

# Optimizing Models Using Continuous Ant Algorithms

Oleg Kovářík, Pavel Kordík

*Department of Computer Science and Engineering, FEE, Czech Technical University,  
Karlovo nám. 13, 121 35 Praha 2, Czech Republic*

kovarol@fel.cvut.cz, kordikp@fel.cvut.cz

**Abstract.** *While constructing inductive models of a given system, we need to optimize parameters of units the system is composed of. These parameters are often real-valued variables and we can use a large scale of continuous optimization methods to locate their optimum. Each of these methods can give different results for problems of various nature or complexity. In our experiments, the usually best performing gradient based Quasi-Newton method was unable to optimize parameters for a well known problem of two intertwined spirals; its classification accuracy was close to 50%. Therefore, we compared several continuous optimization algorithms performance on this particular problem. Our results show that two probabilistic algorithms inspired by ant behaviour are able to optimize parameters of model units for this problem with the classification accuracy of 70%.*

## Keywords

Ant Algorithms, Continuous Optimization, Inductive Modelling.

## 1 Introduction

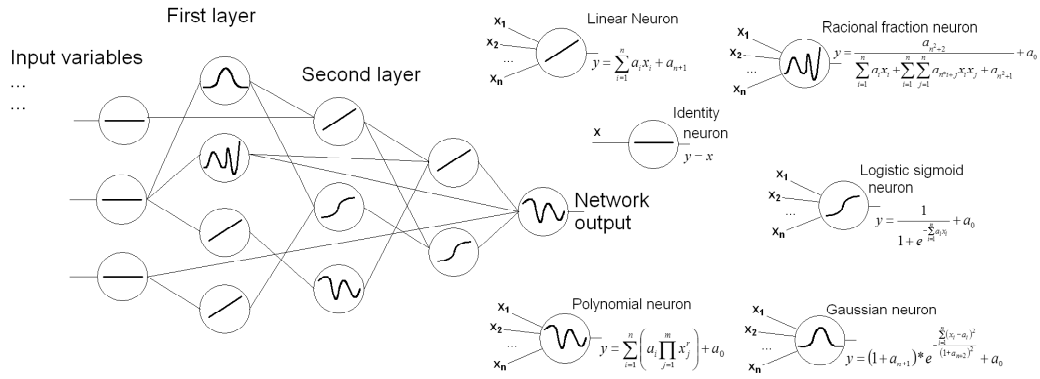
In many applications it is important to find optimal model of an unknown system. These include classification, approximation, prediction etc. Such models can be constructed in two essentially different ways – deductive or inductive. The GAME artificial neural network (ANN) [1], which we use in this paper, is based on inductive approach. The structure of the ANN is constructed during a learning process from functional units. These units have parameters which we want to optimize so that the network is able to produce a model as exact as possible. Because we have a problem with modelling two intertwined spirals using Quasi-Newton [2] optimization method, we examine two algorithms originally inspired by pheromone laying behaviour of real-world ants. Before evaluation of these methods, we shortly describe GAME Neural Network and examined algorithms.

## 2 Theoretical Part

### 2.1 GAME Neural Network

The GAME ANN extends concept of GMDH network [3]. The GMDH uses only one type of a structural unit (neuron with one transfer function). On the other hand, in GAME ANN there are neurons with many different transfer functions (linear, sigmoid, polynomial, etc.). The GAME has a feed-forward structure; for example of its architecture see fig. 1.

The GAME ANN is build from scratch during the training phase. Each new layer is constructed in the following way: first, large number of new neurons is generated. Neurons differ in transfer function and in the number of connections to the previous layer. The next step is to find optimal setup of internal parameters and the best of the connections to neurons in previous layer. Advanced genetic algorithm is used to find the best connections, transfer function, its parameters and optimization method. After that, all neurons are evaluated on a testing set and the worst neurons are deleted from the



**Fig. 1.** Example of the GAME model structure.

layer. Then the layer is "frozen" and the algorithm creates next layer. This is repeated until a neuron with satisfactory output accuracy is found.

## 2.2 Continuous Ant Algorithms

Ant algorithms are inspired by pheromone laying and evaporation in ant colonies. This process of indirect communication by environment modification (called stigmergy) enables ants to share information between individuals and to solve problems in parallel. Originally, ant methods were used in discrete optimization to solve Travelling Salesman Problem. Later on, several modifications for continuous and mixed-variable domains were developed. We examine Extended Ant Colony Optimization and Direct ACO in this work.

### 2.2.1 Direct ACO (DACO)

Direct ACO [4] uses two kinds of pheromones - one for mean values and one for standard deviation of optimized multidimensional variable. These values are used by ants to create new solutions and are updated with regard to quality of these solutions. Each variable  $x_i$  is associated with its distribution parameters  $N(\mu_i, \sigma_i^2)$ . These values directly represent the amount of pheromone. One iteration consists of pheromone evaporation and intensification. Evaporation can be implemented by multiplication by a constant:

$$\begin{aligned} \vec{\mu}(t) &= (1 - \rho)\vec{\mu}(t - 1) \\ \vec{\sigma}(t) &= (1 - \rho)\vec{\sigma}(t - 1) \end{aligned} \quad (1)$$

where  $\rho \in \langle 0, 1 \rangle$  is the evaporation factor. More interesting is the intensification phase:

$$\begin{aligned} \vec{\mu}(t) &= \vec{\mu}(t) + \vec{\rho} \\ \vec{\sigma}(t) &= \vec{\sigma}(t) + \vec{\rho}|\vec{x} - \vec{\mu}(t - 1)| \end{aligned} \quad (2)$$

where  $\vec{x}$  is the solution selected for intensification. In this case, the globally best solution is used for this process.

### 2.2.2 Extended Ant Colony Optimization (ACO\*)

Extended ACO [5] follows the original Ant Colony Optimization metaheuristic [6]. Its structure allows user to solve mixed discrete-continuous optimization problems. An ant travels across all variable dimensions and for each dimension it chooses a value with a probability given by distribution  $P(x)$ . In continuous domains, the discrete distribution changes to the continuous one - the Probability Density Function (PDF):

$$g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3)$$

This function has the advantage of a simple random number generation but has only one maximum and thus can describe only one promising area at a time. That is why a mixture of normal kernels is used:

$$P(x) = G(x, \vec{\omega}, \vec{\mu}, \vec{\sigma}) = \sum_{j=1}^k \omega_j \cdot g(x, \mu_j, \sigma_j) \quad (4)$$

where  $\vec{\omega}$  is a vector of weights,  $\vec{\mu}$  are mean values and  $\vec{\sigma}$  stands for standard deviations vector. Weights are calculated from the solution fitness with a rank-based method transformed by Gaussian function with the argument of rank  $r$ , mean 1.0 and standard deviation  $qk$ :

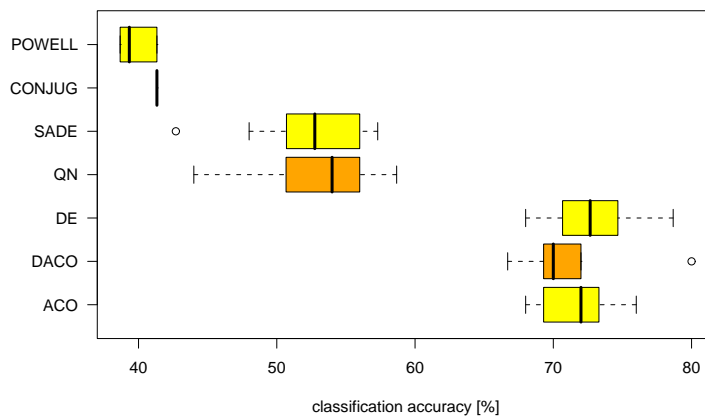
$$\omega_j = \frac{1}{qk\sqrt{2\pi}} \cdot e^{-\frac{(r-1)}{2q^2k^2}} \quad (5)$$

The pheromone maintenance can be done by adding new promising PDFs to the mixture, by removing PDFs with low quality, evaporation of  $\vec{\omega}$  or by increasing deviations  $\vec{\sigma}$  (which has the effect of spreading the Gaussian curve and thus dispersing sampled values).

### 3 Experiments

The well known classification problem of two intertwined spirals was used to compare selected ant algorithms with other continuous optimization algorithms. Results from the previous experiments showed that this dataset is difficult for Quasi-Newton method, which usually gives the best results for other datasets. Spirals dataset was problematic for other methods as well. Typical classification accuracy was around 50%, which corresponds with random behaviour. That is why the settings of GAME were adjusted to use 99% of training set for learning and validation on training set was turned on. This setting leads to an overfitting and it usually results in bad performance on testing dataset. Also, different GAME network unit types were used from which sinus neuron led to a better ant algorithms accuracy.

We compared ant algorithms with several methods: Quasi-Newton (QN), Differential Evolution (DE), Conjugate Gradient (CONJUG), SADE Genetic Algorithm (SADE) and Powell's method (POWELL). For each examined method we constructed ensemble of 20 models on a training set. Accuracy of this ensemble on the testing dataset was observed. This was repeated 10 times to determine a variability in the quality of solutions.

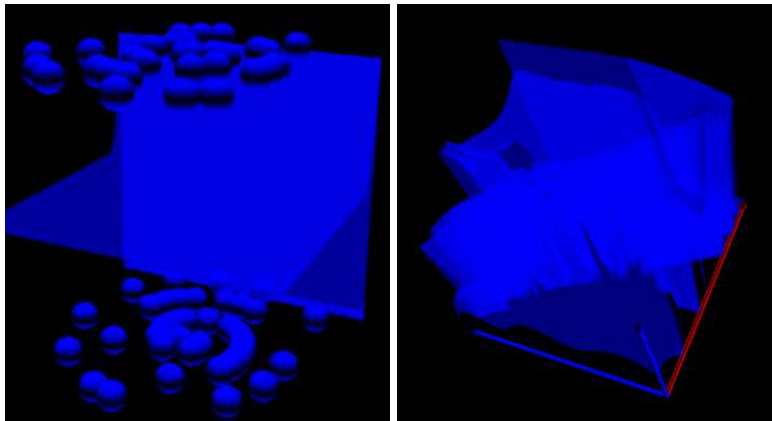


**Fig. 2.** Classification accuracy on a testing set of two intertwined spirals problem. Box-plots are generated from results of 10 model ensembles for each optimization method, each ensemble consists of 20 models.

## 4 Results

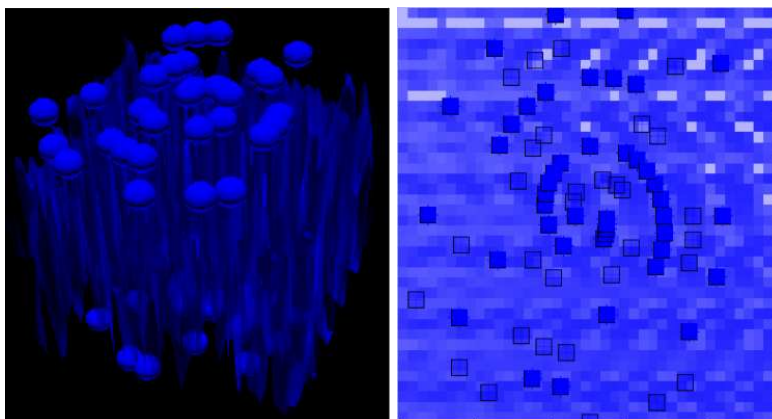
Comparison of selected methods is summarized in fig. 2. Closer to the left side, we can see the methods with results around 50% of classification accuracy on testing dataset, which indicates random guess. Some of these methods created extremely overfitted models (which is a result of settings mentioned in previous section), others were not able to make use of available units during the GAME network construction process. Closer to the right side are results of the ant methods together with a Differential Evolution with accuracy around 70%.

The reason why we used the validation on training data can be seen in fig. 3. Depicted model responses are typical responses of GAME networks for Quasi-Newton unit parameter optimization. The first one was acquired with standard settings (i.e. no validation on training data) and shows flat response. The second one is a chaotic response of an overfitted model with validation on training data. This shows that the Quasi-Newton method was not able to adapt to the problem.



**Fig. 3.** Example of poor responses of models optimized by Quasi-Newton method on spiral dataset. The left one was trained using standard settings; it shows almost no knowledge about the two classes represented by spheres. Picture on the right was generated with a validation on training data (99% of training dataset). It is highly overfitted and gives a random response. Both models have around 50% classification accuracy on testing dataset.

On the other hand, the examined ant algorithms and Differential Evolution method were able to generate fractal-like solutions with the assistance of sinus units (see fig. 4). These solutions reached a surprisingly high accuracies on a testing dataset.



**Fig. 4.** Example response of model optimized by ant inspired algorithm (DACO). The response seems chaotic and overfitted, but these models have around 70% classification accuracy thanks to this "fractal" shape. Picture on the right is 2D view of the example response.

## 5 Conclusion

We compared performance of several continuous optimization algorithms on the problem of adapting parameters of inductive model units to the two intertwined spirals dataset. This was an unsolvable task for gradient based Quasi-Newton method (with only about 50% classification accuracy) which often gives the best results on other datasets. We show that two methods inspired by ant behaviour (ACO\*, DACO) together with Differential Evolution algorithm outperform Quasi-Newton method by gaining 70% accuracy. We can make use of this behaviour by combining different optimization methods to solve larger scale of problems.

Ant algorithms were able to utilize sinus units of the GAME network and their best results were achieved using validation on training data (99% of a training dataset). This overfitting setting was advantageous only for ant algorithms and Differential Evolution. We expected GAME to approximate spirals by sets of overlapping gaussian units peaks. However, visual inspection revealed "fractal" response of models.

These results lead to a conclusion that the ant algorithms can be used for problems where the gradient based methods fail. For the future research we suggest to examine GAME in order to find the reason why our expectations about gaussian units utilization were not fulfilled. Further research should also focus on finding datasets with similar results in order to identify their general characteristics.

## Acknowledgements

This research is partially supported by the grant Automated Knowledge Extraction (KJB201210701) of the Grant Agency of the Academy of Science of the Czech Republic, the research program "Transdisciplinary Research in the Area of Biomedical Engineering II" (MSM6840770012) sponsored by the Ministry of Education, Youth and Sports of the Czech Republic and by the internal grant of Czech Technical University in Prague, IG: CTU0807213 - "New modifications to ant algorithms".

## References

- [1] Pavel Kordík. *Fully Automated Knowledge Extraction using Group of Adaptive Models Evolution*. PhD thesis, Czech Technical University in Prague, FEE, Dep. of Comp. Sci. and Computers, FEE, CTU Prague, Czech Republic, September 2006.
- [2] Salane and Tewarson. A unified derivation of symmetric quasi-newton update formulas. *Applied Math*, 25:29–36, 1980.
- [3] A.G. Ivakhnenko, E.A. Savchenko, and G.A. Ivakhnenko. Gmdh algorithm for optimal model choice by the external error criterion with the extension of definition by model bias and its applications to the committees and neural networks. *Pattern Recognition and Image Analysis*, 12(4):347353, 2002.
- [4] Min Kong and Peng Tian. A direct application of ant colony optimization to function optimization problem in continuous domain. In *Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006, Brussels, Belgium, September 4-7, 2006. Proceedings*, Lecture Notes in Computer Science, pages 324–331. Springer, 2006.
- [5] Krzysztof Socha. ACO for continuous and mixed-variable optimization. In Marco Dorigo, Mauro Birattari, Christian Blum, Luca Maria Gambardella, Francesco Mondada, and Thomas Stützle, editors, *Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, ANTS 2004, Brussels, Belgium, September 5 - 8, 2004, Proceedings*, volume 3172 of *Lecture Notes in Computer Science*, pages 25–36. Springer, 2004.
- [6] Marco Dorigo, Gianni Di Caro, and Luca Maria Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.