Cuckoo Search for Business Optimization Applications

Xin-She Yang[1], Suash Deb[2], Mehmet Karamanoglu[1] and Xingshi He[3]

1) School of Science and Technology, Middlesex University, Hendon Campus, London NW4 4BT, UK
2) Cambridge Institute of Technology, Cambridge Village, Tatisilwai, Ranchi-835103, Jharkhand, India
3) School of Science, Xi'an Polytechnic University, No. 19 Jinhua South Road, Xi'an, China

Abstract— **Cuckoo search has become a popular and powerful metaheuristic algorithm for global optimization. In business optimization and applications, many studies have focused on support vector machine and neural networks. In this paper, we use cuckoo search to carry out optimization tasks and compare the performance of cuckoo search with support vector machine. By testing benchmarks such as project scheduling and bankruptcy predictions, we conclude that cuckoo search can perform better than support vector machine.**

Keywords-algorithm; cuckoo search; optimization; swarm intelligence; metaheuristics

I. INTRODUCTION

Almost in all optimization applications, an efficient algorithm is essential; however, in many cases, there may be no such efficient algorithm at all. In business applications, optimization problems are often large scale with massive data sets [1,2]. To extract useful information among a huge amount of data requires efficient tools for processing vast data sets [3]. This is equivalent to trying to find an optimal solution to a highly nonlinear problem with multiple, complex constraints, which is a challenging task [4].

From the optimization point of view, metaheuristic algorithms have become powerful for solving tough nonlinear optimization problems [4,5]. Modern metaheuristic algorithms have been developed with an aim to carry out global search, typical examples are particle swarm optimisation (PSO) [6], firefly algorithm [7], cuckoo search [8,9,10], accelerated PSO [11], bat algorithm [12] and many other algorithms and their variants [13,14,15]. The efficiency of metaheuristic algorithms can be attributed to the fact that they imitate the best features in nature, especially the selection of the fittest in biological systems which have evolved by natural selection over millions of years [4,16].

Various techniques for data mining and business optimization have been developed over the past few decades [17,18]. Among these techniques, recent studies indicated that support vector machine is one of the best techniques for regression, classification and data mining [19,20].

Metaheuristic algorithms have attracted much attention and new algorithms appear almost on a yearly basis. For example, in 2008, Yang developed the firefly algorithm (FA), which mimics the flashing behaviour of tropic fireflies [4,7]. Firefly algorithm has many advantages over particle swarm optimization, one of which is the automatic subdivision of the whole population into many subgroups, and each subgroup can potentially swarm around each optimum. This often ensures that all optima, including the global optimality, can be obtained simultaneously if there are enough fireflies in the population. As another example, Yang and Deb introduced an efficient cuckoo search (CS) algorithm in 2009 [8,9]. CS is far more effective than most existing metaheuristic algorithms, including particle swarm optimization [6] and genetic algorithm [5].

On the other hand, business optimization often concerns with massive but often incomplete data sets, evolving dynamically over time. Some tasks cannot start before other required tasks are completed, such complex scheduling is often NP-hard, and no universally efficient tool exists. Recent trends indicate that metaheuristics can be very promising, in combination with other tools such as neural networks and support vector machines [20-22]. Business management and many other applications also require efficient techniques for quantitative modelling and predictions [1, 11,23,24,25].

In this paper, we use cuckoo search to carry out business optimization and compare the performance of cuckoo search with that of support vector machine for two benchmarks and a design problem. We first will outline the fundamentals of cuckoo search and standard support vector machine, and then use these studies to test the proposed approach. Finally, we discuss the implications and possible extension for further research.

II. METEHEURISTIC ALGORITHMS

A. Cuckoo Search

Cuckoo search (CS) is one of the latest nature-inspired metaheuristic algorithms, developed in 2009 by Xin-She Yang and Suash Deb [8, 10]. CS is based on the brood parasitism of some cuckoo species. In addition, this algorithm is enhanced by the so-called Lévy flights, rather than by simple isotropic random walks. Recent studies show that CS is potentially far more efficient than PSO and genetic algorithms [4-11].

Cuckoos are fascinating birds, not only because of the beautiful sounds they can make, but also because of their aggressive reproduction strategy. Some species such as the *ani* and *Guira* cuckoos lay their eggs in communal nests, though they may remove others' eggs to increase the hatching probability of their own eggs. Quite a number of species engage the *obligate brood parasitism* by laying their eggs in the nests of other host birds (often other species).

978-1-4673-1953-9/12/$31.00 ©2012 IEEE

For simplicity in describing the cuckoo search, we now use the following three idealized rules: a) Each cuckoo lays one egg at a time, and deposits it in a randomly chosen nest. b) The best nests with high-quality eggs will be carried over to the next generations. c) The number of available host nests is fixed, and the egg laid by a cuckoo can be discovered by the host bird with a probability p_a. In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new nest.

From the implementation point of view, we can use the following simple representations that each egg in a nest represents a single solution, and each cuckoo can lay only one egg (thus representing one solution), the aim is to use the new and potentially better solutions (cuckoos) to replace not-so-good solutions in the nests. Obviously, this algorithm can be extended to the more complicated cases, where each nest has multiple eggs representing a set of solutions. For this present work, we will use the simplest approach where each nest has only a single egg. In this case, there is no distinction between an egg and a solution, a nest or a cuckoo, as each nest corresponds to one egg, which also represents one cuckoo.

There are two key branches or types of generating new solutions in cuckoo search. Once type is to generate solutions by Lévy flights [26]

$$x_i^{(t+1)} = x_i^{(t)} + \alpha s_L,$$

where S_L is a vector drawn from the Lévy distribution

$$L(s,\lambda) = \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s \gg s_0), \qquad (1)$$

where s_0 is the minimum step size and Γ is a Gamma function. Here $\alpha > 0$ is the step size scaling factor, which should be related to the scales of the problem of interest. Here $L(\lambda)$ is the step size drawn from a Lévy distribution.

In general, Lévy flights are governed by

$$L(\beta,\lambda) = \frac{1}{\pi}\int_0^\infty \cos(ks)\exp[-\beta\,|k|^\lambda]dk, \qquad (2)$$

and this integral does not have any explicit form analytically. Thus, it is very difficult to draw random samples. However, under the approximation $s>>s_0>0$, we have the approximation

$$L(\beta,\lambda) \approx \frac{\beta\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi\,|s|^{1+\lambda}}. \qquad (3)$$

For simplicity, we can set $\beta = 1$, and the above expression becomes that given in (1) for many applications.

The other branch of solution generation is that new solutions are generated by using the similarity between the existing eggs/solutions and the host eggs/solutions with a discovery rate p_a. This can be represented mathematically as

$$x_i^{t+1} = x_i^t + s \otimes H(p_a - \varepsilon) \otimes (x_j^t - x_k^t), \qquad (4)$$

where x_i, x_j and x_k are three different solutions. Here $H(u)$ is a Heaviside function of u, and ε is a random number drawn from a uniform distribution in [0,1]. Again s is the step size vector.

Lévy flights are more efficient than Brownial random walks in exploring unknown, large-scale search space. There are many reasons to explain this high efficiency, and one simple reason is that its variance is unbounded as it increases with iterations t in the following manner

$$\sigma^2(t) \sim t^{3-\lambda}, \quad 1 \le \lambda \le 2, \qquad (5)$$

which increases much faster than the linear relationship $\sigma^2 \sim t$ of Brownian random walks.

Many algorithms exist for generating Lévy flights in the literature, however, we found that Mantegna's algorithm works very well [26]. In this algorithm, the step length s can be calculated by

$$s = \frac{u}{|v|^{1/\lambda}}, \qquad (6)$$

where u and v are drawn from normal distributions. That is,

$$u \sim N(0,\sigma_u^2), \quad v \sim N(0,\sigma_v^2),$$

where

$$\sigma_u = \{\frac{\Gamma(1+\lambda)\sin(\pi\lambda/2)}{\Gamma[(1+\lambda)/2]\,\lambda\,2^{(\lambda-1)/2}}\}^{1/\lambda}, \quad \sigma_v = 1.$$

This algorithm will generate samples that obey the expected Lévy distribution for $s>s_0$ where s_0 is the smallest step. In theory, $s_0>>0$, but in practice, s_0 can be taken as a small value such as $s_0 = 0.1$.

B. Support Vector Machine

A support vector machine essentially transforms a set of data into a significantly higher-dimensional space by nonlinear transformations so that the regression and data fitting can be carried out in this high-dimensional space. This methodology can be used for data classifications, pattern recognition and regressions, and its theory was based on the statistical machine learning theory [18, 19,27].

For classifications with the learning examples or data as (x_i, y_i) where $i=1,2,\ldots,n$ and $y_i \in \{-1,+1\}$, a linear support vector machine can be used by constructing two hyperplanes as far away as possible, and no samples should be between these two planes. Mathematically, this is equivalent to two equations

$$wx + b = \pm 1, \qquad (7)$$

and the main objective of constructing these two hyperplanes is to maximize the distance (between the two planes). From the optimization point of view, the maximization of margins can be written as

$$\text{minimize} \quad \Psi = \frac{1}{2}\|w\|^2 + \lambda\sum_{i=1}^n \eta_i,$$

subject to

$$y_i(wx_i + b) \ge 1 - \eta_i,$$

$$\eta_i \ge 0, \qquad (i = 1,2,\ldots,n),$$

where λ>0 is a parameter to be chosen appropriately. Here, the term $\sum_{i=1}^{n}\eta_i$ is essentially a measure of the upper bound of the number of misclassifications on the training data. For most problems in nonlinear support vector machine, we can use $K(x,x_i)=(x\bullet \dot{x}_i)^d$ for polynomial classifiers, $K(x,x_i)=\tanh[k(x,x_i)+\Theta)]$ for neural networks, and by far the most widely used kernel is the Gaussian radial basis function (RBF)

$$K(x,x_i)=\exp[-\frac{\| x-x_i \|^2}{(2\sigma^2)}]=\exp[-\gamma \| x-x_i \|^2], \quad (8)$$

for nonlinear classifiers. This kernel can easily be extended to any high dimensions. Here, σ^2 is the variance and $\gamma=1/2\sigma^2$ is a constant. Following the similar procedure as discussed earlier for linear SVM [18,19], we can obtain the coefficients α_i by solving the following optimization problem

$$\text{maximize} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\alpha_i\alpha_j y_i y_j K(x_i,x_j). \quad (9)$$

It is worth pointing out under Mercer's conditions for the kernel function, the matrix $A=y_i y_j K(x_i,x_j)$ is a symmetric positive definite matrix [18], which implies that the above maximization is a quadratic programming problem, and can thus be solved efficiently by standard QP techniques [27].

III. Business Optimization and Predictions

Following on from the above, we use two benchmarks and a design problem to test the performance of cuckoo search versus support vector machine. We have validated our implementations using the standard test functions, which confirms the correctness of the implementation. The first case study is a standard benchmark in resource-constrained project scheduling [23,24], while the second case study is a heat exchanger design problem and the third case study for bankruptcy prediction.

There are three parameters in cuckoo search [8,9], and they are population size n, discovery rate p_a and Lévy flight exponent λ. In the rest of the simulations, we used fixed values of $n=40$, $p_a=0.5$, and $\lambda=1.5$ except for the second case study where $n=20$ has been used. On the other hand, we used C=127.9 and σ^2=64 for the kernel parameters in support vector machine.

A. Project Scheduling

Scheduling is an important class of discrete optimization with a wider range of applications in business intelligence. For resource-constrained project scheduling problems, there exists a standard benchmark library by Kolisch and Sprecher [23, 24]. The basic model consists of J activities/tasks, and some activities cannot start before all its predecessors h are completed. In addition, each activity $j=1,2,...,J$ can be carried out, without interruption, in one of the M_j modes, and performing any activity j in any chosen mode m takes d_{jm} periods, which is supported by a set of renewable resource R, and non-renewable resources N. The project's makespan or upper bound is T, and the overall capacity of non-renewable resources is K_r^v where $r \in N$. For an activity j scheduled in mode m, it uses k^ρ_{jmr} units of renewable resources and k^v_{jmr} units of non-renewable resources in period $t=1,2,...,T$.

Using the online benchmark library [13], we have solved this type of problem with $J=30$ activities (the standard test set j30). The run time on a modern desktop computer is about 2.7 seconds for $N=1000$ iterations to 15.9 seconds for $N=5000$ iterations. We have run the simulations for 50 times so as to obtain meaningful statistics. The deviations from the known best solution are given in Table I where the results by other methods are also compared.

TABLE I: Mean deviations from the optimal solution (J=30).

Algorithm	N=1000	N=5000
PSO [25]	0.26	0.21
Hybrid GA[17]	0.27	0.06
Adapting GA [20]	0.38	0.22
Current CS	*0.29*	*0.054*

From this table, we can see that cuckoo search start very well, with results comparable with those by other methods such as hybrid genetic algorithm; but it converges more quickly as the number of iterations increase, and much better results are obtained.

B. Heat Exchanger Design Optimization

Let us try to solve a nonlinear design optimization problem in engineering applications using cuckoo search. Such design problems may be difficult to solve by using support vector machine because SVM is mainly for classification and regression. The heat exchanger design is a challenging task [12], which can be expressed in the simplest case as the following minimization problem with eight design variables:

$$\text{Minimize} \quad f(\boldsymbol{x})=x_1+x_2+x_3$$

subject to

$$g_1=0.0025(x_4+x_6)-1\le 0,$$
$$g_2=0.0025(x_5+x_7-x_5)-1\le 0,$$
$$g_3=0.01(x_8-x_5)-1\le 0,$$
$$g_4=833.33252x_4+100x_1-x_1x_6-83333.333\le 0,$$
$$g_5=1250x_5+x_2x_4-x_2x_7-125x_4\le 0,$$
$$g_6=x_3x_5-2500x_5-x_3x_8+1250000\le 0.$$

Using the cuckoo search algorithm with $n=20$ cuckoos, we can easily find the optimal solution for these eight design variables as $\boldsymbol{x}^*$=(579.3068, 1359.9708, 5109.9705, 182.0177, 295.6012, 217.9823, 286.4165,395.6012). This solution is better than those found in the literature [12].

C. Bankruptcy Prediction

Evaluations of business performance of a company, management of risks, and credit rating are an essential part of modern business intelligence and activities [2,3]. Business activities involve a large amount of data spanning many different types of data. In order to extract meaningful knowledge, data-mining techniques are specially useful. For example, for corporate bankruptcy predictions, the classical Altman model used five most important factors or ratios, called Altman's Z-scores [2]. In contrast, about 23 different factors are identified [28] and the actual factors and the number of key factors vary with different regions internationally.

Altman's bankruptcy model can be written as

$$Z = 1.2T_1 + 1.4T_2 + 3.3T_3 + 0.6T_4 + 0.999T_5 \qquad (9)$$

where T_1 is the ratio of working capital to total assets of a corporate firm, T_2 is the ratio of retained earnings to total assets, T_3 is earnings before interest and taxes to the total assets. In addition, T_4 is the ratio of market value of equity to total liabilities, while T_5 is the ratio of sales to total assets. The empirical rule is that a company is considered as safe if $Z>2.99$, while there is a risk of bankruptcy if $Z<1.80$. In the range of $1.8<Z<2.99$, it is considered as a grey zone. This model typically has the accuracy around 70%, up to 80 or 90% in special cares.

TABLE II: COMPARISON OF ALTMAN'S MODEL WITH CS PREDICTIONS.

	T_1	T_2	T_3	Z	SVM
1	0.083	0.179	0.306	-1	-1
2	0.013	0.091	0.054	-1	-1
3	0.232	0.476	0.563	1	1
4	0.007	0.032	0.040	-1	-1
5	0.203	0.029	0.072	-1	-1
6	0.479	0.555	0.703	1	1
7	0.059	0.087	0.165	-1	-1
8	0.006	0.087	0.065	-1	-1
9	0.101	0.567	0.011	-1	-1
10	0.599	0.544	0.375	1	1
11	0.780	0.699	0.283	1	1
12	0.675	0.041	0.595	1	1
13	0.091	0.006	0.223	-1	-1
14	0.619	0.013	0.126	-1	-1
15	0.024	0.105	0.051	-1	-1
16	0.056	0.208	0.377	1	1
17	0.043	0.015	0.204	-1	-1
18	0.006	0.039	0.004	-1	-1
19	0.132	0.041	0.209	-1	-1
20	0.514	0.262	0.324	1	1

As another case study, we now use support vector machine to predict the possibility of corporate bankruptcy. From a simulated database of 20 different companies. We first train the support vector machine using the first 14 data points and then use the other 6 sets for validating predictions. The comparison and predictions are summarized in Table II where Z-status means the status predicted by the Z-scores [2, 28]. In addition, the same data sets were used to first train cuckoo search to find the optimal regression by minimizing the predictions and the known values. Then, the optimal regression obtained by CS is used for predicting the rest 6 data sets.

From Table II, we can see that SVM can predict reasonably well with relatively small number of training data. At the same time, cuckoo search can also make predictions with almost identical accuracy, without the need for any parameter tuning before training from a subset of the data. This may be advantageous because parameter tuning in SVM and the choice of training sample size may be an important issue, while cuckoo search essentially can bypass this parameter tuning problem by solving an optimization problem. However, further studies are needed to extend this to wider applications with real-world data and business settings. Furthermore, sensitivity studies may be needed to confirm this conclusion.

IV. CONCLUSIONS

Business applications can be solved in many ways using support vector machine, artificial neural networks, and optimization algorithms. The use of support vector machine and neural networks often require a substantial subset of the data to be used for training initially and for parameter tuning, then the rest of the data can be used for predictions. On the other hand, optimization algorithms such as cuckoo search can solve these problem equally well. In this sense, cuckoo search is better than support vector machine and they can produce equally accurate predictions. Furthermore, cuckoo search can also solve other types of optimization problems such as designing heat exchangers in engineering applications.

A further improvement is to identify the type of efficient algorithm to suit different purposes. A more detailed and extensive study is required to compare various commonly used methods for business intelligence, including neural networks, support vector regression, subset selection, multiple regression and capacity control. However, validation of these algorithms using real-world, large-scale problems can be very challenging but will be extremely useful.

REFERENCES

[1] M. Ahuja, B. Gupta and P. Raman, "An Empirical Investigation of Online Consumer Purchasing Behavior", *J. Comm. Assoc. for Computing Machinery (CACM)*, vol. 46(12), 2003, pp. 145-151.

[2] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy", *Journal of Finance*, Vol. 23, 1968, pp. 189-209.

[3] J. B. Couette, E. I. Altman, P. Narayanan, *Managing Credit Risk -- the Next Great Financial Challenge*, John Wiley & Sons, New York, 1998.

[4] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2008.

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Reading, Mass.: Addison Wesley, 1989.

[6] J. Kennedy and R. C. Eberhart, “Particle swarm optimization”, *Proc. of IEEE International Conference on Neural Networks*, Piscataway, NJ., 1995, pp. 1942-1948.

[7] X. S. Yang, “Firefly algorithms for multimodal optimization*”, Proc. 5th Symposium on Stochastic Algorithms, Foundations and Applications*, SAGA 2009, Eds. O. Watanabe and T. Zeugmann, Lecture Notes in Computer Science, vol. 5792, 2009, pp.169-178.

[8] X. S. Yang and S. Deb, “Cuckoo search via Lévy flights”, in: *Proceeings of World Congress on Nature & Biologically Inspired Computing* (NaBIC 2009, India), IEEE Publications, USA, 2009, pp. 210-214.

[9] X. S. Yang and S. Deb, “Engineering optimization by cuckoo search”, *Int. J. Mathematical Modelling and Numerical Optimisation*, vol. 1, 2010, pp. 330-343.

[10] A. H. Gandomi, A. H., Yang, X. S., Alavi A. H., Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Engineering with Computaters*, **27**, DOI: 10.1007/s00366-011-0241-y(2011)

[11] X. S. Yang, S. Deb, and S. Fong, “Accelerated particle swarm optimization and support vector machine for business optimization and applications”, in: *Networked Digital Technologies* (NDT2011), Communications in Computer and Information Science, vol 136, Springer, 2011, pp. 53-66.

[12] X. S. Yang and A. H. Gandomi, “Bat algorithm: a novel approach for global engineering optimization”, Engineering Computations, vol. 29, No. 5, 2012, pp.464-483.

[13] S. Walton, O. Hassan, K. Morgan, “Modified cuckoo search: a new gradient-free optimisation algorithm”, *Chaos, Solitons & Fractals*, vol. 44, 2011, pp.710-718.

[14] A. Goghrehabadi, M. Ghalambaz and A. Vosough, “A hybrid power series -- Cuckoo search optimization algorithm to electrostatic deflection of micro fixed-fixed actuators”, Int. J. Multidisciplinary Science and Engineering, vol. 2, No. 4,2011, pp.22-26.

[15] A. Layeb, “A novel quantum-inspired cuckoo search for Knapsack problems”, Int. J. Bio-inspired Computation, vol. 3, 2011, pp.297-305.

[16] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptural comparision”, *ACM Comput. Surv.*, vol. 35, 2003, pp. 268-308.

[17] V. Valls, F. Ballestin and S. Quintanilla, “A hybrid genetic algorithm for the resource-constrained project scheduling problem”, *Euro. J. Oper. Res.*, doi:10.1016/j.ejor.2006.12.033, (2007).

[18] V. Vapnik, *The nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[19] G. R. Shi and X. S. Yang, “Optimization and data mining for fracture prediction in geosciences”, *Procedia Computer Science*, vol. 1, 2010, pp. 1353-1360.

[20] S. Hartmann, “A self-adapting genetic algorithm for project scheduling under resource constraints”, *Naval Res. Log.*, vol. 49, 2002, pp.433-448.

[21] T. Howley and M. G. Madden, “The genetic kernel support vector machine: description and evaluation”, *Artificial Intelligence Review*, vol. 24, 2005, pp.379-395.

[22] K. Kim, “Financial forecasting using support vector machines”, *Neurocomputing*, vol. 55, 2003, pp.307-319.

[23] R. Kolisch and A. Sprecher, “PSPLIB - a project scdeluing problem library, OR Software-ORSEP” (operations research software exchange prorgam) by H. W. Hamacher, *Euro. J. Oper. Res.*, vol. 96, 1996, pp. 205-216.

[24] R. Kolisch and A. Sprecher, The Library PSBLIB, http://129.187.106.231/psplib/

[25] S. K. Tchomté, M. Gourgand and A. Quilliot, “Solving resource-constrained project scheduling problem with particle swarm optimization”, in: *Proceeding of 3rd Multidsciplinary Int. Scheduling Conference* (MISTA 2007), 28-31 Aug 2007, Paris, 2007, pp. 251-258.

[26] Mantegna, R. N., Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes, *Phys. Rev. E*, **49**, 4677-4683 (1994).

[27] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression”, 1998. http://www.svms.org/regression/

[28] S. Tian and Y. Yu, “Forecasting corporate bankruptcy: an international evidence”, *Proceedings of Joint Statistical Meeting (JSM)*, July 31-Aug. 5, Vancouver, 2010, pp. 307-308.