Optimising Neural Networks for Identification of Wood Defects Using the Bees Algorithm

D.T Pham
Manufacturing Engineering Centre (MEC)
Cardiff University
The Parade, Cardiff
CF24 3AA UK

Anthony J. Soroka
MEC
Cardiff University
The Parade, Cardiff
CF24 3AA UK
SorokaAJ@cf.ac.uk

Afshin Ghanbarzadeh
MEC
Cardiff University
The Parade, Cardiff
CF24 3AA UK

Ebubekir Koc
MEC
Cardiff University
The Parade, Cardiff
CF24 3AA UK

Sameh Otri
MEC
Cardiff University
The Parade, Cardiff
CF24 3AA UK

Michael Packianather
MEC
Cardiff University
The Parade, Cardiff
CF24 3AA UK

Abstract – **This paper presents an application of the *Bees Algorithm (BA)* to the optimisation of neural networks for wood defect detection. This novel population-based search algorithm mimics the natural foraging behaviour of swarms of bees. In its basic version, the algorithm performs a kind of neighbourhood search combined with random search. Following a brief description of the algorithm, the paper gives the results obtained for the wood defect identification problem demonstrating the efficiency and robustness of the new algorithm.**

I. INTRODUCTION

Many complex multi-variable optimisation problems cannot be solved exactly within polynomially bounded computation times. This has generated interest in search algorithms that find near-optimal solutions in reasonable running times. The algorithm described in this paper is a search algorithm capable of locating good solutions efficiently. The algorithm is inspired by the food foraging behaviour of honey bees and could be regarded as belonging to the category of "intelligent" optimisation tools [1].

This paper presents an application of the new algorithm to the problem of identifying defects in plywood veneer. An example of a sheet of wood veneer is shown in Fig. 1. It can be seen that the sheet contains several defects. These could create quality problems when the sheets are bonded together. Researchers have developed systems for automatically detecting and identifying defects in plywood veneer. Such systems generally involve the use of image processing techniques, feature extraction to capture the essential characteristics of all defects and a classifier to recognise these defects.

The following sections provide an introduction to the algorithm and an explanation of the wood defect problem and the neural network used to identify the defects.

II. INTELLIGENT SWARM BASED OPTIMISATION

Swarm–based optimisation algorithms (SOAs) mimic nature's methods to drive a search towards the optimal solution. A key difference between SOAs and direct search algorithms such as hill climbing and random walk is that SOAs use a population of solutions for every iteration instead of a single solution. As a population of solutions is processed in an iteration, the outcome of each iteration is also a population of solutions. If an optimisation problem has a single optimum, SOA population members can be expected to converge to that optimum solution. However, if an optimisation problem has multiple optimal solutions, an SOA can be used to capture them in its final population. SOAs include the Ant Colony Optimisation (ACO) algorithm [2], the Genetic Algorithm (GA) [3] and the Particle Swarm Optimisation (PSO) algorithm [4].

Fig. 1. An example of a wood veneer sheet containing several defects.

Common to all population-based search methods is a strategy that generates variations of the solution being sought. Some search methods use a greedy criterion to decide which generated solution to retain. Such a criterion

0-7803-9701-0/06/$20.00 ©2006 IEEE

would mean accepting a new solution if and only if it increases the value of the objective function (assuming the given optimisation problem is one of optimisation). A very successful non-greedy population-based algorithm is the ACO algorithm which emulates the behaviour of real ants. Ants are capable of finding the shortest path from the food source to their nest using a chemical substance called pheromone to guide their search. The pheromone is deposited on the ground as the ants move and the probability that a passing stray ant will follow this trail depends on the quantity of pheromone laid. ACO was first used for functional optimisation by Bilchev [5] and further attempts were reported in [5, 6].

The Genetic Algorithm is based on natural selection and genetic recombination. The algorithm works by choosing solutions from the current population and then applying genetic operators – such as mutation and crossover – to create a new population. The algorithm efficiently exploits historical information to speculate on new search areas with improved performance [3]. When applied to optimisation problems, the GA has the advantage of performing global search. The GA may be hybridised with domain-dependent heuristics for improved results. For example, Mathur et al [6] describe a hybrid of the ACO algorithm and the GA for continuous function optimisation.

Particle Swarm Optimisation (PSO) is an optimisation procedure based on the social behaviour of groups or organisations, for example the flocking of birds or the schooling of fish [4]. Individual solutions in a population are viewed as "particles" that evolve or change their positions with time. Each particle modifies its position in search space according to its own experience and also that of a neighbouring particle by remembering the best position visited by itself and its neighbours, thus combining local and global search methods [4].

There are other SOAs with names suggestive of possibly bee-inspired operations [7-10]. However, as far as the authors are aware, those algorithms do not closely follow the behaviour of bees. In particular, they do not seem to implement the techniques that bees employ when foraging for food.

III. THE BEES ALGORITHM

A. Bees in the Natural Environment

A colony of honey bees can be seen as a diffuse creature which can extend itself over long distances in multiple directions in order to exploit a large number of food sources at the same time [7, 8]. In principle, flower patches with plentiful amounts of nectar or pollen that can be collected with less effort should be visited by more bees, whereas patches with less nectar or pollen should receive fewer bees [9, 10].

The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees search randomly from one patch to another. During the harvesting season, a colony continues its exploration, keeping a percentage of the population as scout bees [8].

When they return to the hive, those scout bees that found a patch which is rated above a certain threshold (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the "waggle dance" [7].

This dance is essential for colony communication, and contains three vital pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive and its quality rating [7, 10]. This information helps the bees to find the flower patches precisely, without using guides or maps. Each individual's knowledge of the outside environment is gleaned solely from the waggle dance. This dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it [10]. After waggle dancing on the dance floor, the dancer bee (i.e. the scout bee) goes back to the flower patch with follower bees that were waiting inside the hive. The number of follower bees assigned to a patch depends on the overall quality of the patch. This allows the colony to gather food quickly and efficiently.

While harvesting from a patch, the bees monitor its food level. This is necessary to decide upon the next waggle dance when they return to the hive [10]. If the patch is still good enough as a food source, then it will be advertised in the waggle dance and more bees will be recruited to that source.

B. The Bees Algorithm

As mentioned above, the Bees Algorithm is an optimisation algorithm inspired by the natural foraging behaviour of honey bees [1]. Figure 2 shows the pseudo code for the algorithm in its simplest form. The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of sites selected out of n visited sites (m), number of elite sites out of m selected sites (e), number of bees recruited for the best e sites (nep), number of bees recruited for the other (m-e) selected sites (nsp), initial size of patches (ngh) which includes site and its neighbourhood and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitnesses of the sites visited by the scout bees are evaluated in step 2. The valuation of fitness would depend on the optimisation problem, but in general 'fitness' is taken as the value of the objective function being optimised.

In step 4, bees that have the highest fitnesses are designated as "selected bees" and sites visited by them are chosen for neighbourhood search. Then, in steps 5 and 6, the algorithm conducts searches in the neighbourhood of the selected sites, assigning more bees to search near to the best e sites. The bees can be chosen directly according to the fitnesses associated with the sites they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Searches in the neighbourhood of the best e sites which represent more promising solutions are made more detailed by recruiting more bees to follow selected bees than the other selected

bees. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm.

However, in step 6, for each patch only the bee with the highest fitness will be selected to form the next bee population. In nature, there is no such a restriction. This constraint is introduced here to reduce the number of points to be explored. In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts to its new population – representatives from each selected patch and other scout bees assigned to conduct random searches.

The Bees Algorithm was adopted in this work as it had proved to have a more robust performance than other intelligent optimisation methods for a range of complex problems [1].

IV. WOOD VENEER DEFECTS

In this study, using a charge-coupled device (CCD) matrix camera, the wood veneer defects were captured and stored on a digital computer. The wood veneer data acquisition rig shown in Fig. 3. These images were converted into grey level histograms after applying segmentation and image processing algorithms. From the first and second order statistical features extracted from the histogram, 17 features were selected for training the neural network. These are shown in Table 1. All together 12 wood veneer defects and clear wood shown in Fig. 4 were included in the examples used for training and testing the neural networks. Automated Visual Inspection (AVI) systems for identifying defects using neural networks have been proposed by Pham and Alcock [11] and Packianather and Drake [12]. The generic process for the visual inspection of wood defects is given in Fig 5.

1. Initialise population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met)
//Forming new population.
4. Select sites for neighbourhood search.
5. Recruit bees for selected sites (more bees for the best e sites) and evaluate fitnesses.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitnesses.
8. End While.

Fig. 2. Pseudo code of the basic Bees Algorithm

Fig. 3. Inspection rig. The inspection rig for wood defect detection.

The wood panels are automatically moved to the image capture area by a conveyor belt. The system uses a Hamamatsu monochrome CCD matrix camera (resolution 739 x 575 pixels) to take images of the wood veneer.

Uniform illumination is provided by a back light (58W fluorescent lamp) and front lighting system (halogen lamps: edges 500W and middle 300W). Basic image processing functions (e.g. thresholding and filtering) are implemented in hardware. Image segmentation algorithms are used to detect the boundaries of the sheet and open defects and defect detection algorithms are used to find potential defect areas.

Fig. 4. Wood veneer defect types. There are 12 distinct types of defect that need to be identified by the neural network plus clear wood.

TABLE 1
FEATURES SELECTED FOR TRAINING OF NEURAL NETWORK

Feature No.	Feature	Description
1	Mean grey level (μ)	
2	Median grey level	*below which 50% of the values fall*
3	Mode grey level	*the most frequent value*
4	Standard deviation of the grey levels (s)	*the spread*
5	Skewness	*direction, extent of departure from symmetry*
6	Kurtosis	*measures the "peakedness" of the histogram*
7	Number of pixels with a grey level ≤ 80	*number of dark pixels*
8	Number of pixels with a grey level ≥220	*number of bright pixels*
9	Grey level (p) for which there are 20 pixels below	*lowest grey level* - The grey level p is used as the lowest grey level to accommodate for potential noise pixels
10	Grey level (s) for which there are 20 pixels above	*highest grey level* – The grey level s is used as the highest grey level to accommodate for potential noise pixels
11	Histogram tail length on the dark side (q-p)	q is the grey level below which there are 2000 pixels
12	Histogram tail length on the bright side (s-r)	r is the grey level above which there are 2000 pixels
13	Number of edge pixels after thresholding a segmented window at mean value	*defined to detect dark and bright defects*
14	Number of pixels after thresholding at μ -2σ	
15	The number of edge pixels for feature 14	*f14 and f15 defined to detect dark defects*
16	Number of pixels after thresholding at μ +2 σ	
17	Calculate the number of edge pixels for feature 16	*f16 and f17 defined to detect bright defects*

Fig. 5. Generic Automated Visual Inspection system for wood defect identification.

For the particular application studied here, 232 examples (both defects and clear wood) were employed. This represents the complete set of examples available to the authors. Each example is a vector containing 17 features. Table 2 shows thirteen different classes of vectors and the number of examples in class. The initial classification of these examples had been performed by a human inspector. For subsequent neural network classification experiments, for each class, 80% (185 in total) of the examples were selected at random to form the training set and the remaining 20% (47 in total) formed the test set.

TABLE 2
PATTERN CLASSES AND THE NUMBER OF EXAMPLES USED FOR TRAINING AND TESTING

Pattern Class	Total	Used for training	Used for Testing
Bark	20	16	4
Clear wood	20	16	4
Colored streaks	20	16	4
Curly grain	16	13	3
Discoloration	20	16	4
Holes	8	6	2
Pin knots	20	16	4
Rotten knots	20	16	4
Roughness	20	16	4
Sound knots	20	16	4
Splits	20	16	4
Streaks	20	16	4
Wormholes	20	16	4
Total	232	185	47

V. NEURAL NETWORKS AND THEIR OPTIMISATION

A. Neural Network

Neural networks (also known as Artificial Neural Networks) are computational models inspired by the human brain. The multilayered feedforward neural network has an input layer to receive inputs from sensors or other sources, an output layer to communicate with the outside world and one or more hidden layers for data processing to transform the inputs into outputs.

The architecture of a feedforward neural network with one hidden layer is shown in Figure 6. Each layer is made up of processing elements called neurons. Every neuron has a number of inputs, each of which must store a connection weight to indicate the strength of the connection. Connections are initially made with random weights.

The neuron sums the weighted inputs and computes a single output using an activation function. A number of different activation functions can be used. In this study, a hyperbolic tangent function is used in order to increase the difference between the outputs and further details are provided in [12]. Each neuron in a layer is fully connected to every neuron in the subsequent layer forming a fully connected feedforward neural network. In a feedforward neural network, information flows from the input layer to the output layer without any feedback.

There is one bias neuron for each hidden layer and the output layer, as illustrated in Figure 6, and they are connected to each neuron in their respective layer. These connections are treated as weights. During the training process these weights are adjusted to achieve optimal accuracy and coverage.

B. Optimisation

The standard method for refining such a neural network is using an error backpropagation algorithm. During the training phase, the feedforward calculation is combined with backward error propagation to adjust the weights. The error term for a given pattern is the difference between the desired output and the actual output (the output from feedforward calculation).

In this work the Bees Algorithm is used to optimise the weights the neural network in place of the backpropagation

Fig. 6. Feedforward neural network with one hidden layer.

TABLE 3
PARAMETERS USED FOR OPTIMISATION

Parameter	Value
n	100
m	20
nsp	10
nep	50
e	1
ngh	0.1

algorithm described above and utilised in the work described in [12]. In order to maintain comparability the neural network structure (number of hidden layers) remained the same in both neural networks, with 17 input neurons, 51 hidden neurons and 13 output neurons. The parameters used by the Bees Algorithm are given in Table 3.

The optimisation using the Bees Algorithm will involve the "Bees" searching for the optimal values of the weights assigned to the connections between the neurons within the network where each bee represents a neural network with a particular set of weights. The aim of the Bees Algorithm is to find the bee producing the smallest value of the error function.

VI. RESULTS

Table 4 shows the mean accuracies achieved in thirty experiments using the conventional backpropagation method [12], those from the Bees Algorithm and the results using a non-neural network approach, in this case, a Minimum Distance Classifier (MDC).

The results show that the Bees Algorithm is able to achieve an accuracy that is comparable to the backpropagation method. As the two mean accuracies are identical, this suggests that for this particular neural network structure and for the data set being used for training accuracy in the region of 86% is the highest that can be achieved.

The results also show that the neural network based classifier has better generalisation capability compared to the Minimum Distance Classifier [12] which can only achieve 63.12% accuracy.

A notable difference between the Bees Algorithm and backpropagation is the number of iterations required to optimise the neural network. The former required a total of 1300 iterations whereas the latter required 65305 iterations.

TABLE 4
RESULTS FOR DEFECT IDENTIFICATION

Method	Mean Accuracy (%)
NN - Back propagation	86.52
NN - Bees Algorithm	86.52
MDC (Non-NN)	63.12

VII. CONCLUSIONS

This paper has presented a novel application of the Bees Algorithm to the optimisation of neural networks for the identification of defects in wood veneer sheets.

The accuracy obtained is comparable to that achieved using backpropagation. However, the Bees Algorithm proved to be considerably faster. This work therefore confirms the usefulness of the algorithm as an optimisation tool, particularly when considering that it has produced even higher accuracies than backpropagation in other applications [13].

The studies also re-confirm the suitability of the neural network approach in the identification of defects in wood veneer sheets.

VIII. FURTHER WORK

It is intended to apply the Bees Algorithm to other types of Artificial Neural Network for this wood defect identification problem and to compare the results obtained with those produced using standard training methods.

IX. ACKNOWLEDGEMENTS

This work was funded under the European Regional Development Fund (ERDF) Objective 1 Project SUPERMAN and by the I*PROMS European Union, Framework Programme 6 Network of Excellence.

X. REFERENCES

[1] D.T. Pham , A. Ghanbarzadeh , E. Koc , S. Otri , S. Rahim , M. Zaidi. "The Bees Algorithm – A Novel Tool for Complex Optimisation Problems" Proceedings of *2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2006),* The Internet, July 2006.

[2] Dorigo, M. and T. Stützle, *Ant Colony Optimization.* 2004, Cambridge: MIT Press.

[3] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning.* 1989, Reading: Addison-Wesley Longman.

[4] Eberhart, R., Y. Shi, and J. Kennedy, *Swarm Intelligence.* 2001, San Francisco: Morgan Kaufmann.

[5] Bilchev, G. and I.C. Parmee, *The Ant Colony Metaphor for Searching Continuous Design Spaces*, in Selected Papers from AISB Workshop on Evolutionary Computing. 1995, Springer-Verlag: London. p. 25-39.

[6] Mathur, M., S.B. Karale, S. Priye, V.K. Jayaraman, and B.D. Kulkarni, *Ant Colony Approach to Continuous Function Optimization.* Ind. Eng. Chem. Res., 2000. 39(10): p. 3814-3822.

[7] Frisch, K.V., Bees: *Their Vision, Chemical Senses and Language.* Revised Edition ed. 1976, Ithaca, N.Y.: Cornell University Press.

[8] Seeley, T.D., *The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies.* 1996, Cambridge, Massachusetts: Harvard University Press.

[9] Bonabeau, E., M. Dorigo, and G. Theraulaz, *Swarm Intelligence: from Natural to Artificial Systems.* 1999, New York: Oxford University Press.

[10] Camazine, S., J.-L. Deneubourg, N.R. Franks, J. Sneyd, G. Theraula, and E. Bonabeau, *Self-Organization in Biological Systems.* 2003, Princeton: Princeton University Press.

[11] Pham DT and Alcock RJ. *Automatic Detection of Defects on Birch Wood Boards.* Proc. I Mech E, Part E, J. of Process Mechanical Engineering. 210 (1996), p 45-52.

[12] Packianather, M.S. and Drake, P.R. (2005) *Identifying defects on plywood using a minimum distance classifier and a neural network.* In: Pham D.T., Eldukhri E., Soroka A ed(s) *1st Virtual International Conference on Intelligent Production Machines and Systems (IPROMS 2005).* Elsevier, Oxford, p 543-548.

[13] Pham D.T., Koç E, Ghanbarzadeh A., Otri S. "Optimisation of the Weights of Multi-Layered Perceptrons Using the Bees Algorithm" IMS 2006: 5th International Symposium On Intelligent Manufacturing Systems, Sakarya, Turkey.