Joint strength prediction in a pulsed MIG welding process using hybrid neuro ant colony-optimized model

N. Raghavendra · Rakshit Koranne · Sukhomay Pal · Surjya K. Pal* · Arun K. Samantaray

Abstract

In this work, a pulsed metal inert gas welding process (PMIGW) is modeled by using a hybrid soft computing technique. Ant colony optimization and back propagation neural network models are combined to predict the ultimate tensile strength of butt-welded joints. A large number of experiments have been conducted; and comparative study shows that the hybrid neuro ant colony-optimized model produces faster and also better weld-joint strength prediction than the conventional back propagation model.

Keywords Back propagation neural network, Ant colony optimization, PMIGW, Weld strength.

1 Introduction

Pulsed metal inert gas welding (PMIGW) is a type of arc welding process which offers certain inherent advantages such as deep penetration, smooth weld bead, high welding speed, large metal deposition rate, lower spatter, lower distortion and shrinkage, and

* Corresponding author

N. Raghavendra · Rakshit Koranne · Sukhomay Pal · Surjya K. Pal (✉) · Arun K. Samantaray
Department of Mechanical Engineering,
Indian Institute of Technology Kharagpur,
721 302, India
Tel.: +91 3222 282996
Fax: +91 3222 255303

lesser fusion defects [1]. Although PMIGW has found wide spread acceptance in the industry, stricter quality control norms being followed now-a-days require a robust and reliable method to precisely predict the weld quality in different operating conditions. A lot of research has been carried out for online weld quality prediction; but a low cost, reliable and easily deployable monitoring system for industrial use is yet to be developed.

Weld quality can be measured directly or indirectly. Direct methods are visual inspection and vision sensing [2, 3] of the weld puddle; indirect methods are arc sensing [4-6], infrared sensing [7, 8], radiographic sensing [9], inductive sensing [10], arc sound, and acoustic and ultrasonic sensing [11-13]. Among the various sensors used, arc sensors, i.e., current and voltage sensors, are considered to be the most reliable, simple and competitive [14, 15].

Arc sensors monitor the electrical parameters of the arc, i.e., current and/or voltage. A large number of researchers [1, 4-6, 16] have proposed arc sensing technique for seam tracking in the arc welding process. In addition to this, arc sensing technique has been used for online monitoring and control of the welding process [17].

Johnson *et al*. [18] conducted a series of experiments by using two different power sources and three different metal transfer modes. In their experiment, audible range of the sound, welding current and voltage fluctuations were recorded. These recorded signals were then correlated to detect droplet transfer mode with the aid of the high-speed film data. Rajasekaran *et al*. [19] determined the droplet detachment in a pulsed gas metal arc welding (GMAW) process by using current and voltage signals. Time-domain analysis of the voltage signal for monitoring the welding quality in a short circuit GMAW process has been developed by Adolfsson *et al*. [20]. Wang *et al*. [21]

monitored high frequency and hybrid pulsed tungsten inert gas micro-welding process through arc sensing technique wherein the mean voltage, the probability density distribution and the dynamic voltage-current graph of the arc were used to determine the weld penetration. Chu *et al*. [22] analyzed the current and voltage signals by using power spectral density and time-frequency domain analysis for welding stability and weld quality monitoring in a short circuit GMAW process.

Welding is a highly complicated and nonlinear process; which is influenced by many variables, such as process parameters, composition of workpiece and electrode materials, shielding gas environment, welding position etc. As a consequence, it is very difficult to obtain an analytical model, which can accurately predict the weld quality from the arc or any other sensor data. Artificial neural network (ANN) model can effectively be used to map the nonlinear relationship between the sensor signal features and the weld quality. Andersen *et al.* [23] pioneered the application of artificial neural network in the modeling of the arc welding process. Cook *et al.* [24] have used two back propagation network models for modeling and control of variable polarity plasma arc welding process and obtained good agreement with the experimental outputs. Kang *et al*. [25] developed an ANN model to select welding parameters, such as welding current, arc voltage, welding speed and weaving length for the required weld bead shape specification. Lee & Um [26] predicted the geometry of back-bead of MIG-weld plates by using an ANN model and then compared the result with that from a multiple regression analysis model to demonstrate that the prediction error from the ANN model was less than that of the multiple regression model. Chi & Hsu [27] developed a fuzzy radial basis function neural network to predict weld quality characteristic of a plasma arc welding process. Di *et al.*

[28] developed an ANN-based fuzzy logic control for fine tuning of the membership function and automatic fuzzy rules generation in modeling of an arc welding process. Other researchers [29, 30] have also developed back propagation neural network models to predict the bead geometry and weld penetration. Lightfoot *et al.* [31] predicted the distortion in welded plates by using an ANN model, in which the standard deviation was considered as the measure of the actual and the predicted distortions. Kim *et al.* [32] developed an intelligent system for automatic determination of the optimal welding parameters through a back propagation neural network (BPNN) models which was validated by comparing the results from a finite element model (FEM).

Quero *et al.* [33] used current signal as one of the inputs of the ANN model to monitor the weld quality. In another significant development, Ohshima *et al.* [34] proposed a neuro-arc sensor model to simultaneously detect the deviation, the attitude and the height of the torch.

Selection of optimum welding parameters is very essential to get a good weld quality. Many researches have been carried out for optimization of different welding processes. Ant colony optimization (ACO) algorithm, which is a non-traditional optimization technique, is inspired by the foraging behavior of real ants, which often follow the shortest path to the source of food and then back to nest. ACO has an advantage that it can find both the local and the global minima [35].

ACO algorithm can be used in the training of the neural network to achieve better convergence. Su-bing and Ze-min [36] applied this hybrid neuro-ACO technique in ATM (Asynchronous Transfer Mode) network traffic control. Blum and Socha [37] applied this novel hybrid technique for pattern classification and showed that this algorithm is comparable to specialized algorithms for neural network training, and it has many advantages over other general purpose optimizers. Liu *et al* [38] have used this hybrid technique to predict coal ash fusion temperature based on the chemical compositions of the ash. This hybrid neuro-ACO model is not yet explored in the field of weld quality prediction or monitoring. In this paper, an attempt has been made to evaluate the efficiency of the hybrid ACO-BPNN model for prediction of the welded-joint strength. Time-domain features of the arc signals are acquired through current and voltage sensors during the butt welding of mild steel work-pieces in a PMIGW process and those are used along with the welding process parameters as inputs of the developed model.

2 Soft Computing models

2.1 Back propagation neural network

Artificial neural network resembles the biological neural network and the neurons are logically interconnected in layered patterns, as shown in Fig. 1. ANN maps the input with the output without any conventional equation. It also possesses the property of fault tolerance i.e. it can predict accurately even if a few training data are faulty.

A feed-forward neural network governed by a back propagation training algorithm is the most versatile model of ANNs. Single hidden layer feed forward network has a layered structure with an input, hidden and output layers as essential constituents of the network. The layers are in turn made up of units called nodes, which are connected in such a way that a node in a particular layer has connections to all the nodes of the adjacent layers. The number of input and output nodes depends on the problem in hand, while the number of hidden nodes depends on the complexity of the problem. The knowledge is presented through the interconnecting weights, which are adjusted during the learning stage to produce minimum mean square error.

A bias element, with unit output, is connected to all the nodes of the network. The weights associated with the bias interconnection act as a threshold function on the combined inputs of each node. Every node is fed with the weighted sum of all the outputs of the nodes in the previous layer, and produces an output through an activation function. The activation function $f(x)$ provides a nonlinear relationship between input and output vectors. The weighted sum and the activated output from the j^{th} node are, respectively, represented by

$$x_j = \sum_{i=1}^{n} y_i w_{ij} \,, \tag{1}$$

and

$$O_j = f(x_j)\,, \tag{2}$$

where y_j is the i^{th} input form the previous layer to the j^{th} node of the next layer, x_j is the weighted sum of all inputs at the j^{th} node, O_j is the activated output from node j, and n is the total number of nodes present in a layer.

This process is carried out for all nodes, and the input vector is mapped nonlinearly onto the output vector. The input vectors from the training pattern are fed through the neural network one by one and the output vectors are obtained. These are compared with the expected output vectors, which are already known from the experimental results. The difference between the expected output and the output from the neural network is the error associated with that input vector:

$$error(j) = d(j) - O(j) , \tag{3}$$

where $error(j)$, $d(j)$ and $O(j)$ are, respectively, the associated error, the desired output and the neural network output from the j^{th} output node.

The average squared error energy (*ASEE*) is then calculated as

$$ASEE = \frac{1}{N_{train}} \sum_{N} \sum_{j=1}^{C} \frac{1}{2} error(j)^2 , \tag{4}$$

where N_{train} and C are the number of training patterns and the number of neurons in the output layer, respectively.

For a particular training set, *ASEE* represents the cost function as the measure of the training performance [39]. The synaptic weights are adjusted using back propagation algorithm based on gradient descent method to minimize the *ASEE*.

2.1.1 Pseudo code for BPNN

Step 1 : *Definition*

- The ANN architecture is defined. Type of the activation function, values of the learning rate and the momentum parameter are assigned.

Step 2 : *Initialization*

- The synaptic weights and biases are initialized in this step.

Step 3 : *Forward pass*

- The output from each node is calculated by applying the user defined activation function to the weighted sum of a node. This becomes the input to the next layer.
- This process is carried out first for each hidden node, and then for each output node.

- The difference between the actual output and the model's prediction, i.e. the error for that output node, is calculated.

Step 4 : *Backward pass*

- All the synaptic weights are updated according to the gradient descent algorithm.

Step 5 : *ASEE Calculation*

- **Steps 3 and 4** are repeated for the entire input vector.
- *ASEE* is calculated.

Step 6 : *Termination*

- **Steps 3**, **4** and **5** are repeated until the *ASEE* converges within a user defined error tolerance.

Step 7 : *Return*

- The updated weights for the particular architecture are returned by the procedure.

Step 8 : *End*

- The BPNN training terminates.

The aforementioned BPNN training algorithm is shown in a flowchart form in Fig. 2.

2.2 Ant colony optimization

Ant colony optimization algorithm was proposed by Dorigo *et al.* [40]. ACO is one of the swarm intelligence algorithms, which are based on the collective behavior of decentralized and self-organized systems, such as ants, flock of birds etc. Self-organized ant systems provide highly coordinated behavior of real ants. Among different aspects of the ant colony behaviour, foraging behavior inspires the ACO algorithm. Ants in a colony coordinate their activities via stigmergy, which is a form of indirect communication. Ants follow the shortest path to the source of food and then back to the nest. The principle behind this phenomenon is that the ants secrete pheromone, an odorous chemical substance which ants deposit and smell, on the path they move; thus marking a trail on the path followed. A new ant, which encounters a trail, prefers to move in a path having higher pheromone density. This is an emergent behavior resulting from each ant's preference to follow trail pheromones deposited by other ants. The pheromone evaporates at a constant rate. Pheromone evaporation can be seen as an exploration mechanics that avoids quick convergence of all the ants towards a suboptimal path. The shortest path is followed with a high probability, i.e. more ants move in this path. So the pheromone concentration becomes high on the shortest path. Finally, all the ants in the colony follow the same path (shortest path).

In ACO algorithm, a starting node is selected randomly, and the path is probabilistically selected according to the amount of pheromone present on the possible paths from the starting node. The ant then reaches the next node and selects the next path. This process continues until the ant reaches the destination node. The completed tour of the ant is analyzed for minimization of distance traveled. The trail is adjusted such that the better solution gets a higher trial than the weaker solution. This cycle is repeated until the convergence is reached (i.e. most ants choose the same path on every cycle).

ACO algorithms are mostly used to solve minimization problems. The ant traces the path in the forward pass and remembers the path followed. Pheromone is deposited while tracing back (backward pass) a previously followed path. In the beginning of the search process, pheromone is assigned to all the paths randomly between 0 and 1.

The probability of k^{th} ant, located at i^{th} node to choose j^{th} node as the next node is given by

$$P_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}}{\sum_{l \in N_i^k} \tau_{ij}^{\alpha}}, & \text{if } j \in N_i^k; \\ 0, & \text{if } j \notin N_i^k; \end{cases} \tag{5}$$

where N_i^k is the neighborhood of the k^{th} ant when at the l^{th} node, and τ_{ij} is the pheromone value between the i^{th} and the j^{th} nodes.

In the path traversed by the k^{th} ant, the pheromone value is changed from τ_{ij}^{old} to τ_{ij}^{new} as follows

$$\tau_{ij}^{new} = \tau_{ij}^{old} + \Delta \tau^{k} \tag{6}$$

Where $\Delta \tau^{k} = 1/L_{ij}$ and L_{ij} is the distance between the i^{th} and the j^{th} nodes.

After all the ants have moved to the next node, the pheromone evaporates according to

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old}, \tag{7}$$

where ρ is the pheromone evaporation rate.

2.3 Hybrid ant colony-optimized back propagation neural network (ACO-BPNN)

Ant colony optimization has already been successfully applied to many complex optimization problems [41]. In BPNN, the initial weights are generated randomly, so that the behavior of back propagation (BP) algorithm is inconsistent. In a hybrid ACO-BPNN algorithm, the best values for the initial weights are achieved by using ACO; and then the

BP algorithm is used to get their optimized values [36-38]. Better and steady convergence can be achieved by using this hybrid algorithm.

In this hybrid ACO-BPNN algorithm, pheromones are equivalent to the synaptic weights of the ANN model. Now, for a specified number of ants (say *m*) equally distributed amongst the input nodes are allowed to traverse along the network one by one. All the nodes where a particular ant arrived during its journey towards output nodes are recorded. While an ant moves from a node in one layer to a node in the next layer, the selection of destination node is done probabilistically according to the following equation,

$$P_{ij}^{k} = \begin{cases} \dfrac{w_{ij}}{\sum_{l \in N_i^k} w_{il}}, & \text{if } j \in N_i^k; \\ 0, & \text{if } j \notin N_i^k; \end{cases} \tag{8}$$

where $P^k{}_{ij}$ is the probability of k^{th} ant moving from i^{th} node in the present layer to j^{th} node in the next layer, and w_{il} is the weight of the synaptic link between i^{th} node in the present layer to l^{th} node in the next layer.

As ants move to the next node, the deposited pheromone evaporates according to the Eq. (7). After each ant reaches the output layer, they start traveling back in the same path laying down the pheromone trails which depends on the pheromone secretion rate (that is assumed to be same for every ant). The weights are updated (w_{ij}^{new}), according to the maximum error associated with the particular output node, where k^{th} ant arrived during its forward journey (calculated previously), according to the following equation:

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w^k, \tag{9}$$

where, w_{ij} is the weight of the synaptic link between the i^{th} node in the present layer to the j^{th} node in the next layer, and

$$\Delta w^k = \frac{\gamma}{\max\_error(l)}, \tag{10}$$

where γ is a positive constant [36] used to control the pheromone adjustment rate and $\max\_error(l)$ is the maximum error for l^{th} output node where the k^{th} ant had finished its forward pass.

This procedure is continued until a particular number of iterations (predefined) have been executed. Subsequently, these sub-optimized weights are used as the initial weights in the BP algorithm.

2.3.1 Pseudo code for ACO-BPNN

Step 1 : *Definition*

- The architecture, activation function, learning rate, momentum parameter, and pheromone secretion and evaporation rates are defined in this stage.

Step 2 : *Initialization*

- The following variables are initialized in this step: ants per node (m), weights (pheromone) for each synaptic link (τ), pheromone evaporation rate (ρ), pheromone secretion rate (λ), Tabu matrix [42] to store the path followed by ant, momentum parameter, learning rate, and activation constants.

Step 3 : *ACO-Forward pass*

- The output of each node is calculated by applying an activation function to the weighted sum of its inputs. This acts as input to the next layer.

- This process is carried out for each hidden node and then for each output node, to get to the final output.
- The error between the actual output and the computed output for each output node, and for each training set is calculated.
- Then the maximum error for each output neuron is calculated.

Step 4: *Traversal of the ants*

- For all the ants, the first elements of Tabu matrix with their starting nodes are set.
- The probability with which an ant can travel, for each synaptic link, is calculated.
- By following roulette wheel selection [39], the destination-hidden node and the output node for each ant are found and then those values are set as the second and the third elements of the Tabu matrix, respectively.

Step 5: *Pheromone update*

- All the synaptic weights, considering pheromone evaporation, are updated by

$$w_{new} = w_{old}(1-\rho). \tag{11}$$

- The weights of the path followed by all the ants are updated according to Eq.(9).

Step 6: *Termination*

- **Steps 3, 4** and **5** are repeated until the termination criterion is satisfied.
- These weights are then returned to the back-propagation algorithm

Step 7: *BPNN training*

- The neural network model is then trained by using a back propagation training algorithm which is initialized with the weights determined from **step 6**.

Step 8: *End*

- The neural network training is terminated upon achieving sufficient accuracy.

A flowchart representation of the ACO-BPNN algorithm is shown in Fig. 3.

3 Experimental procedure

3.1 Specimen preparation

In the present work, two mild steel specimens, with dimensions of 125 mm × 100 mm × 8 mm of each were used as the workpiece. Optical emission spectroscopy (OES) has been done to find out the chemical composition of the base metal, and it is shown in Table 1. These specimens were prepared with a V-shaped groove, as shown in Fig. 4, where the groove angle, the root face and the root gap were 30^0, 2 mm and 2 mm, respectively. Thereafter, 53 pairs of such specimen with constant groove angle and root face were prepared, and faces were cleaned by a surface grinder. To make a butt joint, two plates were tacked at the two ends along the width, with a constant root gap of 2 mm. Once the welding is over, all weld plates were cut by using a DoAll Counter band saw cutter machine, to a required shape for conducting tensile test, as shown in Fig. 5. Tensile tests were conducted at room temperature by using a universal testing machine (Losenhausenwerk, Germany) on a 30 Ton scale.

3.2 Equipment

A Fronius make MIG welding machine is used in the present study. The power source is a constant voltage source, Transarc 500, and the control unit is of VR131 type. A schematic diagram of the experimental setup is shown in Fig. 6.

The welding torch or welding gun (AW502) was mounted on a fixed arm. Mild steel plates were clamped on a motor-driven carriage with a variable speed in the range of 1 mm/sec to 16 mm/sec. Copper coated mild steel wire of 1.2 mm diameter is used in the experiment as the electrode. This wire is fed through the welding gun by a four-roller drive system. The shielding gas (argon, in this case) was supplied in a regulated manner at a constant flow rate of 15 l/min and at a constant pressure of 10 kgf/cm^2.

A Hall-effect current transducer (LEM, LT 500S) was used to monitor the welding current. Moreover, the potential difference was sensed between the workpiece and the tip of the electrode. The voltage across the electrodes was scaled down in 1:11 ratio before being fed to the data acquisition card (Measurement computing corporation, PCI-DAS 4020/12). The analog outputs from these sensors were converted into digital signals by the A/D card fitted to an IBM PC. The signals were sampled at 10 kHz and the magnitude of the sensor outputs were measured in $\pm$ 5V range.

3.3 Experimental design

Response surface method [43] was used to understand the effect of different process parameters on the weld bead geometry. Three levels, six factors and half fraction central composite experimental design with nine center points was performed. This design requires fifty-three experimental runs. Software package MINITAB [44] was used to setup the design matrix. Process parameters with their notations, units and values at different levels are listed in Table 2. The experimentally obtained values of the ultimate tensile strengths, which correspond to various process parameter settings, are shown in Table 3.

4 Results and Discussion

4.1 Optimum model parameter selection

In a model, optimal values of important parameters must be used to get accurate predictions. The learning rate and the momentum coefficient are two important parameters of a BPNN model. Likewise, pheromone secretion and evaporation rates are important parameters of the hybrid ACO-BPNN model. At higher secretion rates, weights change by a large amount; and larger number of epochs (steps in the training process of an artificial neural network) are required. On the other hand, if the secretion rate is low then the weights will be updated at a slow rate, thereby leading to higher number of epochs. Therefore, one needs to select optimum values for both these parameters so as to obtain good results from any particular data set. From a pool of 53 patterns available from experiments, 43 patterns were randomly selected for training and the rest were used for testing the neural network by using ACO-BPNN algorithm. The results obtained for ACO-BPNN while varying the pheromone secretion rate are shown in Table 4, and those obtained while varying pheromone evaporation rate are shown in Table 5. This study was carried for the 8-13-1 (8 input nodes, 13 hidden nodes, and 1 output node) network architecture and 7 ants were allowed to pass through each node. This architecture was chosen because it had shown the best performance for prediction of welding joint strength. All other network parameters were kept constant**.** The variation of mean percentage error with pheromone secretion and pheromone evaporation rates are plotted in Figs. 7 and 8**,** respectively. From this study, the optimum values of pheromone secretion and evaporation rates for ACO-BPNN model are chosen as 0.1 and 0.01, respectively.

4.2 Comparative study

The same 43 experimental patterns selected earlier were used for training a BPNN model and the rest of the patterns were used for testing both BPNN and ACO-BPNN models. The common network parameters for both the models, i.e. the momentum coefficient, the learning rate and the constants of the activation function, were kept same. The optimum values of pheromone secretion rate and pheromone evaporation rate were selected for ACO-BPNN. The selected values of all these parameters are given in Table 6. The number of input nodes and number of output nodes for both BPNN and ACO-BPNN models are 8 and 1, respectively. The number of hidden nodes is varied so as to choose the best network architecture. The number of ants traversing from each input node in ACO-BPNN model is also varied to choose the optimum number of ants to be passed in order to obtain good performance for a particular architecture. The results obtained for BPNN and ACO-BPNN models, by using the selected parameter values, while varying the hidden nodes, is shown in Table 7**.** For comparing the performance of different possible architectures, all these architectures used in this work are trained up to a fixed value of ASEE (0.002, in this case). The variations of ASEEs, showing the convergence rates, with the number of epochs for training of the neural network by using ACO-BPNN and BPNN algorithms are shown together in Fig. 9. The variations of mean error against the number of hidden nodes for both the models are shown in Fig. 10**.** These results show that the best weld strength prediction is obtained from an 8-4-1 network architecture for the BPNN model and from an 8-13-1 network architecture for ACO-BPNN model; whereas, 8-10-1 and 8-11-1 network architectures give the fastest convergence of ASEEs in case of ACO-BPNN model and BPNN model, respectively.

From the results obtained, it may be concluded that the ACO-BPNN algorithm converges at faster rate than the BPNN algorithm. Moreover, it is observed that the mean percentage error in predictions from the ACO-BPNN model is less than that from the BPNN model.

5 Conclusions

In this work, the ultimate tensile strengths of butt-weld joints in mild steel plates have been predicted by using a novel hybrid technique, called ACO-BPNN. Experiments were carried out for a large number of mild steel plates by varying six process parameters; namely, the background voltage, the pulse voltage, the pulse frequency, the pulse duty factor, the wire feed-rate and the table feed-rate. Time-domain features of arc signals, i.e. current and voltage, were used along with those six process parameters as the inputs to the hybrid model. Different network architectures were also considered in this work.

It is observed that the ultimate tensile strength prediction performance of the neuro-ACO model is superior to the conventional BPNN model. The optimum network architectures and the number of ants, giving the best prediction performance and faster convergence during network training, have also been obtained.

References

1. Amin M (1983). Pulsed current parameters for stability and controlled metal transfer in arc welding. Metal construction 15(5): 272-278.
2. Sweet LM (1985). Sensor-based control systems for arc welding robots. Robotics and computer integrated manufacturing 2: 125-133.
3. Sforza P, Blasiis DD (2002). On-line monitoring system for arc welding. NDT&E international 35: 37-43.
4. Munezane Y, Watanabe T, Iochi A, Sekino T (1987). A sensing system for arc welding. J.D. Lane (Ed.), Robotic welding, IFS Publications ltd., London 265-274.
5. Cook GE, Andersen K, Fernandez KR, Shepard ME, Wells AM (1987). Electric arc sensing for robot positioning control. J.D. Lane (Ed.), Robotic welding, IFS Publications ltd., London 182-216.

6. Hughes RV, Walduck RP (1987). Electromagnetic arc path control I plasma welding. J.D. Lane (Ed.), Robotic welding, IFS Publications ltd., London. 2244-2263.
7. Nagarajan S, Chen WH, Chin BA (1989). Infrared sensing for adaptive arc welding. Weld. J 68: 462s-466s.
8. Chen WH, Chin BA (1990). Monitoring weld penetration using infrared sensing techniques. Weld. J 69: 181s-185s.
9. Guu A, Rokhlin S (1992). Arc weld process control using radiographic sensing. Mater. Eval 50: 1344-1348.
10. International institute of welding (1985). Automation and robotisation in welding and allied processes, Pergamon press, Oxford.
11. Grad L, Grum J, Polajnar I, Slabe JM (2004). Feasibility study of acoustic signals for on-line monitoring in short circuit gas metal arc welding. Int. J. Mach. Tools Manuf. 44: 555-561.
12. Saini D, Floyd S (1998). An investigation of gas metal arc welding sound signature for on-line quality control. Weld. J 77: 172s-179s.
13. Carlson NM, Johnson JA (1988). Ultrasonic sensing of weld pool penetration, Weld. J 67: 239s-246s.
14. Li D, Yonglun S, Feng Y (2000). On line monitoring of weld defects for short-circuit gas metal arc welding based on the self-organize feature map neural networks. Proc. Int. Joint Conf. Neural Networks 5: 239-244.
15. Siewert T, Samardzic I, Klaric S (2002). Application of an on-line weld monitoring system, Proceedings of the 1st Int. Conf. on Advanced Technologies for Developing Countries, Slavonski Brod, Croatia.
16. Cook GE (1981). Feed back and adaptive control in automated arc welding systems. Metals construction 13: 551-556.
17. Quinn TP, Smith C, Mccowan CN, Blachowiak E, Madigan RB (1999). Arc sensing for defects in constant voltage gas metal arc welding. Weld. J 79: 322s-328s.

18. Johnson JA, Carlson NM, Smartt HB, Clark DE (1991). Process control of GMAW: sensing of metal transfer mode. Weld. J 70: 91s–99s.

19. Rajasekaran S, Kulkarni SD, Mallya UD, Chaturvedi RC (1998). Droplet detachment and plate fusion characteristics in pulsed current gas metal arc welding. Weld. J 78: 254s-269s.

20. Adolfsson S, Bahrami A, Bolmsjo G, Claesson I (1999). On-line quality monitoring in short-circuit gas metal arc welding. Weld. J 78: 59s–73s.

21. Wang J, Kusumoto K, Nezu K (2003). Microweld penetration monitoring techniques by arc sensing. Proc. Int. Conf. on advanced intelligent mechatronics, IEEE/ASME: 1027-1030.

22. Chu YX, Hu SJ, Hou WK, Wang PC, Marin SP (2004). Signature analysis for quality monitoring in short circuit GMAW. Weld. J 83: 336s-343s.

23. Andersen K, Cook GE, Karsai G, Ramaswamy K (1990). Artificial neural networks applied to arc welding process modeling and control. IEEE Transactions on Industry Applications 26: 824-830.

24. Cook GE, Barnett RJ, Andersen K, Strauss AM (1995). Welding modeling and controlling using artificial neural networks. Industry applications, IEEE Trans 31: 1484-1491.

25. Kang SI, Kim GH, Lee SB (1999). A study on the horizontal fillet welding using neural networks, 3rd Int. conf. on knowledge-based intelligent information Eng. Sys., Australia: 217-221.

26. Lee J, Um K (2000) A comparison in a back-bead predication of gas metal arc welding using multiple regression analysis and artificial neural networks, Optical and Laser in Eng. 34: 149-158.

27. Chi SC, Hsu LC (2001). A fuzzy radial basis function neural network for predicting multiple quality characteristics of plasma arc welding, IFSA/NAFIPS, Vancouver Canada: 2807-2812.

28. Di L, Srikanthan T, Chandel RS, Katsunori I (2001). Neural-network based self-organized fuzzy logic control for arc welding. Engineering application of artificial intelligence 14: 115-124.

29. Nagesh DS, Datta GL (2002). Prediction of weld bead geometry and prenetration in shielded metal-arc welding using artifical neural networks. J. Mater. Process. Technol. 123: 303-312.

30. Kim IS, Son JS, Lee SH, Yarlagadda, PKDV (2004). Optimal design of neural networks for control in robotics arc welding. Robotics and computer-integrated manufacturing 20: 57-63.

31. Lightfoot MP, Bruce GJ, McPherson NA, Woods K (2005). The application of artificial neural networks to weld-induced deformation in ship plate. Weld. J 84: 23s-30s.

32. Kim IS, Jeong YJ, Lee CW, Yarlagadde PKDV (2006). Prediction of welding parameters for pipeline welding using an intelligent system. J. of advanced manufacturing technology 22: 713-719.

33. Quero JM, Millan RL, Franquelo LG (1994). Neural network approach to weld quality monitoring. Proc. 20th Int. Conf. on industrial electronics, control and instrumentation, IEEE 2: 1287-1291.

34. Ohshima K, Yabe M, Akita K, Kugai K, Kubota T, Yamane S (1995). Sensor fusion using neural network in robotic welding. Industry Applications Conf., IEEE 2: 1764 - 1769.

35. Engelbrecht AP (2005). Fundamentals of Computational Swarm Intelligence. Wiley, NY.

36. Su-bing Z, Ze-min L (2001). Neural Network Training Using Ant Algorithm in ATM Traffic Control. Proceedings of the IEEE International Symposium on Circuits and Systems, Australia 3: 157-160.

37. Blum C, Socha K (2005). Training feed-forward neural networks with ant colony optimization: An application to pattern classification. Fifth IEEE International Conference on Hybrid Intelligent Systems: 233-238.

38. Liu YP, Wu MG, Qian JX (2007). Predicting coal ash fusion temperature based on its chemical composition using ACO-BP neural network. Thermochimica Acta 454: 64-68.
39. Haykin S (2003). Neural Networks, a comprehensive foundation. Second edition, Pearson Education Pvt. Ltd.
40. Dorigo M, Maniezzo V, Colorni A (1996). Ant system: optimization by a colony of co-operating agents. IEEE Transaction on Systems, Man and Cybernatics- Part B 26(1): 29-41.
41. Johnson S (2001). Emergence: The Connected Lives of Ants, Brains, Cities, and Software, Scribner, New York.
42. Applegate DL, Bixby RE, Chvátal V, Cook WJ (2006). The Traveling Salesman Problem: A Computational Study, Princeton University Press, Princeton.
43. Montgomery DC (1976). Design and analysis of experiments. John Wiley, New York.
44. Minitab Inc., User manual of MINITAB™ statistical software, Release 13.31, State College, PA 16801 USA, 2000.

List of Tables and Table Captions

Table 1. Chemical composition of the workpiece metal (in weight percentage)

C	Mn	Si	P	Cu	S	Ni	Cr	Fe
0.139	0.499	0.151	0.075	0.056	0.044	0.024	0.019	98.993

Table 2. Process parameters and their values

Sl. no.	Process parameter	Level 1	Level 2	Level 3
1	Background Voltage (V_b), volt	14	17	20
2	Pulse voltage (V_p), volt	30	34.6	39
3	Pulse frequency (f), hz	80	130	182
4	Pulse duty factor (η)	0.35	0.50	0.65
5	Wire feed rate (v_w), m/min	7	9	11
6	Table feed rate (v_t), mm/sec	2.456	3.760	5.635

Table 3. Design matrix of the experiment (see nomenclature in Table 2)

Sl. no.	V_b (volt)	V_p (volt)	f (hz)	η	v_w (m/min)	v_t (mm/s)	RMS current (A)	RMS voltage (V)	UTS (MPa)
1	17	34.6	130	0.5	9	3.76	1.1939	2.7429	412.28
2	17	34.6	130	0.5	9	3.76	1.1415	2.7449	415.79
3	14	30	80	0.35	11	5.635	1.4385	1.6834	0
4	14	39	80	0.35	7	5.635	1.1971	2.719	328.71
5	14	30	182	0.65	11	5.635	1.2566	2.3814	385.98
6	20	39	80	0.65	7	5.635	1.2773	3.2596	246.92
7	14	39	80	0.65	7	2.456	1.2791	3.1528	353.4
8	17	34.6	130	0.5	7	3.76	1.0516	2.7334	329.75
9	20	30	80	0.35	11	2.456	1.4692	1.9772	0
10	17	34.6	130	0.5	9	5.635	1.1839	2.6688	214.38
11	17	34.6	182	0.5	9	3.76	1.1434	2.6927	452.31
12	17	30	130	0.5	9	3.76	1.1998	2.3022	190.69
13	14	30	80	0.65	7	5.635	0.9921	2.4823	193.88
14	20	39	80	0.35	7	2.456	1.1052	2.9427	463.03
15	20	30	182	0.65	11	2.456	1.4019	2.3886	231.11
16	17	34.6	130	0.5	9	3.76	1.1493	2.75	412.53
17	17	34.6	130	0.5	9	3.76	1.1945	2.7313	419.28
18	14	30	182	0.35	11	2.456	1.5672	1.7755	0
19	14	39	80	0.65	11	5.635	1.5484	2.9822	461.73
20	14	30	80	0.65	11	2.456	0.7498	2.6086	331.28
21	17	34.6	130	0.5	9	3.76	1.165	2.7508	411.85
22	17	34.6	130	0.65	9	3.76	1.2652	2.8668	419
23	20	30	182	0.35	7	2.456	1.0122	2.4273	371.65
24	17	34.6	130	0.5	9	3.76	1.1989	2.7081	417.33
25	20	39	182	0.35	11	2.456	1.3841	2.6365	375.44
26	17	34.6	80	0.5	9	3.76	1.1516	2.7705	403.06
27	20	30	182	0.35	11	5.635	1.3825	1.9676	0
28	14	34.6	130	0.5	9	3.76	1.1673	2.6496	424.97
29	17	34.6	130	0.5	9	2.456	1.1965	2.7268	463.8
30	14	39	182	0.35	11	5.635	1.3096	2.4468	282.97
31	20	39	182	0.65	7	2.456	1.246	3.2878	263.6
32	20	30	182	0.65	7	5.635	1.0026	2.5803	370.21
33	20	30	80	0.65	11	5.635	1.2856	2.4107	251.88
34	17	34.6	130	0.35	9	3.76	1.2128	2.3691	293.29
35	14	39	182	0.65	11	2.456	1.3787	3.1774	418.95
36	14	39	182	0.65	7	5.635	1.2979	3.1505	232.75
37	20	39	182	0.65	11	5.635	1.2026	2.7367	455.13
38	17	34.6	130	0.5	9	3.76	1.1717	2.7386	420.97
39	14	30	182	0.35	7	5.635	1.0123	2.0132	11.523
40	20	30	80	0.35	7	5.635	0.9973	2.2885	189.19
41	17	39	130	0.5	9	3.76	1.3221	2.9732	443.87
42	20	30	80	0.65	7	2.456	0.9922	2.6553	436.47
43	14	30	80	0.35	7	2.456	1.0072	2.2308	15.2
44	14	39	80	0.35	11	2.456	1.4916	2.2187	109.34
45	14	30	182	0.65	7	2.456	0.979	2.6119	356.67
46	17	34.6	130	0.5	11	3.76	1.3023	2.5549	402.58
47	20	39	182	0.35	7	5.635	1.1497	2.5602	265.93
48	17	34.6	130	0.5	9	3.76	1.2161	2.7163	410.64
49	20	39	80	0.65	11	2.456	1.3634	3.2698	453.11
50	20	39	80	0.35	11	5.635	1.3265	2.7507	367.01
51	14	39	182	0.35	7	2.456	1.107	2.6931	445.03
52	17	34.6	130	0.5	9	3.76	1.1947	2.6984	413.43
53	20	34.6	130	0.5	9	3.76	1.1786	2.6165	349.2

Table 4. Effect of pheromone secretion rate on ACO-BPNN model

Pheromone secretion rate	Epochs	Maximum error (%)	Minimum error (%)	Mean error (%)
0.05	10752	6.73	0.40	3.69
0.075	11006	9.30	0.774	4.26
0.1	9268	6.19	0.094	**2.90**
0.125	9866	6.17	2.35	4.61
0.15	11308	6.58	0.79	3.68
0.2	**8028**	6.95	1.53	3.78
0.3	11639	7.47	2.23	4.96
0.4	9007	5.79	2.48	4.05

Table 5. Effect of pheromone evaporation rate on ACO-BPNN model

Pheromone evaporation rate	Epochs	Maximum error (%)	Minimum error (%)	Mean error (%)
0.005	10184	6.61	0.86	3.67
0.0075	9714	7.28	0.59	4.16
0.01	**9268**	6.19	0.09	**2.90**
0.0125	13392	41.42	0.003	10.08
0.015	9859	7.82	0.84	4.70
0.02	10795	6.33	0.60	4.38
0.03	9328	9.57	1.14	4.99
0.04	9613	6.52	0.51	4.07
0.05	9338	8.09	2.03	4.94

Table 6. Parameters of BPNN and ACO-BPNN models

Model parameter	Value
Learning rate	0.9
Momentum parameter	0.9
Pheromone secretion rate	0.1
Pheromone evaporation rate	0.01

Table 7. Comparison between ACO-BPNN and BPNN models, for different architectures

	ACO-BPNN					BPNN			
Architecture	Number of ants	Epochs	Maximum error (%)	Minimum error (%)	Mean error (%)	Epochs	Maximum error (%)	Minimum error (%)	Mean error (%)
8-3-1	1	17686	5.19	0.54	3.68	128230	12.76	2.86	6.52
8-4-1	6	13888	5.42	0.56	3.69	15602	5.21	0.84	**3.71**
8-5-1	8	8566	6.93	0.21	3.45	12904	5.24	2.35	3.86
8-6-1	3	11209	5.32	0.65	3.71	11754	9.37	1.40	4.68
8-7-1	1	9321	6.32	0.47	3.45	11971	6.37	2.19	4.4
8-8-1	1	9273	6.66	0.47	3.35	12518	5.52	1.03	3.80
8-9-1	5	10731	5.90	1.53	3.13	11635	11.67	1.23	5.28
8-10-1	2	**8349**	5.94	1.04	3.59	14983	13.59	1.01	5.57
8-11-1	6	9671	6.33	0.21	3.15	**10911**	10.63	1.50	5.38
8-12-1	2	8512	7.44	0.65	3.43	12334	13.06	1.55	5.98
8-13-1	7	9268	6.19	0.09	**2.90**	12672	7.77	1.23	3.69
8-14-1	6	10065	7.69	0.22	2.96	11202	13.46	1.31	5.85

List of captions for the illustrations