Proceedings of the 11th International Conference on Manufacturing Research (ICMR2013), Cranfield University, UK, 19th – 20th September 2013, pp 481-488

INVITED PAPER

THE BEES ALGORITHM: MODELLING NATURE TO SOLVE COMPLEX OPTIMISATION PROBLEMS

Duc Truong Pham
School of Mechanical Engineering University of Birmingham
Birmingham B15 2TT
UK

Marco Castellani
Institute of Marine Research
P.O. Box 1870
Nordnes, N-5817 Bergen
NORWAY

Hoai An Le-Thi
Theoretical and Applied Computer Science Laboratory
University of Lorraine
Ile du Saulcy, 57 045 Metz
FRANCE

ABSTRACT

The Bees Algorithm models the foraging behaviour of honey bees in order to solve optimisation problems. The algorithm performs a kind of exploitative neighbourhood search combined with random explorative search. This paper describes the Bees Algorithm and presents two application examples: the training of neural networks to predict the energy efficiency of buildings, and the solution of the protein folding problem. The Bees Algorithm proved its effectiveness and speed, and obtained very competitive modelling accuracies compared with other state-of-the-art methods.

Keywords: intelligent optimisation, swarm intelligence, bees algorithm, honey bees.

1 INTRODUCTION

Many engineering problems entail tuning a number of system variables to optimise a given quality measure such as the reliability or accuracy of a process, or the quality or cost of a product. Unfortunately, the relationship between the system variables and the desired quality parameter is often complex, highly nonlinear, and ill-behaved. Implicit discontinuities and constraints on the state and input variables are also common.

Biological systems are known to be able to perform complex optimisation tasks, such as the natural adaptation of species, and group foraging and movement in social animals. Nature's near optimal problem-solving strategies often rely on stochastic approaches based on the interaction and self-organisation of large and decentralised ensembles of individuals.

Evolutionary Algorithms (EAs) (Rechenberg 1965), (Fogel, Owens and Walsh 1966), (Holland 1975) were the first optimisation methods inspired by the collective search process of a population of biological agents. Based on the Darwinian principle of survival of the fittest, EAs evolve a population of candidate solutions towards meeting some given quality measure(s).

Swarm Intelligence (SI) (Bonabeau, Dorigo and Theraulaz 1999) (Kennedy 2006) includes many recent model-free metaheuristics inspired by the collective intelligent behaviour of social animals. The SI paradigm is characterised by the use of a population of simple agents, some form of communication between the individuals, a decentralised control structure, self-organisation, and a random component in the agents' behaviour that fosters the exploration of new solutions. SI has found wide application in optimisation (Yang 2010), robotics (Gross and Dorigo 2009), image processing (Jevtic and Andina 2010), and computer graphics (Reynolds 1987).

One of the strengths of the above nature-inspired optimisation methods, is that they make no assumption on the properties of the fitness landscape. As such, they are applicable to any problem amenable to being encoded via a fitness evaluation function, and allowing some sort of parametric representation of the solutions.

This paper presents the Bees Algorithm (Pham, Otri, et al. 2005, Pham and Castellani 2009), a nature-inspired intelligent optimisation method based on the foraging behaviour of honey bees. Two typical application cases are presented. In the first, the Bees Algorithm is used to optimise the weights of an artificial neural network (ANN) (Pham and Liu 1995) to predict the energy efficiency of a building. In the second, the Bees Algorithm is used to obtain the global energy minimum in a molecular cluster to find the configuration of a protein.

2 THE BEES ALGORITHM

In a bee colony, a small part of the population continually scouts the environment in search of new food sources (i.e. flower patches) (Tereshko and Loengarov 2005). When a scout finds a new flower patch, it rates the discovery according to its profitability (Seeley 1996). A bee that found a rich food source communicates the location of its discovery to idle nest mates through a ritual called the "waggle dance" (Seeley 1996). The length of the waggle dance depends on the scout rating of the food source, allowing more bees to be recruited to harvest the best rated sources. Some of the recruited foragers may also perform the waggle dance upon their return to the hive, mobilising further foragers to exploit the food source.

In the Bees Algorithm, a population of artificial bees is split into a small number of 'scouts' and a larger group of 'foragers'. The scouts randomly sample the solution space, and evaluate the fitness of the visited locations (solutions). The foragers perform local search in the vicinity of known good solutions, looking for further fitness improvement. The amount of foragers allocated to a neighbourhood (flower patch) depends on the fitness of the solution, according to a mechanism mimicking the waggle dance of biological bees.

The algorithm is composed of several optimisation cycles, where new solutions are generated and compared with the best-so-far findings, and the highest ranking ones are selected for local search. If local search fails to bring improvements of fitness around a solution for a given number of cycles, the flower patch is considered exhausted (the local fitness peak has been attained) and is abandoned (Pham and Castellani 2009). The whole procedure is run until a satisfactory solution is found, or a given number of optimisation cycles are completed.

Without loss of generality, continuous optimisation tasks will be considered henceforth.

2.1 Representation Scheme

Given the space of feasible problem solutions $U=\{x \in R^n;\ max_i<x_i<min_i\ \ i=1,\dots,n\}$, and a fitness function $f(x)$:U→R, each candidate solution is expressed as an n-dimensional vector of decision variables $x=\{x_1,\dots,x_n\}$.

2.2 Initialisation Routine

The initial population is randomly distributed with uniform probability across the solution space. The population is fixed to *ns* scout bees. Each scout assesses the fitness of the visited site. The algorithm then enters the main loop. The sequence of evolution cycles is interrupted when the stopping criterion is met.

2.3 Waggle Dance Routine

The population is ranked, and the scout bees that visited the *nb* locations of highest fitness are selected. Each of these *nb* scouts recruit nest mates for local exploration of the flower patch it discovered. The number of recruited bees is allocated deterministically. The first n*e* elite (top-rated) sites amongst the best n*b* locations discovered by the scouts are allocated *nre* foragers, and the remaining (*nb-ne*) sites are allocated $nrb \leq nre$ foragers.

2.4 Local Search Routine

For each of the *nb* selected flower patches, the following procedure is repeated. The recruited foragers (*nre* for the elite sites and *nrb* for the others) are sequentially placed with uniform probability in a neighbourhood of the high fitness location marked by the scout bee. This neighbourhood is defined as an *n*-dimensional hyper-box of sides $a_1,\ldots,a_n$ centred on the position indicated by the scout. If one of the foragers lands in a position of higher fitness than the scout bee, that recruited bee is chosen as the new scout. At the end, only the fittest bee of each flower patch is retained. This bee becomes the dancer once back at the hive.

2.4.1 Neighbourhood Shrinking Subroutine

The size $a=\{a_1,\ldots,a_n\}$ of the flower patches is initially set to a large value. For each variable a_i, it is set as follows:

$$
\begin{aligned}
&a_i(t) = ngh(t) * (\max_i - \min_i) \\
&ngh(0) = 1.0
\end{aligned}
\tag{1}
$$

where t denotes the t^{th} iteration of the Bees Algorithm main loop. The size of a patch is kept unchanged as long as the local search procedure yields higher points of fitness. If the local search fails to bring any improvement in fitness, the size a is decreased. The neighbourhood size is updated according to the following heuristic formula:

$$
ngh(t+1) = 0.8 * ngh(t) \tag{2}
$$

That is, the local search is initially defined over a large neighbourhood, and has a largely explorative character. As the optimisation process advances, the search is made increasingly detailed (exploitative) to refine the current local optimum.

2.4.2 Site Abandonment Subroutine

The neighbourhood shrinking procedure is applied each time the local search procedure fails to yield fitness improvement in a flower patch. After a pre-defined number (*stlim*) of consecutive stagnation cycles, the search procedure is assumed to have found the local fitness peak. In this case, the local search is ended and a new flower patch centred on a randomly generated solution is created. If the location being abandoned corresponds to the best-so-far fitness value, the location of the peak is recorded. If no better solution is found during the remaining of the search, the recorded best fitness site is taken as the final solution.

2.5 Global Search Routine

In the global search phase, *ns*-*nb* bees are randomly scattered across the fitness landscape to evaluate new solutions.

2.6 Population Update Routine

At the end of an iteration, the new population of the bee colony is formed out of two groups. The first group comprises the *nb* bees associated with the centre (the best solution) of each flower patch, and represents the results of the local exploitative search. The second group is composed of the *ns*-*nb* scout bees associated with a randomly generated solution, and represents the results of the global explorative search.

2.7 Stopping Criterion

The stopping criterion depends on the problem domain, and can be either the location of a solution of fitness above a pre-defined threshold, or the completion of a pre-defined number of evolution cycles.

3 APPLICATION EXAMPLES

The Bees Algorithm is used to solve two real-world optimisation tasks. The first task requires the optimisation (training) of the weights of two multi-layer perceptron (MLP) (Lippmann 1987) neural networks to predict the energy efficiency (heating load and cooling load) of buildings. The energy efficiency of a building is evaluated as a function of eight construction parameters, and the optimisation task entails the minimisation of the prediction error. The second task requires finding the minimal energy structure of a molecular cluster to find the configuration of a protein.

3.1 Energy efficiency prediction task

A data set of 768 samples of eight real-valued construction parameters (the independent variables) and two energy real-valued efficiency parameters (heating load and cooling load, the dependent variables) was created by A. Xifara and A. Tsanas at University of Oxford, UK, and is available through the UCI Machine Learning repository (http://archive.ics.uci.edu/ml).

The energy efficiency of a building is determined using one separate MLP to predict each of the two independent variables from the corresponding eight construction parameters. The two ANNs are trained to minimise the prediction error using the standard Backpropagation algorithm (Lippmann 1987) and the Bees Algorithm, and the results of 10 independent trials of the two procedures are compared. For each trial, the data set is randomly divided into a training set containing 80% (614) of the entries, and a test set containing the remaining (154) data samples. The structure of the MLPs is optimised by trial and error, and comprises eight input neurons (one per construction parameter), one layer of 10 hidden units, and one output variable (heating load or cooling load). The neurons of the hidden and output layers use respectively hyper-tangent and sigmoidal activation functions. The settings of the two training algorithms are detailed in Table 1.

Table 2 reports the median and percentiles of the root mean square (RMS) errors obtained in the 10 trials using BP and Bees Algorithm. For each output, the statistical significance of the difference of the results obtained by the two algorithms is tested using Mann-Whitney U tests (Table 2). In both cases the p-value is well above the standard significance threshold of 5% ($p<0.05$), and the null hypothesis cannot be rejected. Using the nonlinear non-parametric Random Forests method, Xifara and Tsanas (2012) obtained a mean average error of 0.51 with a standard deviation of 0.11 on the heating load prediction task, and 1.42 ± 0.25 on the cooling load prediction task. Overall, it can be concluded that the Bees Algorithm obtains results that are comparable to those obtained by competing methods presented in the literature.

The evolution of the average RMS prediction error on the two dependent variables is shown in Figure 1, together with two sample plots of modelling results.

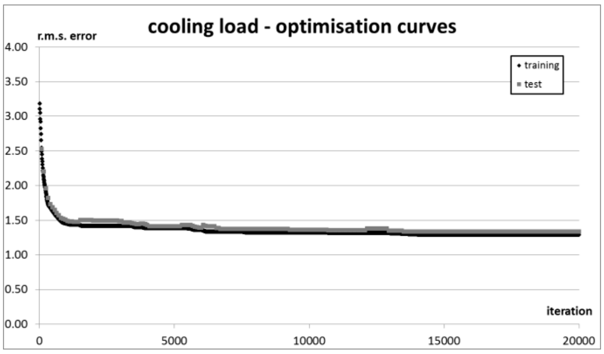
Table 1: Energy efficiency benchmark - BP rule and Bees Algorithm settings

BP Settings		Bees Algorithm Settings		Common settings	
Trials	10	Trials	10	Init. range for MLP weights	[-0.05, 0.05]
Learning cycles	100,0000	Evolution cycles (*lc*)	20,000	Training set	80%
Learning coefficient	0.1	Scout bees (*ns*)	8	Test set	20%
Momentum term	0.01	Elite sites (*ne*)	2		

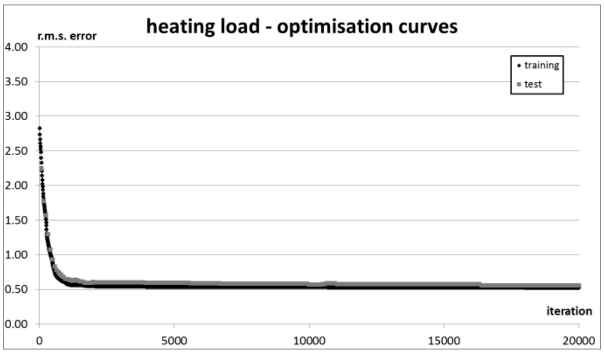
		Best sites (*nb*)	8		
		Recruited elite (nre)	75		
		Recruited best (*nrb*)	25		
		Stagnation limit (*stlim*)	5		

Table 2: Energy efficiency benchmark - Minimisation results

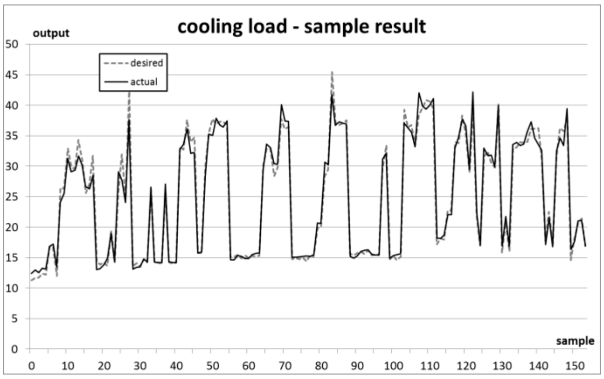
output	algorithm	10th percentile	median	90th percentile	M-W test
heating load	BP	0.46	0.52	0.57	p=0.1655
heating load	Bees Algorithm	0.47	0.55	0.65	
cooling load	BP	1.14	1.31	1.62	p=0.5787
cooling load	Bees Algorithm	1.26	1.33	1.64	



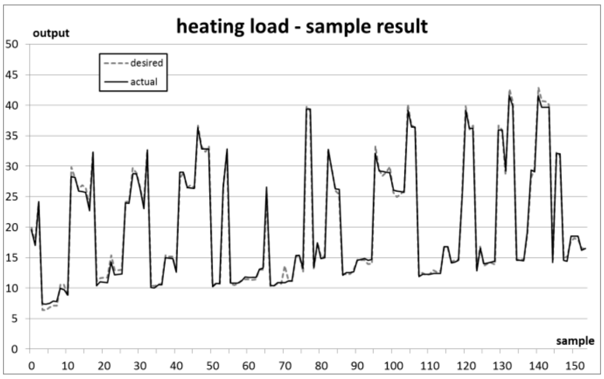
a) output 1 - optimisation curves



b) output 2 - optimisation curves



a) output 1 - sample MLP modelling result



b) output 2 - sample MLP modelling result

Figure 1: Optimisation curves and results of the Bees Algorithm

3.2 Protein folding benchmark

The protein folding problem consists of finding the minimal energy structure of a molecular cluster to determine the configuration of a protein (Vavasis, 1994). The energy function of the molecular structure is defined by the following non-linear partially separable function:

$$f(x_1, \dots, x_n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} r\left[\sqrt{\sum_{k=1}^{3}\left(x_{ik} - x_{jk}\right)^2}\right] \tag{3}$$

where $r(s)$ is the Lennard-Jones potential

$$r(s) = s^{-12} - 2 \cdot s^{-6} \tag{4}$$

The vectors $x_l = \{x_{l1}, x_{l2}, x_{l3}\}, l \in [1, n], l \in I^+$ correspond to the positions of *n* atoms in the 3D space. The function *f* is non-convex, and has an exponential number of local minima (Mongeau, et al. 2000). The global minimum is easy to determine for $x \leq 4$, whilst it is unknown for $x > 4$ (Vavasis, 1994) (Mongeau, et al. 2000).

In this study, the four cases n=3-6 are considered. They will be henceforth called pf3, pf4, pf5, and pf6. The global minima for the four functions are given in Table 4, for n>4, they are taken from the optimisation results of Coleman et al. (1994).

The solutions are encoded using $3 \cdot n$ long strings of real numbers, which are built chaining the 3D vectors of Cartesian coordinates of atoms positions. The Bees Algorithm is employed to locate the minima of the functions. The algorithm is stopped when the known minimum has been approximated with a precision better than 0.001, or cmax optimisation cycles have elapsed. The optimisation parameters of the Bees Algorithm are given in Table 3.

The results of 50 independent optimisation trials are given in Table 4. The table reports the number of successful trials (the solution found approximates the known minimum with a precision better than 0.001), the average error (difference between found solution and known minimum), the average number of iterations needed to find the known minimum, and the known minimum.

Table 4 shows that the Bees Algorithm is able to locate consistently and accurately the global minimum of the four protein folding minimisation problems. Figure 2 shows the fitness progress of the Bees Algorithm in the four minimisation tasks.

In general, the results obtained by the Bees Algorithm compare well with the literature. For a reference, the reader may compare the results plotted in figures 2a-d with those obtained by other six public domain optimisation methods reported by Mongeau et al. (2000).

4 CONCLUSIONS

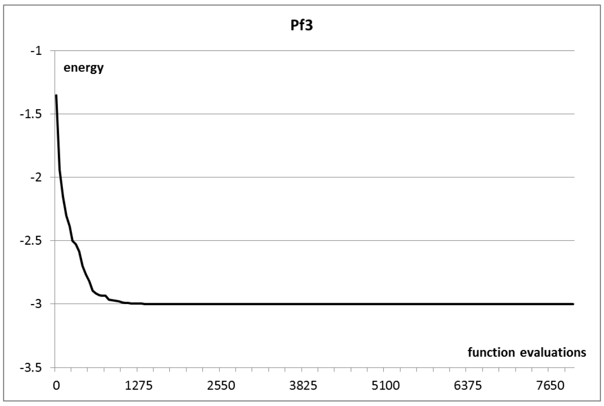
This paper presented the Bees Algorithm, a recently developed nature-inspired parameter learning method. The effectiveness of the Bees Algorithm was demonstrated on two examples of real-world minimisation problems: the training of MLPs to predict the heating load and cooling load (energy efficiency) of buildings (minimisation of average prediction error), and the solution of the protein folding problem (minimisation of energy in molecular dynamics). The results obtained using the Bees Algorithm were comparable to those attained using state-of-the-art techniques, confirming the ability of the Bees Algorithm to solve complex optimisation tasks.

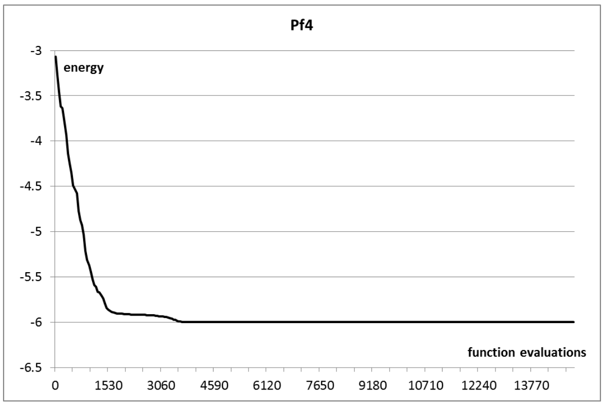
Table 3: Protein folding benchmark problem - Bees Algorithm settings

Function	Colony size	*cmax*	*stlim*	*ne*	*nre*	*nb*	*Nrb*
pf3	51	10000	10	2	20	4	5
pf4	51	10000	10	2	20	4	5
pf5	51	10000	10	2	20	4	5
pf6	102	5000	10	1	40	4	20

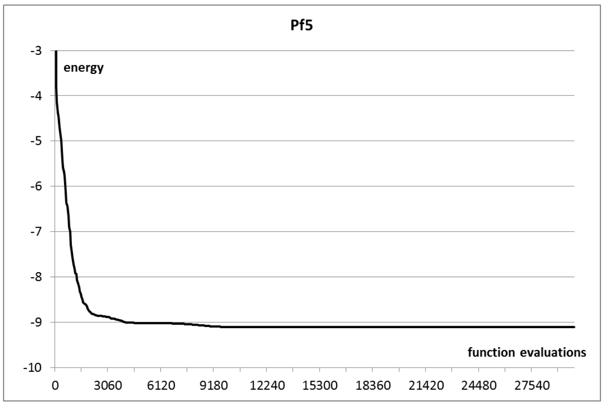
Table 4: Protein folding benchmark problem - minimisation results

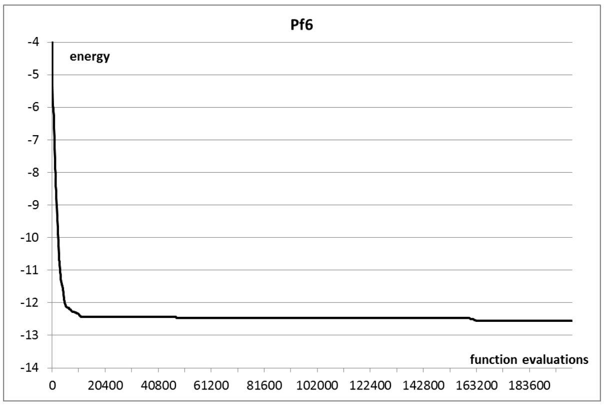
Function	Bees Algorithm			Known minimum
	success	mean error	speed	
pf3	50	0.0000	1454	-3.0
pf4	49	0.0007	2346	-6.0
pf5	50	0.0000	3188	-9.1038
pf6	48	0.0008	148642	-12.712





a) *pf3* function b) *pf4* function





a) *pf5* function b) *pf6* function

Figure 2: Optimisation curves of the Bees Algorithm

REFERENCES

Poon T., Choy, K., Chan, F. and Lau, H. (2011), "A real-time production operations decision support system for solving stochastic production material demand problems", Expert Systems with Applications, vol. 38, no. 5, pp. 4829-4838.

Bonabeau, E., M. Dorigo, and G. Theraulaz. 1999. *Swarm intelligence: from natural to artificial systems.* New York: Oxford University Press.

Coleman, T., D. Shalloway, and Z. Wu. 1994. A parallel build-up algorithm for global energy minimizations of molecular clusters using effective energy simulated annealing. *Journal of Global Optimization* 4: 171-185.

Fogel, L.J., A.J. Owens, and M.J. Walsh. 1966. *Artificial intelligence through simulated evolution.* New York: Wiley.

Gross, R., and M. Dorigo. 2009. Towards group transport by swarms of robots. *International Journal Bio-Inspired Computation* 1: 1-13.

Holland, J.H. 1975. *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press.

Jevtic, A., and D. Andina. 2010. Adaptive artificial ant colonies for edge detection in digital images. In *Proceedings IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society.* Glendale, AZ. 2813 – 2816.

Kennedy, J. 2006. Swarm Intelligence. In *Handbook of Nature-Inspired and Innovative Computing*, ed. A. Zomaya, 187-219. USA: Springer.

Lippmann, R.P. 1987. An introduction to computing with neural nets. *IEEE ASSP Magazine*: 4-22.

Mongeau, M., H. Karsenty, V. Rouzé, and J.B. Hiriart-Urruty. 2000. Comparison of public-domain software for black box global optimization." *Optimization Methods and Software* 13: 203-226.

Pham, D.T., and M. Castellani. 2009. The Bees Algorithm – Modelling Foraging Behaviour to Solve Continuous Optimisation Problems. *Proceedings of the Institution of Mechanical Engineers, Part C* 223: 2919-2938.

Pham, D.T., and X. Liu. 1995. *Neural Networks for Identification, Prediction and Control.* London, UK: Springler-Verlag.

Pham, D.T., S. Otri, E. Koc, A. Ghanbarzadeh, S. Rahim, and M. Zaidi. 2005. *The Bees Algorithm.* Cardiff, UK: Manufacturing Engineering Centre, Cardiff University.

Rechenberg, I. 1965. *Cybernetic Solution Path of an Experimental Problem.* Library Translation no. 1122, Farnborough, Hants UK: Ministry of Aviation, Royal Aircraft Establishment.

Reynolds, C. 1987. Flocks, herds and schools: A distributed behavioural model. *Computer Graphics* 21: 25–34.

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

Tereshko, V., and A. Loengarov. 2005. Collective Decision-Making in Honey Bee Foraging Dynamics. *Journal of Computing and Information Systems* 9: 1-7.

Tsanas, A., and A. Xifara. 2012. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools." *Energy and Buildings* 49: 560-567.

Vavasis, , S.A. 1994. Open problems. *Journal of Global Optimization* 4: 343-344.

Yang, X.S. 2010. Nature-Inspired Metaheuristic Algorithms, 2nd Edition. Frome, UK: Luniver Press.