Chemical Product and Process Modeling

Volume 7, *Issue* 1	2012	*Article* 3

Search for Optimum Operating Conditions for a Water Purification Process Integrated to a Heat Transformer with Energy Recycling using Artificial Neural Network Inverse Solved by Genetic and Particle Swarm Algorithms

Youness El Hamzaoui, *Autonoumus University of Morelos State*
Bassam Ali, *CIE- UNAM*
J. Alfredo Hernandez, *Autonoumus University of Morelos State*
Obed Cortez Aburto, *Universidad Politécnica de Puebla*
Outmane Oubram, *IIM-UNAM*

Recommended Citation:
El Hamzaoui, Youness; Ali, Bassam; Hernandez, J. Alfredo; Aburto, Obed Cortez; and Oubram, Outmane (2012) "Search for Optimum Operating Conditions for a Water Purification Process Integrated to a Heat Transformer with Energy Recycling using Artificial Neural Network Inverse Solved by Genetic and Particle Swarm Algorithms," *Chemical Product and Process Modeling*: Vol. 7: Iss. 1, Article 3.
DOI: 10.1515/1934-2659.1614

Search for Optimum Operating Conditions for a Water Purification Process Integrated to a Heat Transformer with Energy Recycling using Artificial Neural Network Inverse Solved by Genetic and Particle Swarm Algorithms

Youness El Hamzaoui, Bassam Ali, J. Alfredo Hernandez, Obed Cortez Aburto, and Outmane Oubram

Abstract

The coefficient of performance (COP) for a water purification process integrated to an absorption heat transformer with energy recycling was optimized using the artificial intelligence. The objective of this paper is to develop an integrated approach using artificial neural network inverse (ANNi) coupling with optimization methods: genetic algorithms (GAs) and particle swarm algorithm (PSA). Therefore, ANNi was solved by these optimization methods to estimate the optimal input variables when a COP is required. The paper adopts two cases studies to accomplish the comparative study. The results illustrate that the GAs outperforms the PSA. Finally, the study shows that the GAs based on ANNi is a better optimization method for control on-line the performance of the system, and constitutes a very promising framework for finding a set of "good solutions".

KEYWORDS: artificial intelligence, evolutionary algorithms, swarm intelligence, objective function, heat pump

Author Notes: Y. El Hamzaoui, expresses his gratitude to CONACyT for the scholarship awarded to postgraduated Ph.D. Studies. Corresponding autor: baali@cie.unam.mx

1. Introduction

The absorption heat transformer is a system that consists of a thermodynamic device capable of producing useful heat at a thermal level superior to the one in the source (Torres, 1997). This heat transformer plays a special role in the process industries to minimize the energy consumption, because, it could be used in any other system that requires a temperature greater than the one provided by the origin. In addition, an absorption heat transformer is used extensively and regularly in water purification process. It is known that the coefficient of performance (COP) is a very important variable for determining the performance of an absorption heat transformer according to Equation (1). This COP is defined as the heat delivered in the absorber per unit of heat load supplied to generator and evaporator (Huicochea et al. 2004).

$$COP = \frac{Q_{AB}}{Q_{GE} + Q_{EV}} \tag{1}$$

However, to describe the behavior of the COP, Siqueiros and Romero (2007) have used a thermodynamic model to simulate COP values for a water purification process integrated to an absorption heat transformer. This model was based on a set of assumptions as a heat loss, pressure drops in the tubing, whenever these considerations are so difficult to be fulfilled in practice. Thus, controlling this process is impossible according to the thermodynamic model in steady-state. Therefore in practice an artificial intelligence tools such as an artificial neural network (ANN) can provide a new approach to process without take into account any previous assumptions (Hernandez et al. 2008 and Hernandez et al 2009). ANN is a collection of interconnecting computational elements which function like neurons in biological brain. It has the ability to model processes by learning from input and output data, without mathematical knowledge of the process.

The COP could be calculated on-line, when the input variables are well known in the water purification process integrated to the heat transformer with energy recycling. Indeed, the problem is that this COP computed by ANN is not ideal in the system, and therefore it is necessary that its input variables are well known when a given COP is required, that means ‘Find the effect of this cause.’ However, if it would be used to convert observed measurements into information about system, then this kind of problems belongs to the class of inverse problem, there exist many fields of sciences where inverse problems appear. Some examples are: astronomy (blurred images of the Hubble satellite), econometrics (instrumental variables), financial mathematics (model calibration of the volatility), medical image processing (X-ray tomography), and quantum physics (quantum homodyne tomography). These are problems where we have indirect observations of an object (a function) that we want to reconstruct.

The common structure of all these problems, coming from very different fields, is that we only have access to indirect observations due to its indirect nature, solving an inverse problem is usually rather difficult. Nevertheless, in this investigation, we have found that the control strategy on COP's system is an inverse problem was developed through inverting an artificial neural network (ANNi) to determine the optimum input variables on a required COP in the system (Hernandez et al. 2009). The proposed method ANNi is a new tool which inverts artificial neural network (ANN) and it uses an optimization method to find the optimum parameter value (or unknown parameter) for given required conditions in the process. In order to do so, first, it is necessary to build the artificial neural network (ANN) model that simulates the output parameters of water purification process is constituted of a feedforward network with one hidden layer to simulate output, considering one or more well-known input parameters of the process.

Levenberg-Marquardt learning algorithm, hyperbolic tangent sigmoid transfer-function, linear transfer-function and several neurons in the hidden layer (due to the complexity of the process) are considered in the built model. As soon as the model was validated, the second step was to invert the model. With the required output and some input parameters it is possible to calculate the unknown input parameters.

However, it is important to note that the analytical solution with one neuron in the hidden layer neural model exists, and it is described in the subsection 4.1.1. Nevertheless, in the case that a proposed ANN model has more than one neuron in the hidden layer it is necessary to use an optimization method. Hernandez et al. (2009) have applied the inverted model of a neural network in one absorption heat transformer with energy recycling to predict only one input parameter which could be controlled in order to find the ideal value of COP by using a Nelder Mead Algorithm, which is a well defined numerical method for twice differentiable and unimodal problems, the inconvenient of this algorithm is only suitable for unconstrained optimization of linear functions, also, he couldn't estimate more than one parameter, and may only give local solutions (Lagarias et al. 1998). But when the optimization problem required more than one variable, in this situation, the process could be more complicated.

In this work, we propose to solve the problem of optimization previously mentioned of the fitness function trough the ANNi using the genetic algorithms (GAs) and particle swarm algorithm (PSA). In addition, a comparaison study between those algorithms was also carried out. Many works have used (GAs), (PSA) and others technics trying to solve diverse problems of simulation and optimization in the chemical engineering process (Chang et al. 2011; El

Hamzaoui et al. 2010; Gharebagh and Mostoufi 2004; Hamidipour et al. 2005; Iranshahi et al. 2004; Mehrpooya et al. 2010; Mostoufi et al. 2005; Ravagnani et al. 2005).

The paper is organized as follows, second section will give an overview about system description and experimental data. Third section will discuss the artificial neural network model. Fourth section is devoted to the optimization approach applied on ANNi, then the fifth section is assigned to the results and discussions. Finally, the conclusions on this work are drawn.

2. System description and experimental data

Figure. 1 shows a schematic diagram for a heat transformer. Useful heat (Q_{AB}) is the result of the reaction between working fluid vapour and absorbent solution (which comes from the evaporator and generator, respectively). After this process, a diluted Water/LiBr solution, goes to the generator. In the generator, the aqueous solution receives a quantity of heat (Q_{GE}) from an external heat supply. Under these conditions, working fluid steam leaves the generator and goes through a condenser where it loses heat (Q_{CO}) and the fluid is condensed. This condensate goes to the evaporator where external heat (Q_{EV}) is supplied and the working fluid evaporates at high pressure and goes to the absorber. At the same time, a concentrated Water/LiBr solution goes to the absorber, and at this point, the cycle starts again. Also, it can be observed that the heat transformer is integrated to the water purification system. The absorber gives the unique useful heat delivered (Q_{AB}) in the heat transformer. Q_{AB} is used to heat the impure water until it reaches its boiling point and partly evaporates. The two phases (liquid water and steam) leave the absorber and are separated through a phase separator. The liquid phase returns to the suction pump and the steam produced goes through an auxiliary condenser where heat is transferred as steam condenses while the heat source stream is heated (Siqueiros et al. 2007). In addition on the above of this Figure shows, also the process of water purification system integrated to a heat transformer with heat recycling in the heat source. (Huicochea and Siqueiros 2010).

Experimental database provided by Morales (2005) consists of different COP values, obtained from a portable water purification process coupled to an absorption heat transformer with energy recycling. The experimental data set was obtained at different initial concentration of LiBr in LiBr+H_2O mixture, different temperatures in the absorber, the generator, the evaporator, and the condenser as well as different pressures in the absorber and the generator. The transitory and steady states were taken into account for each initial concentration of the mixture. After 2 h from start-up, data were collected for 4 h. The experiments were carried out at eight different initial conditions with at least two replicates. The arrangement was 8 x 2 with 4 h of information acquisition. Thus, a database of 11882 samples was obtained. A summary of 16 operating parameters (10 levels of temperatures, 4 of concentrations and 2 of pressures) is shown in Table 1. The thermodynamic properties of the LiBr + H_2O mixture were estimated with Alefeld correlations cited by (Torres, 1997). The input and output temperatures of each component (AB, GE, CO and EV) were obtained experimentally. At the same time, the pressure of two components (AB and GE) was registred with a temperature pressure acquisition system (thermocouple conditioner and Agilent equipment with commercial software). The input and output concentrations in the AB and GE were established by a refractometer (refraction index). In this process, LiBr + H_2O mixture was used as the working mixture in the absorber and generator, while only H_2O was used in the evaporator and condenser.

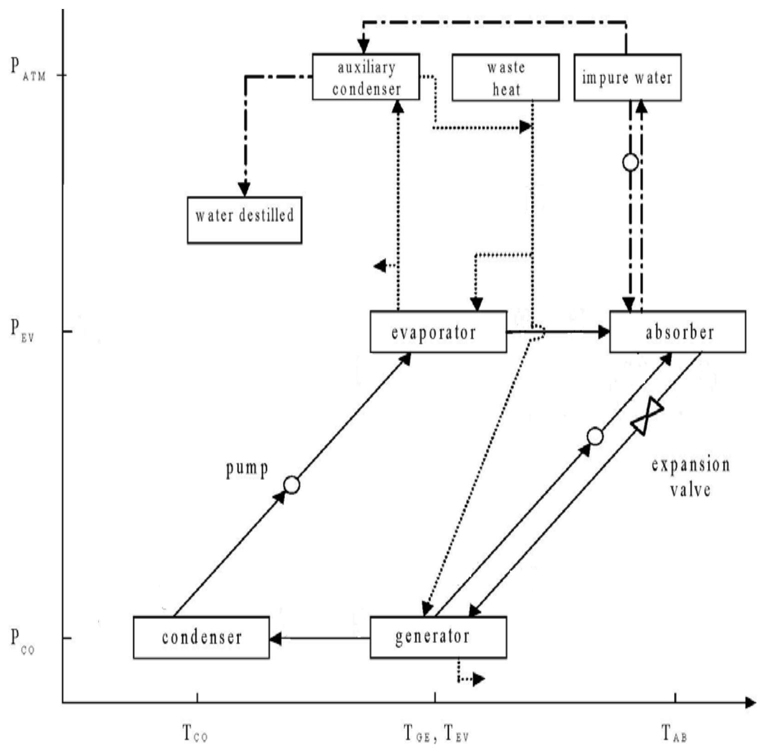

Fig. 1 Schematic diagram of the integration of the water purification process to an absorption heat transformer with energy recycling. The continuous line (-) represents the absorption heat transformer, the line and point (- . -) are the water purification process and the dotted lines (….) are the recycling energy.

Table 1 Experimental operation range conditions studied to obtain the COP values

Variables	Mean±Std	Range
Operation parameters		
^{o}C $T_{in.GE\text{-}AB}$	88.60±1.99	76.29-91.53
$T_{in.EV\text{-}AB}$	83.96±2.39	74.56-89.93
$T_{out.AB\text{-}GE}$	93.85±2.28	84.31-98.27
$T_{in.AB\text{-}GE}$	87.17±1.53	74.99-92.58
$T_{out.GE\text{-}CO}$	88.60±1.99	76.29-91.53
$T_{out.GE\text{-}AB}$	82.03±0.88	77.03-83.89
$T_{in.CO}$	48.67±3.47	40.37-65.03
$T_{out.CO}$	30.44±1.53	26.77-33.79
$T_{in.EV}$	35.57±7.70	28.52-85.33
$T_{out.EV\text{-}AB}$	83.96±2.40	74.56-89.93
Operational parameters, %		
$X_{in.AB}$	53.84±1.31	51.66-55.36
$X_{out.AB}$	52.77±1.33	50.75-54.36
$X_{in.GE}$	52.76±1.32	50.75-54.36
$X_{out.GE}$	55.01±1.03	53.16-56.07
Operational parameters, in Pascal (Pa) (absolute)		
P_{AB}	8.64±1.12	7-11.5
P_{GE}	20.54±0.49	19-21.10

3. Artificial neural network model

According to Hernandez et al. (2009) and Kůrková (1992), the databases previously mentioned in the section 2 was successfully used to train the ANN model using backpropagation procedure, in order to predict the COP in water purification systems integrated to a heat transformer with energy recycling, for this motivation to determine the best backpropagation training algorithm, ten backpropagation algorithms were studied. In addition, three neurons were used in the hidden layer for all backpropagation algorithms.

Table 2 shows a comparison of different backpropagation training algorithms. Levenberg-Marquardt backpropagation training algorithm could have small root mean square error $(RMSE)$ and relative standard deviation (RSD), respectively, on the other hand, we found training with Levenberg Marquardt algorithm can run smoothly in computer with lower expanded memory specification (EMS), and the training time is quickly, than the other backpropagation algorithms. Because, the Levenberg-Marquardt algorithm was designed to approach second order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feed-forward networks), then the Hessian matrix can be approximated as:

$$H = J^T J \tag{2}$$

However, the gradient can be computed as:

$$g = J^T e \tag{3}$$

Where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix. The Levenberg- Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton like up date:

$$X_{k+1} = X_k - \left[J^T J + \mu I\right]^{-1} J^T e \tag{4}$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. According to Hagan and Menhaj 1994, Newton's method is faster and more accurate near an error minimum, so the aim is to shift toward Newton's method as quickly as possible (, thus μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this context, the performance function is always reduced at each iteration of the algorithm. So, for these arguments, the Levenberg-Marquardt algorithm was considered the training algorithm in the present study. However, the performance of the ANN model was statistically measured by the root mean square error *(RMSE)*, relative standard deviation *(RSD)* and regression coefficient *(R^2)*, which are calculated with the experimental values and network predictions. These calculations are used as a criterion for model adequacy obtained as follows:

$$RMSE = \sqrt{\frac{\left(\sum_{n=1}^{N}\left(y_{n,pred} - y_{n,\exp}\right)^2\right)}{N}} \tag{5}$$

$$RSD = \frac{RMSE}{y_m} \times 100 \tag{6}$$

$$R^2 = 1 - \frac{\sum_{n=1}^{N}\left(y_{n,pred} - y_{n,\exp}\right)^2}{\sum_{n=1}^{N}\left(y_{n,\exp} - y_m\right)^2} \tag{7}$$

Where N is the number of data points, $y_{n,pred}$ is the network prediction, $y_{n,\exp}$ is the experimental response, y_m is the average of actual values and n is an index of data. The Figure 2 depicts also, the recurrent network architecture for a COP values and procedure used for neural network learning. While, the Table 3 shows, also some samples of the experimental measurement and simulated giving by ANN model.

Consequently, *RMSE* and *RSD* were used as the error function which measures the performance of the network. Therefore, the network having minimum *RMSE*, minimum *RSD* and maximum R^2 was selected as the best ANN model.

According to Hernandez et al. (2008), the proposed model is represented by the following equation:

$$COP = \sum_{i=1}^{S}\left[LW(1,s).\left(\frac{2}{1+\exp\left(-2\left(\sum_{k=1}^{K}\left(IW(s,k).In(k)+b1(s)\right)\right)\right)}-1\right)+b2\right] \quad (8)$$

Where s is the number of neurons in the hidden layer, k is the number of the input, and IW, W_o and b are weights and bias, respectively. Table 4 shows the adjustable parameters *(IW, W_o, b_1 and b_2)* of the proposed model.

Table 2 Comparison of 10 backpropagation algorithms with three neurons in the hidden layer

Backpropagatin algorithms	Function	RMSE	R^2	RSD %	Best linear equation
Levenberg-Marquardt	*trainlm*	9.5219×10^{-7}	0.998	2.9343×10^{-4}	Y=0.998X+0.3
Batch gradient descent	*traingd*	8.3247×10^{-6}	0.988	0.0026	Y=0.986X+0.9
Batch gradient descent with momentum	*traingdm*	7.8918×10^{-5}	0.987	0.0243	Y=0.988X+0.8
Polak-Ribiere conjugate gradient	*traincgp*	7.9934×10^{-5}	0.979	0.0246	Y=0.957X+2.5
Scaled conjugate gradient	*trainscg*	6.4627×10^{-4}	0.974	0.1992	Y=1.020X-0.7
BFGS quasi-Newton	*trainbfg*	6.9871×10^{-4}	0.971	0.2153	Y=0.982X+1.2
Powell-Beale conjugate gradient	*traincgb*	6.9997×10^{-4}	0.965	0.2157	Y=0.960X+2.0
One step secant backpropagation	*trainoss*	6.9287×10^{-3}	0.782	2.1352	Y=0.617X+45.
Fletcher-Reeves conjugate gradient	*traincgf*	6.4159×10^{-2}	0.725	19.7716	Y=0.425X+34.
Variable learning rate	*traingdx*	6.9781×10^{-1}	0.718	215.0416	Y=0.386X+38

4. Optimization approach

Optimization involves finding the minimum/maximum of an objective function f(x) subject to some constraint x Є *S*. If there is no constraint for x to satisfy-or, equivalently, S is the universe-then it is called an unconstrained optimization; otherwise, it is a constrained optimization.

For optimization the desired results are computed in inverse form. A given required parameter is desired. What parameters can be set to obtain those conditions? This is a big problem we often face in a control room. Therefore, according to Equation (8), that it is possible to simulate the COP in water purification systems integrated to a heat transformer with energy recycling, when input parameters are well known. However, in many cases, the problem is that this COP calculated by ANN is not ideal in the system, and therefore it is necessary that its input variables are well known when giving a required COP.

The new control strategy which is proposed here using the ANN model of heat transformer integrated with energy recycling system described above as a standard model. The proposed solution is to use an inverse of neural network for optimizing the performance parameters of the process by using the optimization algorithms (GAs and PSA) to find the optimal input values for the required output. In this work, as mentioned above, the required output is the coefficient of performance, but the optimal input operating parameters to be found are the pressures in the absorber and generator, respectively, because, those pressures are the key of COP's system, in addition, could be modified with the operational form. That's why, these are two variables which the control system have to manipulate to achieve the required coefficient of performance.

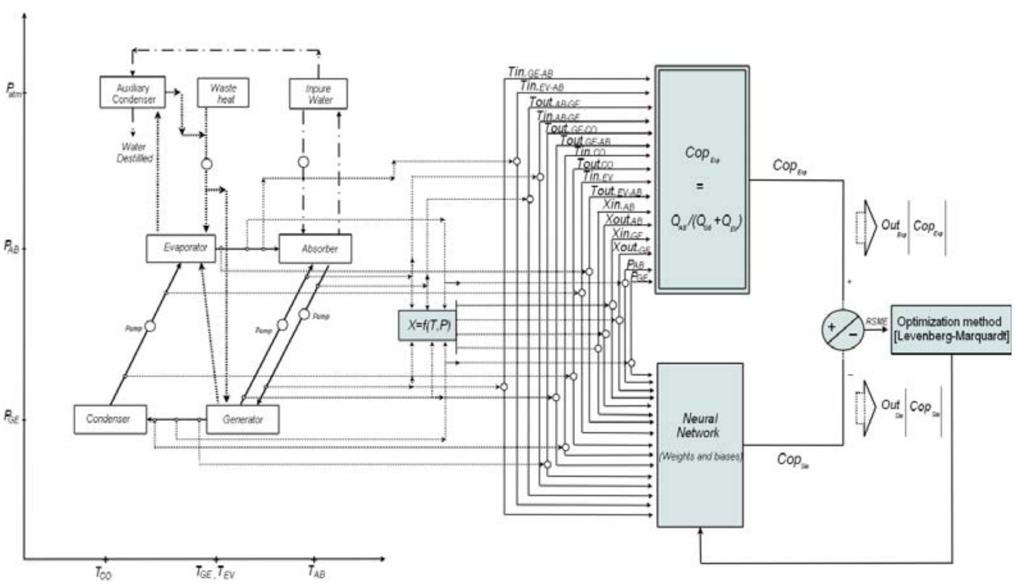

Fig. 2 Recurrent network architecture for COP values and procedure used for neural network learning

Table 3 Some samples of the experimental and simulated information of the system

	Test					
Input	**A**	**B**	**C**	**D**	**E**	**F**
$T_{in.GE\text{-}AB}$	90.54	89.52	89.72	89.49	89.64	87.7
$T_{in.EV\text{-}AB}$	83.79	82.63	85.79	85.94	79.61	81.67
$T_{out.AB\text{-}GE}$	96.06	92.67	94.17	92.67	95.1	94.32
$T_{in.AB\text{-}GE}$	85.68	86.18	88.93	85.26	89.12	87.93
$T_{out.GE\text{-}CO}$	90.54	89.52	89.72	89.49	89.64	87.7
$T_{out.GE\text{-}AB}$	81.56	82.1	83.33	80.88	82.51	81.28
$T_{in.CO}$	48.92	48.15	48.98	49.6	42.32	56.09
$T_{out.CO}$	30.89	33	30.53	29.39	28.52	33.23
$T_{in.EV}$	31.84	34.56	37.5	31.77	36.5	34.86
$T_{out.EV\text{-}AB}$	83.79	82.63	85.79	85.94	79.61	81.67
$X_{in.AB}$	52.39	52.98	54.3	53.6	55.31	55.25
$X_{out.AB}$	51.08	51.74	53.16	52.28	54.33	54.33
$X_{in.GE}$	51.08	51.74	53.16	52.28	54.33	54.33
$X_{out.GE}$	53.49	54.65	55.36	55.47	56.07	55.97
P_{AB}	10	9.5	8	7.5	11	9
P_{GE}	20	20	20.5	21	21	21.1

	Test					
Output	**A**	**B**	**C**	**D**	**E**	**F**
COP_{Exp}	0.35	0.3	0.38	0.21	0.39	0.35
COP_{Sim}	0.3474	0.2987	0.3767	0.2133	0.3913	0.3506

Table 4 Adjustable parameters obtained (weights and bias) in the proposed model with S=3, K=16

$\mathbf{W_i}$ $_{(s,k)}$	1.0471 1.6234 -8.5413 1.7987 7.1516 -1.1964 -1.0951 2.5118 -0.0034 -2.3428 242.3930 159.4554 72.5887 20.8372 -19.4769 -65.5883 20.0710 -6.4628 -46.4698 6.3265 26.4226 -16.5479 14.6389 37.4539 0.3227 4.4175 -216.8707 135.7784 115.0829 10.2770 17.5255 -47.0385 -3.7487 0.1137 3.5127 0.5452 0.6408 0.3451 -0.0254 0.0001 0.0032 0.1114 -0.4314 -7.1419 12.6125 3.2912 0.1576 1.1184
$\mathbf{W_o}$ $_{(1,s)}$	-0.1866 0.0239 -0.8825
b1 $_{(s)}$	129.0939 -17.7516 -6.4906
b2 $_{(L)}$	0.2427

4.1. Artificial neural network inverse (ANNi)

4.1.1. Inverse Neural Network with one neuron in hidden layer in ANN model

We can express analytically the solution of ANNi when we have just one neuron in the hidden layer:

- If *tansig* and *purelin* are considered as the hyperbolic tangent sigmoid and linear transfer function in the hidden layer and output layer, and *k=1*

$$Out\ (1) = LW\ (1,1).\left(\frac{2}{1+\exp\left(-2.\left(IW\ (1,k).In\,(k)+b1\right)\right)}-1\right)+b2 \tag{9}$$

This can be transformed into:

$$Out\ (1) = \frac{2.LW\ (1,1)}{1+\exp\left(-2.\left(IW\ (1,k).In\,(k)+b1\right)\right)}-LW \tag{10}$$

$$1+\exp\left(-2\left(IW\ (1,k).In\,(k)+b1=\frac{2.LW\ (1,k)}{Out\ (1)+LW\ (1,1)-b2}\right)\right) \tag{11}$$

$$\exp\left(-2.\left(IW(1,k).In(k)+b1\right)\right)=\frac{2.LW(1,1)-Out(1)-LW(1,1)+b2}{Out(1)+LW(1,1)-b2} \tag{12}$$

$$-2.\left(IW\ (1,k).In\,(k)+b1\right)=Ln\left(\frac{LW\ (1,1)-Out\ (1)+b2}{Out\ (1)+LW\ (1,1)-b2}\right) \tag{13}$$

$$IW\ (1,1) = -\frac{1}{2}\,Ln\left(\frac{LW\ (1,1)-Out\ (1)+b\,2}{Out\ (1)+LW\ (1,1)-b\,2}\right)-b1 \tag{14}$$

Let *k=k(1)* would be the input parameter to be calculated when one output parameter is required. Then:

$$k(1) = -\frac{1}{2.IW(1,1)} Ln\left(\frac{LW(1,1) - Out(1) + b2}{Out(1) + LW(1,1) - b2}\right) - \frac{b1}{IW(1,1)} \quad (15)$$

- If *logsig* and *purelin* are considered as the logistic tangent sigmoid and linear transfer function in the hidden layer and output layer, and *k=1*

$$Out(1) = LW(1,1).\left(\frac{1}{1 + \exp(-IW(1,1).In(k) + b1)}\right) + b2 \quad (16)$$

Then the input parameter can be calculated from Equation (14):

$$k(1) = \frac{b1 - Ln\left(\frac{LW(1,1)}{Out(1)} - 1\right)}{IW(1,1)} \quad (17)$$

4.1.2. Inverse Neural Network for a water purification process integrated to a heat transformer with energy recycling

To obtain the desired output from the water purification process integrated to a heat transformer with energy recycling it is obvious to choose the correct manipulability variables from the external circuits. An inverted ANN can be considered to be a model-based method of supervisory control, in which the values of the manipulable variables are obtained by solving an on-line optimization problem to obtain the desired output (Kohlenbach, 2006; Lecuona et al. 2009). A general network is shown in Figure 3 is constituted by TANSIG and PURELIN transfer function. Then, the output of ANNi is presented as bellow in order to avoid any ambiguity:

$$y_k = b2_{(l)} + \sum_s \left\{ Tansig\left(b1_{(s)} + \sum_k IW_{(s,k)}.In(k)\right).LW_{(l,s)} \right\} \quad (18)$$

As `tansig` function is used, then y_k is given by

$$y_k = b2_{(l)} + \sum_s \left\{ \left[\frac{2}{1+\exp\left(-2\left(b1_{(s)} + \sum_k IW_{(s,k)}.In(k)\right)\right)} - 1 \right] .LW_{(l,s)} \right\} \quad (19)$$

As $y_k = COP$

According to Equation (19), the Equation (8) could be expressed as follow:

$$COP = b2_{(l)} - \sum_s LW_{(l,s)} + \sum_s \frac{2.LW_{(l,s)}}{1+\exp\left(-2\left(b1_{(s)} + \sum_k IW_{(s,k)}.In(k)\right)\right)} \quad (20)$$

At this step, in general case, we have obtained the function which has to be minimised at zero to find the optimal input parameter(s) $In_{(k=x)}$,since, the parametres in COP's system which we can control are the pressures in the absorber and generator, respectively, are needed for a required coefficient of performance, therefore, the Equation (20) could be expressed as:

$$COP = b2_{(l)} - \sum_s LW_{(l,s)} + \sum_s \left[\frac{2.LW_{(1,s)}}{1+\exp\left(-2\left(IW_{(s,k)}.In_{(x)} + \sum_{k\neq x} IW_{(s,k)}.In_{(k)} + b1_{(s)}\right)\right)} \right] \quad (21)$$

Where x is the unknown optimum operating conditions for a system, which must be found. Nevertheless, in the case that a proposed ANN model has more than one neuron in the hidden layer it is necessary to use an optimization method. In this investigation, the artificial neural network inverse was solved by (GAs) and (PSA), because both methods which use a higher number of initial search points and have a higher probability of finding the global optimum.

4.2. Genetic algorithms

The term genetic algorithms, almost universally abbreviated now a days to GAs, was first used by Jhon Holland (1975). A genetics algorithms is a search techniques used in computing to find exact or approximate solutions to

optimization and search problems, however the canonical steps of the GAs can be described as follows:
Step 1: An initial population is generated randomly, this procedure guarantees a diversified initial population covering the complete space search.
The following steps are performed in order to pass from the actual population (*P*) to the next one (*P+1*). First an intermediate population is generated in step 2 to 4.
Step 2: An elitism procedure selects the best individuals for each criterion, which is then placed in intermediate population.
Step 3: An equal number of individuals are chosen for each criterion using the biased Goldberg's roulette.

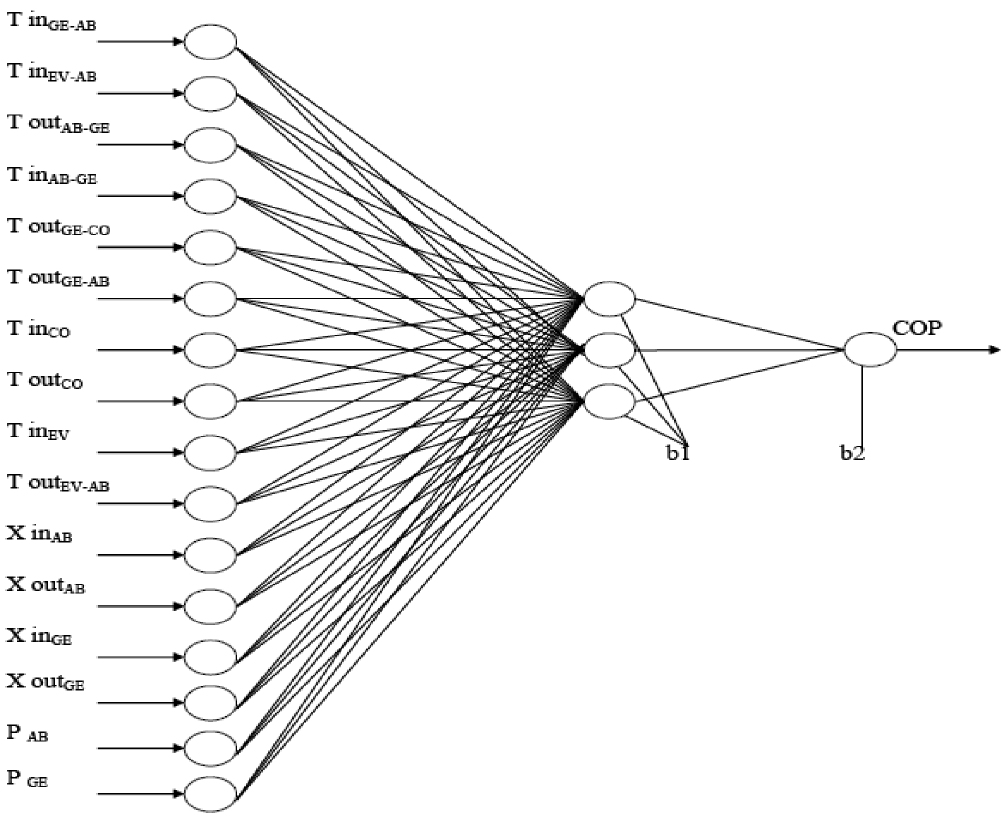

Fig. 3 Model for prediction of COP values

Step 4: The intermediate population is completed with individuals generated by the crossover procedure. Let us note that the individuals to whom this procedure is applied are chosen randomly from the population (*P*).

Step 5: The mutation procedure is applied to a fixed number of individuals chosen randomly. Only one point of the chromosome is modified, changing its value from 0 to 1 or the opposite.
Step 6: The new population becomes the current one and the steps 2 to 5 are repeated until the maximal number of generations is reached.
Any way, the details of GAs process has been addressed by several researchers hence are not repeated in the present paper. More information on GAs process can be found in the publication of Goldbreg (1989).

4.3. Particle swarm algorithms

The particle swarm algorithm (PSA) proposed by Kennedy and others (Kennedy et al, 1995; Shi et al, 1998) has been applied as a simulation of a simplified social system in which individual members of a school can profit from the discoveries and previous experiences of all the other members of the school during the search. A practice swarm in PSA refers to a number of potential solutions to the optimization problem, where each potential solution is refered to as a particle position. Each particle is initialized by a random position in multi-dimensional problem space and then is "flown" through this space to locate the best position. Each particle then moves toward two positions, the best position of each particle (pbest) and the best position obtained so far by the entire particles in the population (gbest), with two weighting coefficients, inertial weights and an acceleration constant, are shown in equations (22) and (23):

$$v_{i,k+1} = wv_{i,k} + c_1.r_1.(y_{i,k} - x_{i,k}) + c_2.r_2.(y_g - x_{i,k}) \tag{22}$$

$$x_{i,k+1} = x_{i,k} + v_{i,k+1} \tag{23}$$

Where v_i represents the velocity of particle i; k stands for the iteration number w represents inertial weight, c_1 and c_2 are acceleration constants, r_1 and r_2 are two random function in the range [0,1], x_i represents the current position of particle i, y_i is the personal best position of particle i (pbest) and y_g is the best position of all the particles found at present (gbest).

4.4. Tuning parameters for GAs and PSA

The efficiency of a GAs and PSA are greatly dependent on its tuning parameters. Using ANNi as the fitness function, GAs and PSA were implemented to optimize the COP's system. they are used to perform global search algorithms to update the input parameters of neural network. The combinations of control parameters used for running GAs and PSA are shown in Table 5.

Table 5. The parameters used for running GAs and PSA

Algorithms	Parameters	Value
GAs	Number of population	49
	Number of generations	1000
	Crossover probablity	0.40
	Mutation probability	0.05
	Elitism	1
PSA	Number of particles	49
	Number of generations	1000
	Inertial weight	1.00
	Acceleration constants	2.00

5. Results and discussion

5.1. Sensitivity analysis

In order to assess the relative importance of the input variables, the process was based on the neural net weight matrix and Garson equation (1991). He proposed an equation based on the partitioning of connection weights:

$$I_j = \frac{\sum_{m=1}^{Nh}\left(\left(\frac{\left|W_{jm}^{ih}\right|}{\sum_{k=1}^{Ni}\left|W_{km}^{ih}\right|}\right)\times\left|W_{mn}^{ho}\right|\right)}{\sum_{k=1}^{Ni}\left\{\sum_{m=1}^{Nh}\left(\frac{\left|W_{km}^{ih}\right|}{\sum_{k=1}^{Ni}\left|W_{km}^{ih}\right|}\right)\times\left|W_{mn}^{ho}\right|\right\}} \tag{24}$$

Where, I_j is the relative importance of the jth input variable on the output variable, N_i and N_h are the number of input and hidden neurons, respectively and W is connection weight, the superscripts $'i'$, $'h'$ and $'o'$ refer to input, hidden and output neurons, respectively. Note that the numerator in equation (24) describes the sums of absolute products of weights for each input. However, the denominator in equation (24), represents the sum of all the weights feeding into hidden unit, taking the absolute values.

In Table 6 shows the relative importance of the input variables calculated by (Eq. (24)).

In fact, the sensitivity analysis (Table 6) showed that all studied input variables have effect on the COP's system. In addition, the concentrations are the most influential parameter, followed by pressures. However, in real time, the concentrations as the $X_{AB\text{-}GE}$ and $X_{GE\text{-}AB}$ cannot be controlled on line, because the system is closed. On the other hand, the pressures have strong effect than the temperatures on the absorption heat transformer. Furthermore, based on the experimental experience was carried out by Hernandez et al, 2009 have proven that the P_{AB} and P_{GE} are the key parameters to control about the performance of the COP values. As for the temperature parameters that appear in the model could modified of their ideal form, and controlled on line in real time, as other example of ANNi's applications. However, in this case it is not done because it would have been enlarged this article very much.

5.2. Adequate value estimated by mean of ANNi

Two different cases are studied here to show the applicability of the method. The desired output or fitness function to be optimised in all case studies is the coefficient of performance, while the number of optimal operating parameters to be calculated differs. We considered constraints to obtain the input(s) parameter(s): the inputs values are in the operation range illustrated in Table 1, and the Equations (25) is a criterion to confirm the optimal values.

In Case 1 the pressure of absorber is the only the manipulable input variable which is calculated to obtain the desired coefficient of performance. In Case 2 the COP's of the system is controlled by two manipulable input variables: the pressure of absorber and generator. In all cases the constraints are considered.

In this case, as mentioned above, the desired output is the COP's of the system with the pressure of absorber (P_{AB}) is an unknown parameter (manipulable variable).

From the Equation (8) for *Case I* with one unknown (P_{AB}) parameter take the following form:

$$COP = b2_{(l)} - \sum_{s} LW_{(l,s)} + \sum_{s} \left[\frac{2.LW_{(1,s)}}{1+\exp\left(-2\left(IW_{(s,k)}.P_{AB} + \sum_{k \neq P_{AB}} IW_{(s,k)}.In_{(k)} + b1_{(s)}\right)\right)} \right] \quad (26)$$

Case 2

The difference between *Case 2* and *Case 1* is that the COP's of the system is controlled by two parameters: P_{AB} and P_{GE}.
Here, the Equation (26) for the two parameters takes a slightly different form, so this equation, can be expressed as follows:

$$Minimize\ COP = Fun\left(P_{AB}, P_{GE}\right) \quad (29)$$

Where

$$Fun(P_{AB}, P_{GE}) = \sum_{s} \left[\frac{2LW_{(l,s)}}{1+e^{-2\left(IW_{(s,15)}P_{AB} + IW_{(s,16)}P_{GE} + \sum_{\substack{k \neq P_{AB} \\ k \neq P_{GE}}} IW_{(s,k)} In_{(k)} + b1_{(s)}\right)}} \right] + \left(b2_{(1)} - LW_{(1,1)} - LW_{(1,2)} - LW_{(1,3)}\right) \quad (30)$$

The parameters of GAs algorithm are set up as follows: The population size is 49, the generation number is 1000, the probability of crossover is 0.40, the probability of mutation is 0.05 and the elitism is 1, concerning, the parameters of PSA
algorithms are: The particle number is 49, the iteration is 1000, the inertial weight is 1, and the acceleration constants are 2.
In practice, however, the calculations required for system are so complicated, that's why the CPU time was calculated on LINUX system; Intel(R)D CPU 2.80 Ghz, 2.99 GB of RAM. According to the Table 7, we can distinguish the following results:

Test B

The optimum results calculated seven times by GAs in test B are compared with PSA as shown in Table 7. In the test B, the required COP is 0.3, optimum P_{AB}
was calculated by GAs is better than PSA about 0.63%. On the other hand, with two variables, the optimum parameters P_{AB} and P_{GE} were calculated by GAs are better than PSA between 0.21% and 0.50%.

Table 6 Relative importance of input variables

Input variable	Importance %
$X_{in.AB}$	9.2134
$X_{out.AB}$	8.1695
$X_{in.GE}$	7.9731
P_{AB}	6.5694
$X_{out.GE}$	5.9002
P_{GE}	5.8800
$T_{outAB-GE}$	5.7163
$T_{out.GE-CO}$	5.6955
T_{outCO}	5.6262
$T_{outEV-AB}$	5.6236
$T_{inAB-GE}$	5.6155
$T_{inEV-AB}$	5.6129
$T_{outGE-AB}$	5.6065
T_{inCO}	5.6050
$T_{inGE-AB}$	5.6043
T_{inEV}	5.5886
Total	**100**

Test C

The optimum results calculated seven times by GAs and PSA in test C. In the test C, the required COP is 0.38, now we would like to compute the optimum value for only one variable input (P_{AB}) given by GAs is better than PSA about 0.75%. Right now, the optimum values for two variables input P_{AB} and $P_{GE,}$ the results given by GAs are better than PSA between 0.63% and 1.63%.

Test E

The optimum results calculated seven times by GAs and PSA in the test E, in this case, the required COP is 0.39, we find that the optimum P_{AB} calculated by GAs is better than PSA about 0.64%. While, the optimum parameters P_{AB} and P_{GE} computed by GAs and PSA, the results show that GAs is better than PSA between 0.81% and 1.98%.

In this context, the Figures 4 and 5, respectively, are shown the behavior of fitness function versus the iterations, It can be seen from those figures that the convergence process results in smooth curves, with a rapid increase at the start that gradually slows down. The experiments were implemented ten times to ensure that the fitness function converges to a minimum value.

The convergence performance of the fitness function to the best value, given by PSA has been convergence in 0.41 in first 450 iterations, in case of one

Although, the convergence performance of GAs, the fitness function to the best value has been convergence in 0.39 in first 325 iterations, but in case the two variables, reaches to the best value has been convergence in 0.39 in 500 iterations. This show the GAs algorithm with the parameter value set has a good performance in convergence.

variable (P_{AB}). However, in case the two variables (P_{AB} and P_{GE}) the convergence of the fitness function is 0.41 in 500 iterations.

In general, from the results shown in Table 7, Figures 4 and 5, respectively when the comparison of the optimization results was made between ANNi-GAs and ANNi-PSA model:

ANNi-GAs can perform better, meaning that GAs is very effective in solving ANNi, and often reached an optimal COP, through the genetic operation such as crossover and mutation. ANNi-PSA is not effective in modeling COP's system, this is due to that population of weights for ANNi-PSA may not be able to reach a global optimum when the evolution was simulated over the same number of generations as with ANNi-GAs model. The higher number of generations may be used but this is not recommended due to longer convergence time.

A comparison of the optimal results between GAs and PSA as shown in Tables and Figures. The results obtained by GAs are better than PSA, taking into account the small error and greatly reduce the computational effort to find the optimum solution over three tests (B, C, E), in addition, GAs results in a faster convergence, hence, we can see that GAs performed effectively and gave a solution within 0.36% of the global optimal, GAs provided also interesting solutions, in terms of quality as well as of computational time, which is due to GAs searching from population (not a single point) including genetics operators and its parallel computing nature could be applied to deal with the complexity of our problem with no more complicated mathematical calculation than such simple operators as encoding, decoding and computing values of objectives. However, the advantage of PSA is simple in structure, only its implementation within tuning parameters as: the particle number, iteration number, inertial weight and the acceleration constants.

These results illustrate, also, that the GAs based on ANNi has a higher optimizing ability than PSA.

Table 7 Comparative results between GAs and PSA

Tests	Variables		Particle Swarm Algorithm				Genetic Algorithms			
			Exp	***Opt***	***Error %***	***CPU Times (s)***	***Exp***	***Opt***	***Error %***	***CPU Times (s)***
Test B	One var.	P_{AB}	9.5	9.41	0.95	40.7	9.5	9.47	0.32	11.3
	Two var.	P_{AB}	9.5	9.54	0.42	43.1	9.5	9.52	0.21	14.7
		P_{GE}	20	19.87	0.65		20	19.97	0.15	
Test C	One var.	P_{AB}	8	8.04	0.5	3.95	8	7.98	0.25	09.5
	Two var.	P_{AB}	8	7.84	0.75	4.32	8	7.97	0.37	13.4
		P_{GE}	20.5	20.41	0.44		20.5	20.54	0.20	
Test E	One var.	P_{AB}	11	10.92	0.73	3.14	11	10.99	0.10	10.9
	Two var.	P_{AB}	11	10.87	1.18	3.24	11	11.09	0.81	16.2
		P_{GE}	21	20.79	1.00		21	20.96	0.19	

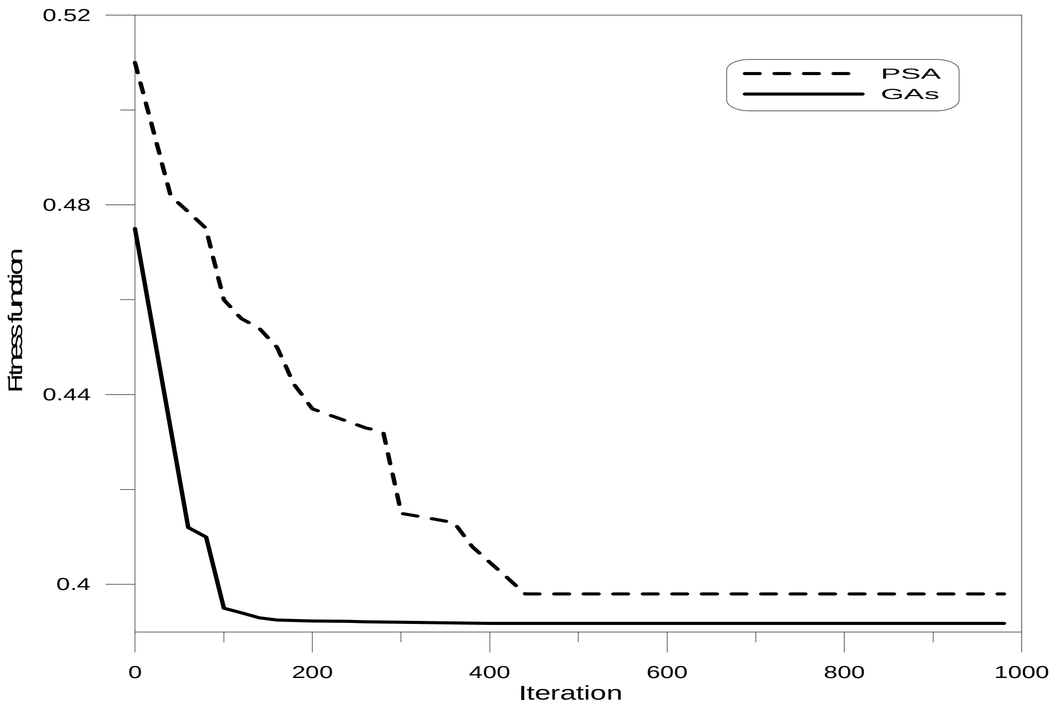

Fig. 4 Fitness function versus iteration for one variable P_{AB}

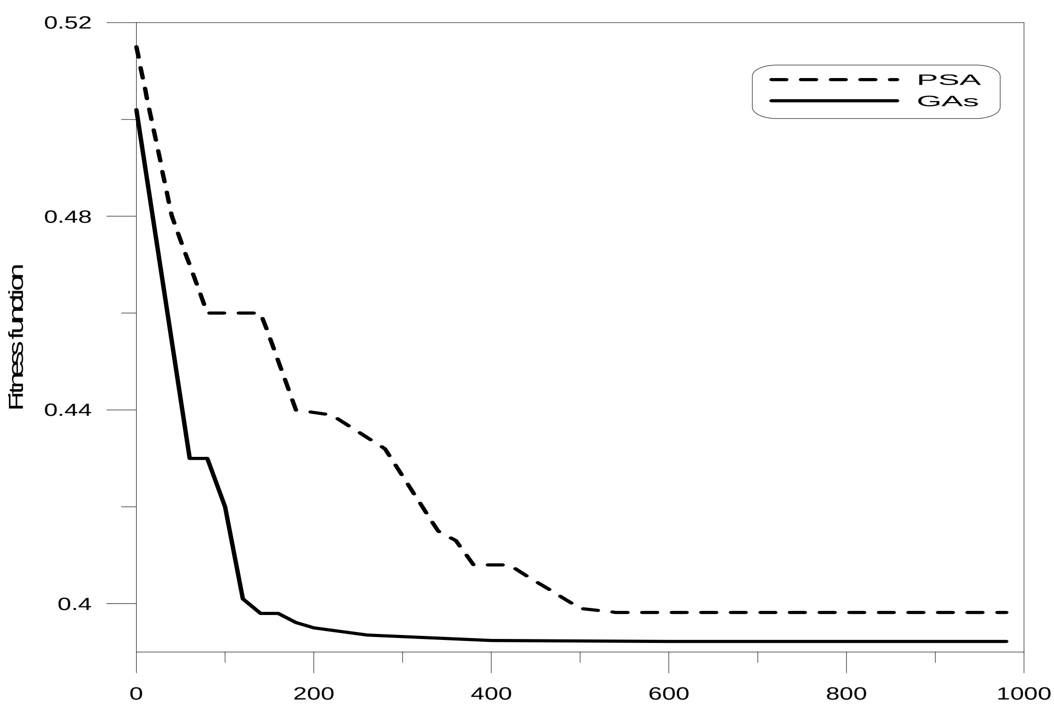

Fig. 5 Fitness function versus iteration for two variables P_{AB} and P_{GE}

6. Conclusions

The coefficient of performance (COP) for a water purification process integrated to an absorption heat transformer with energy recycling was optimized using artificial neural network inverse in order to calculate an ideal input value from an ideal COP and taking into account the above well known input values excepting required input value as P_{AB} and $P_{GE.}$ Then genetic algorithms and particle swarm algorithm were applied in the inverse problem to optimize the optimal operating condition. Further comparison of the optimal results of genetic and particle swarm algorithms shows that the GAs outperforms the PSA, and faster in convergence (<2 seconds), which is sufficiently suitable to direct control of water purification process integrated to the absorption heat transformer considering energy recycling. Then, the genetic algorithms approach described in this paper would be useful to reduce computational effort while optimizing process, and guarantees engineers to arrive at the near optimal solution that could not be easily obtained using general modeling programs or by trial and error. However, these results are important, because they demonstrate the effectiveness of GAs in solving the complicated problem. Finally, the optimization carried out using the trained ANNi integrated with GAs significantly reduced the computational time with better convergence for optimal solution, for enhancing the system performance of the water purification by the condenser of the purification system.
The approach described in this paper would be useful to reduce computational effort while optimizing COP's of the system with large design space.

Nomenclature

b1, b2	bias
COP	coefficient of performance, dimensionless
In	input
K	number of neurons in the input layer
Out	output
P	pressure, inHg
Q	heat flow, Watt
RMSE	root mean square error
S	number of neurons in the hidden layer

X concentration, % w/w

IW, LW matrix weight

Input variables for artificial neural network

$T_{in.GE\text{-}AB}$ input-temperature in the absorber that comes from generator, °C

$T_{in.EV\text{-}AB}$ input-temperature in the absorber that comes from evaporator, °C

$T_{out.AB\text{-}GE}$ output-temperature in the absorber towards generator, °C

$T_{in.AB\text{-}GE}$ input-temperature in the generator that comes from absorber, °C

$T_{out.GE\text{-}CO}$ output-temperature in the generator towards condenser, °C

$T_{out.GE\text{-}AB}$ output-temperature in the generator towards absorber, °C

$T_{in.CO}$ input-temperature of the condenser that comes from generator, °C

$T_{out.CO}$ output-temperature in the condenser towards evaporator, °C

$T_{in.EV}$ input-temperature in the evaporator that comes from condenser, °C

$T_{out.EV\text{-}AB}$ output-temperature in the evaporator towards absorber, °C

P_{AB} and P_{GE} pressure in absorber and generator, respectively, in Pascal (Pa)

$X_{in.AB}$ LiBr input-concentration in the absorber, % w/w

$X_{out.AB}$ LiBr output-concentration in the absorber, % w/w

$X_{in.GE}$ LiBr input-concentration in the generator, % w/w

$X_{out.GE}$ LiBr output-concentration in the generator, % w/w

Sub-Index

AB absorber

CO condenser

EV evaporator

EXP experimental

GE generator

SIM simulated

References

Chang C., Xu G., Yang J., Wang D., Optimization of Cellulase Production using Agricultural Wastes by Artificial Neural Network and Genetic Algorithm, Chemical Product and Process Modeling, 6 (2011) 1-16.

Clauß V., Kühn A., Ziegler F., A new control strategy for solar driven absorption Chillers, 2nd International Conference Solar Air Conditioning Tarragona, Spain, (2007) 44-50.

El Hamzaoui Y., Hernandez J.A., Chavez Cruz M.A., Bassam A., Search for Optimal Design of Multiproduct Batch Plants under Uncertain Demand using Gaussian Process Modeling Solved by Heuristics Methods, Chemical Product and Process Modeling, 5 (2010) 1-22.

Garson G.D., Interpreting neural-network connection weights, AI Expert 6 (1991) 47–51.

Goldberg D.E., Genetic algorithms in search optimization and machine learning. Addison Wesley., 1989.

Hagan M.T., Menhaj M.B., Training Feedforward Networks with the Marquardt Algorithm, IEEE Transactions on Neural Network, 5 (1994) 989–993.

Hamidipour M., Mostoufi N., Rahmat Sotudeh-Gharebagh R., Modeling the synthesis section of an industrial urea plant, Chemical Engineering Journal, 106 (2005) 249-260.

Hernández J.A., Bassam A., Siqueiros J., Romero D.J., Optimum operating conditions for a water purification process integrated to a heat transformer with energy recycling using neural networks inverse, Renewable Energy, 34 (2009) 1084-91.

Hernández J.A., Romero D.J., Morales L.I., Siqueiros J., COP prediction for the integration of a water purification process in a heat transformer: with and without energy recycling, Desalination, 219 (2008) 66–80.

Holland J.H., Adaptation in Natural and Artificial Systems. MIT Press., 1975.

Huicochea A., Siqueiros J., Improved efficiency of energy use of heat transformer using a water purification, Desalination, 257 (2010) 8-15.

Huicochea A., Siqueiros J., Romero R.J., Portable water purification system integrated to a heat transformer, Desalination, 165 (2004) 385–91.

Iranshahi A., Hakimelahi H.R., Sotudeh-Gharebagh R., Mostoufi N., Simulation of an acid-based starch converter, Chemical Engineering and Technology, 27 (2004) 569-577.

Kennedy J., Eberhart R.C., Particle swarm optimization. IEEE Proceeding, 4 (1995) 1942– 1948.

Kohlenbach P., Control Strategies and Transient Chiller Performance. PhD thesis, Technical University of Berlin, Forschungsberichte des Deutschen Kälte und Klimatechnischen Vereins, Nr. 74, Erding, Germany, 2006.

Kůrková V., Kolmogorov's theorem and multilayer neural networks, Neural Networks, 5 (1992) 501-506.

Lagarias J.C., Reeds J.A., Wright M.H., Wright P.E., Convergence properties of the Nelder–Mead simplex method in low dimensions, SIAM Journal of Optimization, 9 (1998) 112–147.

Lecuona A., Ventas R., Venegas M., Zacarías A., Salgado R., Optimum hot water temperature for absorption solar cooling, Solar Energy, 83 (2009) 1806-1814.

Mehrpooya M., Vatani A., Mousavian S.M., Optimum design of integrated liquid recovery plants by variable population size genetic algorithm, The Canadian Journal of Chemical Engineering, 88 (2010) 1054-1064.

Morales G.L.I., Estudio experimental sobre un sistema portátil de purificación de agua integrado a un transformador térmico, MSc Thesis, CIICAp-UAEM. México, 2005.

Mostoufi N., Sotudeh-Gharebagh R., Ahmadpour M., Simulation of an industrial pyrolysis gasoline hydrogenation unit, Chemical Engineering and Technology 28 (2005) 174-181.

Ravagnani M., Silva A.P., Arroyo P.A., Constantino A.A., Heat exchanger network syntesis and optimization using genetic algorithm, Applied Thermal Engineering 25 (2005) 1003–1017.

Shi Y., Eberhart R.A., Modified Particle Swarm Optimizer, IEEE Proceeding, (1998) 69–73.

Siqueiros J., Romero R.J., Increased COP for heat transformer in water purification systems. Part I: temperature increase for heat source, Appliel Thermal Engineering, 27 (2007) 1043–53.

Sotudeh-Gharebagh R., Mostoufi N., Simulation of a catalytic turbulent fluidized bed reactor using the sequential modular approach, Fuel Processing Technology 85 (2004) 189-200.

Torres M.J., Contacteurs gaz-liquide pour pompes á chaleur á absorption multi-étagées. Ph.D Thesis, Institut National Polytechnique de Lorraine, 1997. France.