratios and concentrations of HCl solutions at 100°C for 1–4 h. The acid solution was added stepwise in four portions in predetermined intervals. Reaction mixtures were filtered and washed with water. The filtrate and the washings were mixed and collected in 25-mL volumetric flask. The collected solution was used for determining glucosamine hydrochloride by high-performance liquid chromatography (Mojarrad et al., 2007; Nemati, Valizadeh, Ansarin, & Ghaderi, 2007).

Statistical Design

The experimental design was a modified Box–Behnken design for three variables. The acid concentration (%) (X_1), acid solution to solid ratio (vol/wt) (X_2), and reaction time (X_3) were three independent variables considered in the preparation of glucosamine hydrochloride. The complete design consisted of 24 experimental points, which included two replications of 1–12 experiments. These experiments were carried out in random order. Obtained data were analyzed to fit the following polynomial equation to Y (production yield of glucosamine hydrochloride).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{12} X_1 X_2 + \beta_{13} X_1 X_3 + \beta_{23} X_2 X_3 \quad (1)$$

where β values are constant regression coefficients and X_i values are independent variables. GA and PSO methods were

applied to calculate constant regression coefficients (β values), optimally.

As second approach, an MLP was used to model the relationship through the input data and the output. Here, production yield of glucosamine hydrochloride was approximated by the acid concentration (%) (X_1), acid solution to solid ratio (vol/wt) (X_2), and reaction time (X_3), using an MLP. In fact, MLP acts as a black box that represents a model for the production yield of glucosamine hydrochloride for which the acid concentration (%) (X_1), acid solution to solid ratio (vol/wt) (X_2), and reaction time (X_3) are the model inputs. This nonlinear model is corresponding to the equation (1), in previous approach.

To evaluate the external predictive performance of the models, six more experiments were carried out in duplicate as a test set.

Matlab version 7 software was used for the implementation of the GA, PSO methods, and the MLP.

RESULTS AND DISCUSSION

Statistical Analysis

Preparation of glucosamine hydrochloride can be reasonably optimized by studying three variable factors, including acid concentration, acid solution to solid ratio, and reaction time. The experiments were conducted using 20–37% hydrochloric acid and 100°C to enhance the chitin hydrolysis. Three variable factors were investigated at two or three levels.

As mentioned, for statistical design, we used artificial intelligence methods in two approaches. The detailed illustrations of these processes are given in the following subsections.

GA and PSO Approaches

First, a model is modified by equation (1) that it includes seven unknown parameters. The parameters should be determined by GA and PSO to minimize a cost function as sum square error (SSE), defined in equation (2).

$$\mathrm{SSE} = \sum_{i=1}^{12} (\mathrm{OPY}_i - \mathrm{CPY}_i)^2 \qquad (2)$$

The following values were set for the parameters of the GA and PSO methods, in all simulations. Twenty chromosomes constructed the population. Because high population size causes the algorithm to be complicated and slow where low population size may make an algorithm weak which sticks in local optimum, this population size is suitable. In early stages of the algorithm, when good solutions were not found, the crossover rate (X_{rate}) was set to 0.6. After some generations, the GA found that near-optimal solution's X_{rate} was reduced gradually as more good parents survive and mate. This helps for the more usage of good parents to create excellent chromosomes. Because the search space was complicated, the diversification property of the algorithm should be increased. Therefore, the high value of mutation rate was selected to generate more random offspring that aids for searching the virgin areas. Therefore, the initial value for mutation rate was set to 0.65. Then, gradually decreasing values (starting from 0.65) were defined for the mutation operator. Finally, rank selection method was selected for selection operator. In fact, in this work, selection operator was not so important; therefore, roulette wheel or other operators could be chosen.

According to GA, the suitable number of particles of the PSO procedure was found to be 20 by trial and error. At large swarms, the algorithm was slow and had no major advantages. Particles of small swarms could not escape from search space local optimum traps. Therefore, for this work, 20 particles were well suited. Inertia weight w was set as a time-variant linear function of number of iterations. At any iteration i

$$w_{i+1} = w_{\max} - \frac{w_{\max} - w_{\min}}{N_{\mathrm{iter}}} \times i$$

where N_{iter} is the maximum number of iterations (500) and $w_{\max}$ and $w_{\min}$ are upper and lower bounds of w, which were set to 0.1 and 0.9, respectively. Acceleration coefficients, c_1 and c_2 parameters, chosen were equal to 1 and 2, respectively. This choice causes the particles move with more acceleration toward the global best point than the local best point that speeds up the algorithm.

According to the experience and trial and error, the maximum number of generations for GA method was selected to be 500. Figure 2 shows the best (elite) minimized error (SSE) by GA in 500 generations. As this figure shows, after 50

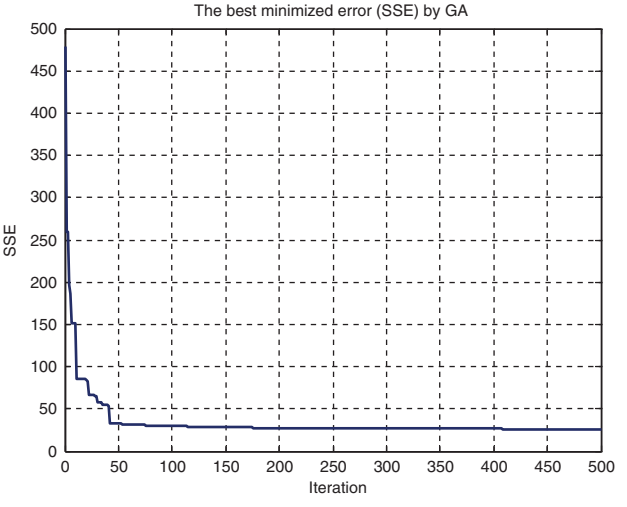

FIGURE 2. The best minimized error (SSE) computed by GA.

generations no superior improvement in cost decreasing is achieved, so one of the mentioned stopping criteria is satisfied that the algorithm can be terminated after 50 generations and the optimal solution (elite) returned.

Like GA method, according to the experience and trial and error, the maximum number of iterations for PSO procedure was set to 500. Figure 3 shows the best minimized error (SSE), obtained by PSO in 500 iterations. As this figure demonstrates, after 270 generations no significant improvement in cost reducing is attained. Therefore, one of the mentioned stopping criteria is satisfied that the algorithm can be stopped after 270 generations and the optimal solution (global best particle) returned.

Tables 1 and 2 show independent factors and the levels. On the basis of GA and PSO, 12 experimental points were performed. The results of the reactions are given in Tables 1 and 2.

The accuracy of the proposed model was validated by conducting other reactions with different conditions and then comparing the obtained results with the model. The internal percent error (PE) of the proposed model can be calculated using equations (2) and (3) for the 12 experiments. The average PEs for all 12 experiments obtained by GA and PSO are 3.86 and 4.05%, respectively (see Tables 1 and 2). The CPY is the production yield of glucosamine hydrochloride that was calculated using equations (4) and (5), and OPY is the production yield of glucosamine hydrochloride that was obtained in defined conditions for each experiment as shown in Tables 1 and 2.

$$\mathrm{PE} = \frac{|\mathrm{OPY} - \mathrm{CPY}|}{\mathrm{OPY}} \times 100 \tag{3}$$

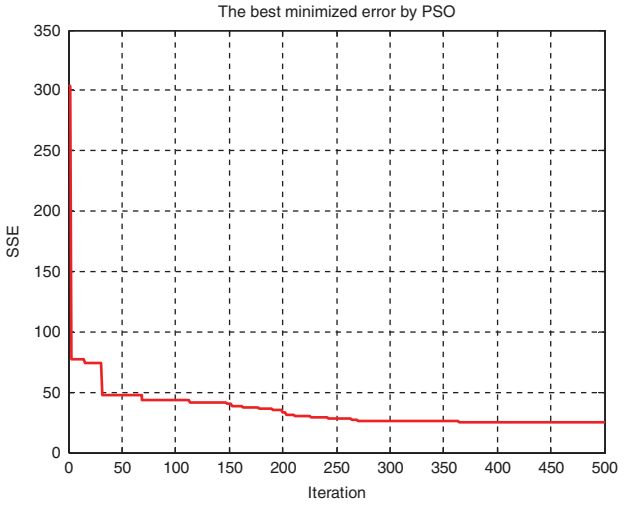

FIGURE 3. The best minimized error (SSE) computed by PSO.

TABLE 1
Experimental Design Results from Various Trials for Preparation of Glucosamine HCl and Internal PE of the Proposed Model for the 12 Experiments by GA

Trial	X_1 (%)	X_2 (vol/wt)	X_3 (h)	OPY (%)	CPY (%)	PE (%)
1	37	3:1	1	51.5	49.47	3.95
2	37	9:1	1	68.5	66.16	3.41
3	30	3:1	1	35.5	35.54	0.11
4	30	9:1	1	51	51.69	1.36
5	37	3:1	4	71.5	74.93	4.80
6	37	9:1	4	97	96.99	0.01
7	30	3:1	4	78.5	70.55	10.13
8	30	9:1	4	97.5	92.06	5.58
9	20	3:1	1	18	15.64	13.11
10	20	9:1	1	30	31.02	3.42
11	20	3:1	4	64	64.28	0.44
12	20	9:1	4	85	85.02	0.02
Average of PE						3.86

TABLE 2
Experimental Design Results from Various Trials for Preparation of Glucosamine HCl and Internal PE of the Proposed Model for the 12 Experiments by PSO

Trial	X_1 (%)	X_2 (vol/wt)	X_3 (h)	OPY (%)	CPY (%)	PE (%)
1	37	3:1	1	51.5	51.50	0.00
2	37	9:1	1	68.5	65.70	4.07
3	30	3:1	1	35.5	36.22	2.03
4	30	9:1	1	51	51.00	0.00
5	37	3:1	4	71.5	75.26	5.26
6	37	9:1	4	97	96.99	0.00
7	30	3:1	4	78.5	70.62	10.03
8	30	9:1	4	97.5	92.94	4.67
9	20	3:1	1	18	14.39	20.05
10	20	9:1	1	30	30.00	0.00
11	20	3:1	4	64	64.00	0.00
12	20	9:1	4	85	87.14	2.52
Average of PE						4.05

The modified model, obtained using GA, for production yield is

$$Y = -55.4701 + 2.4052X_1 + 2.0079X_2 + 24.4036X_3 + 0.0128X_1X_2 - 0.4542X_1X_3 + 0.2978X_2X_3 \tag{4}$$

The modified model, obtained using PSO, for production yield is

$$Y = -63.3256 + 2.7314X_1 + 2.4588X_2 + 25.4212X_3 - 0.0137X_1X_2 - 0.5069X_1X_3 + 0.4185X_2X_3 \quad (5)$$

These models indicate that the production yield of glucosamine hydrochloride depends linearly on acid concentration, solids to acid solution ratio, and time and also the cross-product of acid concentration and time and the cross-product of solids to acid solution ratio and time. The production yield significantly increased with an increase of acid concentration, acid solution ratio, and reaction time. The production yield is inversely related to the cross-product of acid concentration and time. It means that at high acid concentrations, the longer reaction times give lower production yields. This phenomenon could be attributed to the side reactions, which produce impurities and hence low production yield. Although a higher acid concentration can cause the produced glucosamine to degrade, it promotes chitin hydrolysis. According to our preliminary experiments, stepwise addition of acid solution can decrease side reactions and increase the yield (Mojarrad et al., 2007).

Validation. To evaluate the external predictive performance of the models, six more experiments were carried out in duplicate as a test set. Tables 3 and 4 show conditions and results of these reactions. In these tables, CPY is the amount of glucosamine hydrochloride that was calculated using equations (4) and (5). The results revealed that the average PEs for these six experiments by GA and PSO are 6.84 and 7.11%, respectively. Considering the low internal (3.86 and 4.05%) and external (6.84 and 7.11%) PEs, it might be concluded that the models have a good predictive power in the studied range of variables.

Correlation between the actual and the predicted production yield for training and validation sets are shown in Figures 4 and 5, respectively.

TABLE 3
Conditions, Obtained and Calculated Yield, and Corresponding PE of the Test Set Reactions for Validation of the Proposed Model by GA

Trial	X_1 (%)	X_2 (vol/wt)	X_3 (h)	OPY (%)	CPY (%)	PE (%)
1	37	6:1	1	57.5	57.82	0.54
2	37	6:1	2	64	67.29	4.99
3	37	6:1	4	86	85.96	0.04
4	30	6:1	1	43.5	43.62	0.27
5	30	6:1	2	80	56.18	29.78
6	30	6:1	4	86	81.30	5.46
Average of PE						6.84

TABLE 4
Conditions, Obtained and Calculated Yield, and Corresponding PE of the Test Set Reactions for Validation of the Proposed Model by PSO

Trial	X_1 (%)	X_2 (vol/wt)	X_3 (h)	OPY (%)	CPY (%)	PE (%)
1	37	6:1	1	57.5	58.60	1.92
2	37	6:1	2	64	67.78	5.91
3	37	6:1	4	86	86.13	0.15
4	30	6:1	1	43.5	43.61	0.26
5	30	6:1	2	80	56.34	29.58
6	30	6:1	4	86	81.78	4.90
Average of PE						7.11

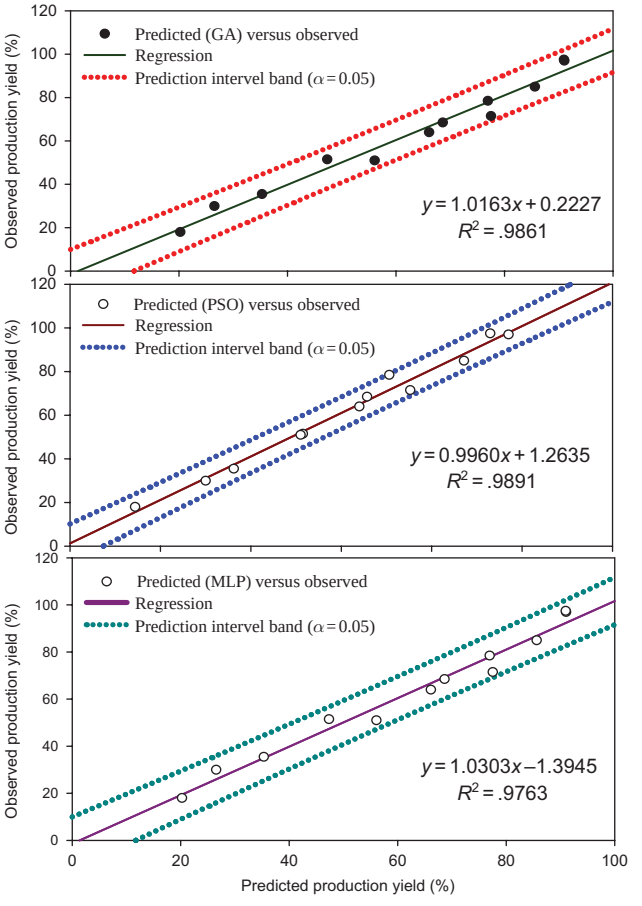

FIGURE 4. Correlation between the actual and the predicted production yield for different models (training data); GA, PSO, and MLP.

Artificial Neural Network Approach

A two-layer network with a sigmoidal transfer function and descent gradient with momentum back propagation was designed in this study. The transfer function possesses

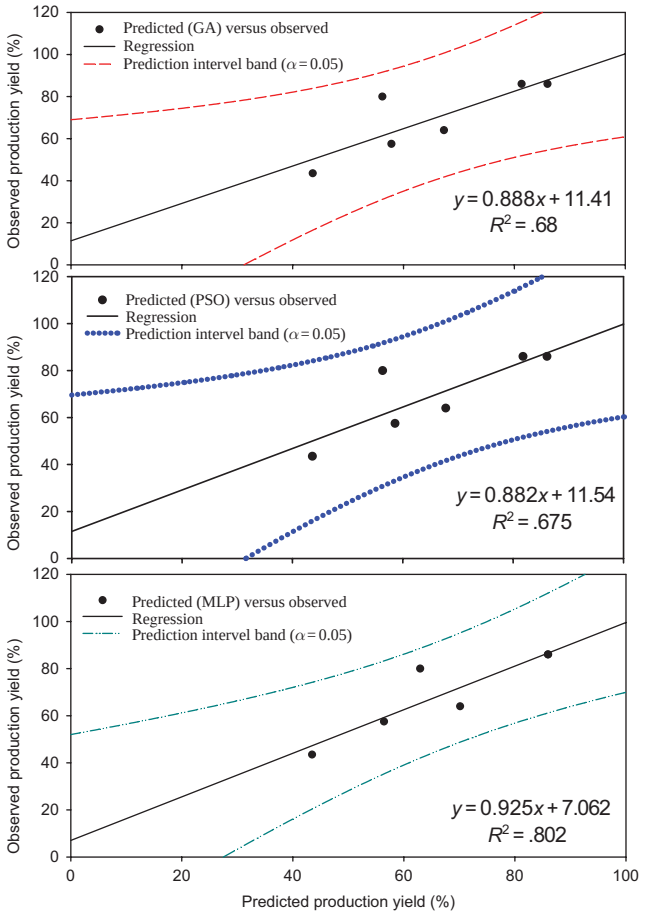

FIGURE 5. Correlation between the actual and the predicted production yield for different models (validation data); GA, PSO, and MLP.

minimum and maximum values of 0 and 1, respectively. Before training process, the input and output values were normalized between 0.1 and 1, which allows the network to slightly exceed the minimum and maximum values that were given in the original data file. After simulation, the values of predicted data sets were transformed to the true values. Inputs were the acid concentration (%) (X_1), acid solution to solid ratio (vol/wt) (X_2), and reaction time (X_3), and the output was production yield of glucosamine hydrochloride. The number of neurons in hidden layer was optimized by trial and error. A layer consisting of three neurons was found optimum, simple and uncomplicated neural network. An MLP with one hidden layer and three neurons in it is fast and easy to implement. Bias terms (an additional neuron in the input and hidden layers with constant value equal to –1) are used to help in better modeling. Therefore, the optimum topology of the networks was 4–4–1.

Because there are several local minima where the model could arrive, when nearly the similar results were acquired by different starting weights, the process considered in optimum condition and the obtained good initialization weights were used as the neural network initial weights. When the maximum number of epochs was allowed to be 1,000, the TSSE was approximately constant. For preventing overtraining problem, first a network that is just large enough to provide an adequate fit is selected by many trials and errors. As mentioned before, the optimum topology of the network obtained was 4–4–1. Second early stopping technique is applied. The data were divided into three subsets. The first subset was the training set. To construct the second and third subsets, the test and the validation sets are considered to be same sets. Then the error on the validation set was monitored during the training procedure. As Figure 6 shows, because the structure of the network was selected optimal, the validation error decreased during the initial phase of training, as did the training set error. Approximately after 140 epochs, the TSSE of the train set decreased slowly and at that point the validation error began to rise. Therefore, according to the early stopping technique, it is logical to stop training after 140 epochs.

Table 5 shows the approximated production yield of glucosamine hydrochloride by MLP network in training set. Mean of the PE% is 5.38% for training data, which is a suitable value for training phase. Now that the model is obtained, all the weights are fixed to examine the generalization ability of the trained neural network by the remaining six unused data sets.

Validation. As dealt with the previous procedure, to evaluate the external predictive performance of the neural network model, six more experiments were carried out in duplicate as a test set. Table 6 shows conditions and results of these reactions. In these tables, CPY is the amount of glucosamine hydrochloride that was calculated using the weight-fixed MLP network. The results revealed that the average PE for these six

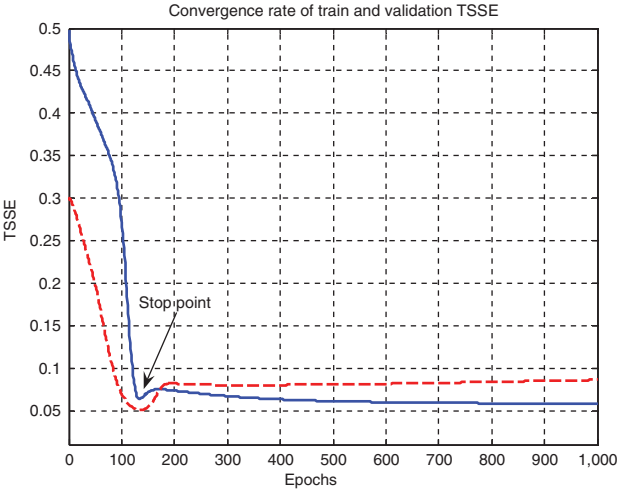

FIGURE 6. Schematic diagram of TSSE (solid line is for training error and dashed line is for validation error).

TABLE 5
Experimental Design Results from Various Trials for Preparation of Glucosamine HCl and Internal PE for the 12 Experiments of the Proposed Model by MLP

Trial	X_1 (%)	X_2 (vol/wt)	X_3 (h)	OPY (%)	CPY (%)	PE (%)
1	37	3:1	1	51.5	47.33	8.09
2	37	9:1	1	68.5	68.63	0.19
3	30	3:1	1	35.5	35.31	0.53
4	30	9:1	1	51	56.07	9.95
5	37	3:1	4	71.5	77.51	8.40
6	37	9:1	4	97	91.03	6.14
7	30	3:1	4	78.5	76.94	1.98
8	30	9:1	4	97.5	90.95	6.71
9	20	3:1	1	18	20.23	12.43
10	20	9:1	1	30	26.53	11.54
11	20	3:1	4	64	66.11	3.29
12	20	9:1	4	85	85.61	0.72
Average of PE						5.83

TABLE 6
Conditions, Obtained and Calculated Yield, and Corresponding PE of the Test Set Reactions for Validation of the Proposed Model by MLP

Trial	X_1 (%)	X_2 (vol/wt)	X_3 (h)	OPY (%)	CPY (%)	PE (%)
1	37	6:1	1	57.50	56.42	1.87
2	37	6:1	2	64	70.13	9.58
3	37	6:1	4	86	85.93	0.07
4	30	6:1	1	43.5	43.46	0.08
5	30	6:1	2	80	62.93	21.33
6	30	6:1	4	86	85.99	0.00
Average of PE						5.49

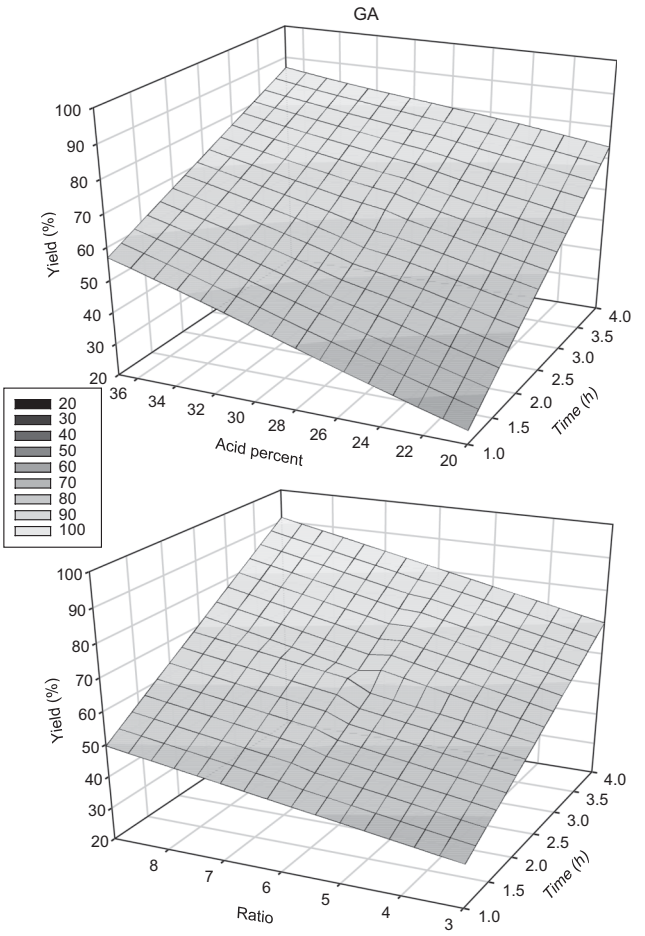

FIGURE 7. Surface plot of yield (%) versus time (h); acid (%) (top) and ratio (acid solution to solid; vol/wt) (bottom) for GA model.

experiments by MLP is 5.49%. Considering the train phase PE (5.83%), it might be concluded that the model should have a good predictive power in the studied range of variables and it has the ability of generalization to unknown cases, with less error.

The predicted models were used to create a surface plot within the experimental region (Figures 7–9). In our study, glucosamine hydrochloride was obtained in optimal conditions, using 30% HCl with a 9:1 acid solution to solid ratio (vol/wt) in 4 h with 96–98% yield.

CONCLUSION

Both intelligent approaches were successful in modeling the relationship between the production yield of glucosamine hydrochloride and three mentioned variables (the acid concentration [%], acid solution to solid ratio [vol/wt], and reaction time) (Figures 4 and 5). Hence, by comparing Tables 1–5, we can find out that at model fitting, GA and PSO methods were slightly better than the ANN model. But in the validation of the models, the MLP model was better than the obtained models by GA and PSO methods. In fact, the MLP model has more nonlinearity than the others that it leads to a more generable model with lesser error in validation. Lesser error in test phase implies the more accurate and reliable model to other unknown cases, which is the major characteristic of the ANNs.

ACKNOWLEDGMENT

We are grateful to the Drug Applied Research Center, Tabriz University of Medical Sciences, for partial financial support.

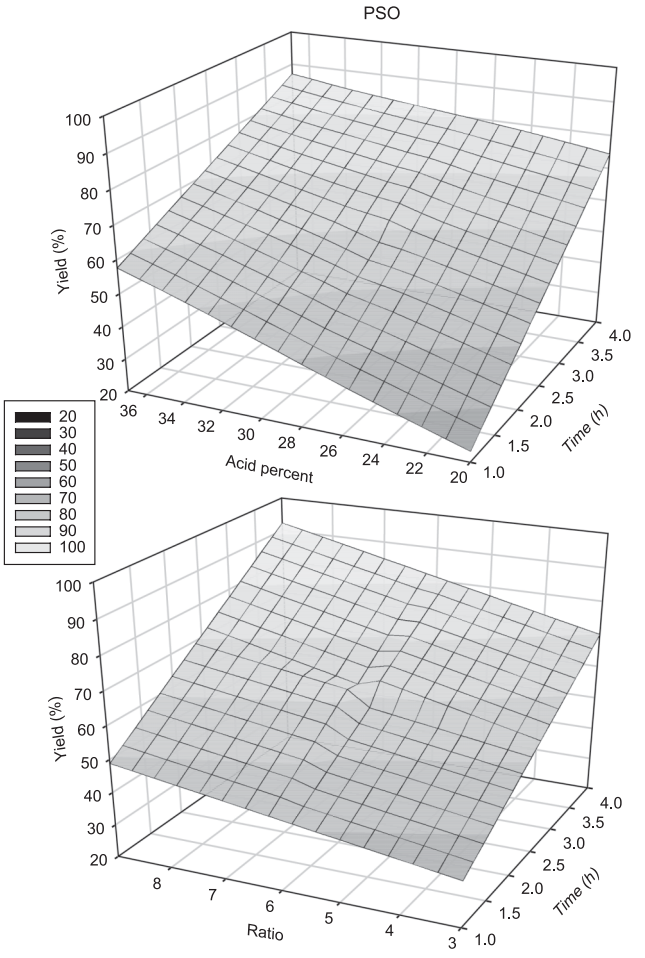

FIGURE 8. Surface plot of yield (%) versus time (h); acid (%) (top) and ratio (acid solution to solid; vol/wt) (bottom) for PSO model.

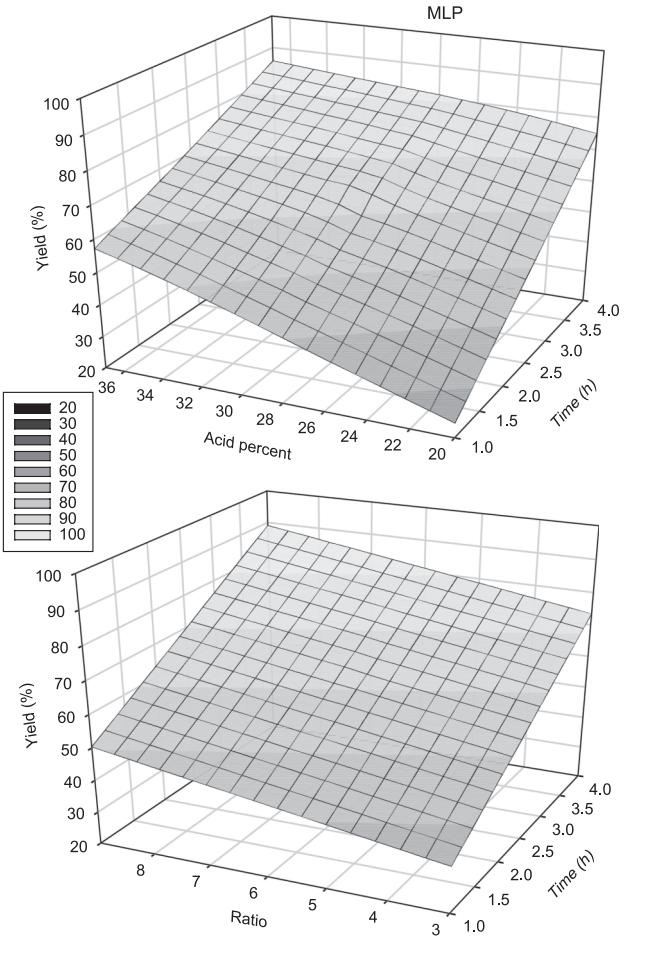

FIGURE 9. Surface plot of yield (%) versus time (h); acid (%) (top) and ratio (acid solution to solid; vol/wt) (bottom) for MLP model.

REFERENCES

Eberhart, R., & Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. *Proceedings of the 7th annual conference on evolutionary programming* (pp. 611–618). Berlin: Springer.

Fausett, L. (1994). *Fundamentals of neural networks*. New Jersey: Prentice Hall.

Fletcher, R. (2000). *Practical methods of optimization* (2nd ed.). Chichester, NY: John Wiley & Sons.

Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.

Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). *Neural network design*. Boston, MA: PWS Publishing.

Hauselman, H. J. (2001). Nutripharmaceuticals for osteoarthritis. *Best Pract. Res. Clin. Rheumatol., 15*, 595–607.

Holland, J. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Kennedy, J. (1997). The particle swarm: Social adaptation of knowledge. *Proceedings of the IEEE international conference on evolutionary computation* (pp. 303–308). Indianapolis, IN: IEEE Press.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE international conference on neural networks* (Vol. 4, 1942–1948). Perth, Australia: IEEE Press.

Mankin, H. J., Brandt, K. D., & Shulman, L. E. (1986). Workshop on etiopathogenesis of osteoarthritis: Proceedings and recommendations. *J. Rheumatol., 13*, 1130–1160.

Michalewicz, Z., & Fogel, D. (2000). *How to solve it: Modern heuristics*. Berlin: Springer Verlag.

Mojarrad, J. S., Nemati, M., Valizadeh, H., Ansarin, M., & Bourbour, S. (2007). Preparation of glucosamine from exoskeleton of shrimp and predicting production yield by response surface methodology. *J. Agric. Food Chem., 55*(6), 2246–2250.

Nelson, M. M., & Illingworth,W. T. (1991). *A practical guide to neural nets*. Reading, MA: Addison-Wesley.

Nemati, M., Valizadeh, H., Ansarin, M., & Ghaderi, F. (2007). Development of a simple and sensitive high-performance liquid chromatography method for determination of glucosamine in pharmaceutical formulations. *J. AOAC Int., 90*(2), 354–357.

No, H. K., Meyers, S. P., & Lee, K. S. (1989). Isolation and characterization of chitin from crawfish shell waste. *J. Agric. Food Chem., 37*(3), 575–579.

Percot, A., Viton, C., & Domard, A. (2003a). Characterization of shrimp shell deproteinization 19. *Biomacromolecules, 4*(5), 1380–1385.

Percot, A., Viton, C., & Domard, A. (2003b). Optimization of chitin extraction from shrimp shells 22. *Biomacromolecules, 4*(1), 12–18.

Randy, L. H., & Sue, E. H. (2004). *Practical genetic algorithms* (2nd ed.). New Jersey: John Wiley & Sons.

Rardin, R. L. (1998). *Optimization in operations research.* New Jersey, USA: Prentice Hall.

Shahidi, F., & Synowiecki, J. (1991). Isolation and characterization of nutrients and value-added products from snow crab (*Chionoecetes opilio*) and shrimp (*Pandalus borealis*) processing discards. *J. Agric. Food Chem.*, *39*(8), 1527–1532.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *Proceedings of IEEE international conference on evolutionary computation* (pp. 69–73). Piscataway, NJ: IEEE Press.

Simpson, P. K. (1990). *Artificial neural systems: Foundations, paradigms, applications, and implementations.* New York: Pergamon Press.

APPENDIX A: PSEUDOCODE FOR A GA PROCEDURE

Begin;
Initialization;
Generate random population of chromosomes;
For i = 1 to number of generations do
For each individual calculate fitness;
Crossover procedure:
Eliminate X_{rate}% of the chromosomes
Select two parents using selection operator to reproduce the eliminated chromosomes;
Generate new offspring, as new solutions, using the crossover;
Mutation procedure:
Select one chromosome at random;
Generate an offspring by mutation;
Calculate the fitness of the offspring;
If offspring is better than the worst chromosome, then replace the worst chromosome by offspring;
Next i;
Check if termination = true;
End;

APPENDIX B: PSEUDOCODE FOR A PSO PROCEDURE

Begin;
Initialization;
Generate random population of particles;
For i = 1 to number of iterations do
For each individual calculate fitness;
For each particle;
Update the global best positions and local best positions;
Calculate the velocities;
Update particle position;
Next i;
Check if termination = true;
End;

Copyright of Drug Development & Industrial Pharmacy is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.