TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 60, Issue IV, November, 2017

ENHANCED KGCS FOR ENGINEERING DESIGN

Florina ȘERDEAN, Radu MORARIU-GLIGOR, Adina CRIȘAN

Abstract: *This paper presents an optimization algorithm called KGCS suitable to be used for different engineering problems. The efficiency of the algorithm is given by the aggregation between the particularities of the Cuckoo Search algorithm and of the Knowledge Gradient policy. The paper also presents the testing of the KGCS algorithm on benchmark engineering problems, such as the optimal design of welded beams, speed reducers or four bar trusses. The results of the tests are compared to the ones reported in the literature for other evolutionary algorithms to prove the high performance of the KGCS algorithm.*

Key words: *evolutionary algorithm, optimization, engineering design, welded beam, speed reducer, four bar truss.*

1. INTRODUCTION

Optimal design problems are widely spread in all engineering fields due to the fact that it is desired to find the best solution with respect to certain criteria and under given circumstances.

A lot of effort has been put into studying optimization methods for engineering design, see for example [1]. Moreover, a considerable amount of research has been dedicated to developing fast and reliable optimization algorithms that find the optimum solutions of mathematical benchmark functions as well as the optimal design solutions of benchmark engineering design problems with more than two variables and several constraints [2]. If the algorithms have a good performance on the benchmark functions then they are suitable to be used for real world applications.

Several published scientific papers present the tests conducted on one or more engineering design benchmark problems. For example, in [3] the authors present the design optimization of a pressure vessel using a metaheuristic called Ant Colony Optimization. The purpose was cost reduction by minimizing weight and respecting constraints that ensure adequate strength and stiffness. Another example is the optimal design of a speed reducer [4]. The problem is formulated as a minimization problem of the total weight while satisfying constraints such as the limits on the bending stress of the gear teeth, surface stress or transverse deflections of shafts. Other papers present algorithms that have been tested on more design optimization problems such as a Particle Swarm Optimization algorithm [5] or a microgenetic algorithm [6].

2. ALGORITHM DESCRIPTION

2.1 Cuckoo Search algorithm

The class of evolutionary algorithms which were developed based on the evolution and/or functioning of different biological systems have proven to be very efficient for solving optimization problems. One of these evolutionary algorithms is Cuckoo Search algorithm which is inspired by specific egg laying and breeding of cuckoos.

This algorithm has been tested and used for engineering design, a domain of great industrial interest [7].

For the optimizations conducted on the benchmark engineering problems presented in this paper, the Lèvy flights version of the algorithm was implemented. According to this version the optimization algorithm generates a random initial population. One egg is laid in a host bird nest by each cuckoo from the population. Those nests in which the eggs are laid are chosen using Lèvy flights from the current nest of each cuckoo. Obviously, the search space limits are respected. An example of a 2D Lèvy flight is presented in Fig. 1. As it can be seen the flight shows a cluster of small steps around the origin point which are separated by extreme jumps, proving that is does a good local search and it also offers the possibility to search other areas. The evaluation of each new cuckoo chick that hatches from the laid eggs and as well as the value of the best cuckoo is updated. The reality concerning egg laying and breeding of cuckoos shows there is a possibility that the eggs are discovered by the host bird. Therefore, in the program a constant is set and it represents the probability to discover the cuckoo eggs. According to it some of the cuckoo chicks are killed and new eggs are laid. This ensures that the cuckoo population does not migrate too soon towards a certain area that seems to be the best and, therefore, reduces the chances of finding a local optimum in optimization problems. After all these steps are performed, a new cuckoo population is formed by combining the best cuckoos of the cuckoo chicks and the cuckoo parents.

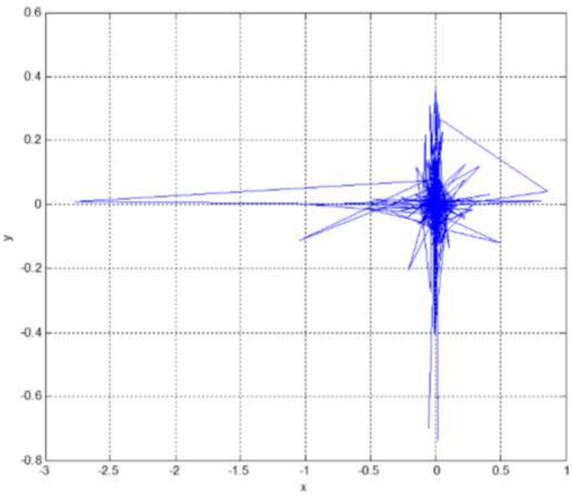

Fig. 1. A 2D Lèvy flight of 500 steps with (0,0) as starting point [8].

The stop condition of the generational process can be formulated in different ways, for example it can be given by a certain number of generations or objective function evaluations. The implemented algorithm was tested on mathematical benchmark functions [8].

2.2 Knowledge Gradient Cuckoo Search (KGCS) algorithm

Usually optimization problems depending on their complexity can require thousands or hundreds of thousands of objective function calls, fact that increases the computational cost. Hence, the purpose is to develop and use algorithms precise, robust, and, of course, extremely e□cient. In order to fulfill this purpose, the already presented version of Cuckoo Search algorithm was improved based on the idea that with greater knowledge, less exploration is necessary. In fact, the growth in knowledge of the cuckoo population during migration was innovatively assessed using the Knowledge Gradient policy presented in more detail in [8, 9].

The enhanced algorithm has two phases. The first one is an exploration phase in which the algorithm uses three cuckoo populations. They are initially randomly generated and then they are let to explore the search space and try identifying the optimal solution independently. After evaluation, the best cuckoo is updated for each of the three populations. During this phase at the end of each generation, an archive with the best cuckoos is updated for each population. This phase lasts for several generations (approximately 5%-10% from the maximum number of generations).

During the second phase, the level of exploration is reduced, while the exploitation is increased. This is achieved by computing the knowledge gradient for each population based on its own archive. Then the search for the optimal solution is continued only by the population which ensures the largest expected improvement according to Knowledge Gradient policy. This improved algorithm was implemented in Matlab and it was tested on mathematical benchmark functions. The results showed and average percentage decrease of the number of required objective function evaluations of 6.13 % when compared to the number of required objective function evaluations when using standard Cuckoo

Search algorithm [8, 9]. An example of the population migration corresponding to the two phases is presented in Fig. 2 for the test conducted on Rosenbrock's 2D function.

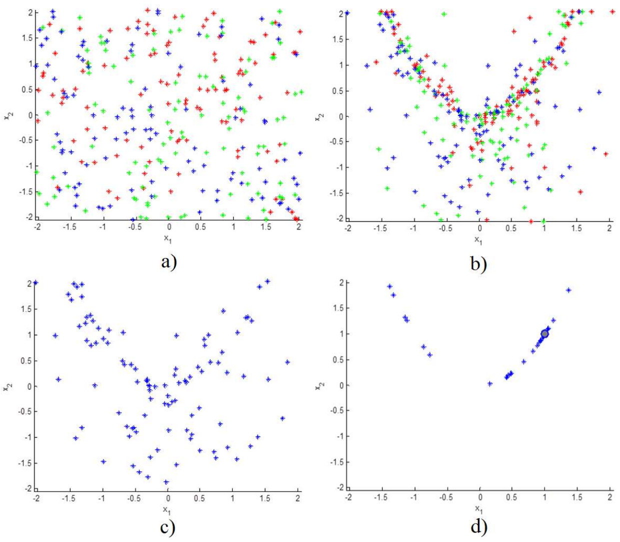

Fig. 2. a) The beginning of the first phase; b) The end of the first phase; c) The beginning of the second phase; d) The end of the second phase [8]

3. OPTIMIZATION PROBLEMS

3.1 Optimization problem formulation

Considering the exponential increase of information and technology nowadays, finding the optimal alternative as fast as possible is a must even in the case of simple decisions. This has caused an increase in the use of implemented algorithms that are capable to solve optimization problems efficiently.

Mathematically, a mono-objective optimization problem can be formulated as follows, without the loss of generality:

$$\min_{x} f(x) \tag{1}$$

subjected to:

$$g_i(x) \geq 0, \ \text{i}=\overline{1,\text{m}} \tag{2}$$

where x is the decision vector, f is the objective function and g_i are the constraint functions.

The proposed algorithm was developed for unconstrained optimization problems. Therefore, a special approach must be defined to handle the constraints. Several constraint-handling methods exist in the literature [10]. The one chosen for the testing on the algorithm on engineering problems consists in adding a penalty term to the value of the objective function. This term is a sum of all the absolute values of the constraint functions violations multiplied by a large penalty factor.

3.2 Welded beam optimal design

Three engineering design problems were selected to test the efficiency and applicability of the enhanced KGCS algorithm for mono-objective design optimization. The first benchmark engineering design problem consists in designing a welded beam (Fig. 3) with minimum cost (see [11]), but respecting the constraints on shear stress, bending stress and buckling load.

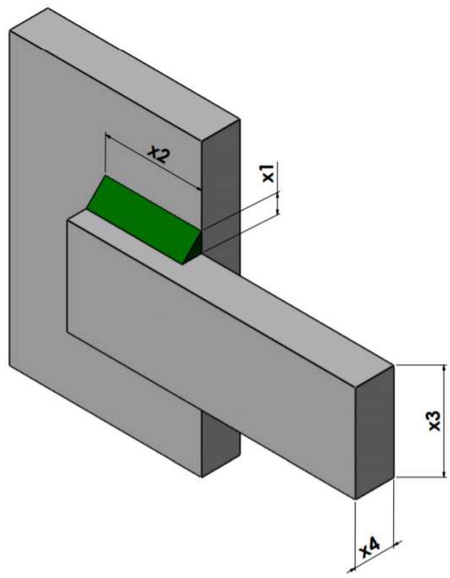

Fig. 3. Welded beam design problem

The problem can be formulated as follows:

$$\min_{x_1,x_2,x_3,x_4} \left(c_1+c_3\right)\cdot x_1^2 \cdot x_2 + c_2 \cdot x_3 \cdot x_4 \cdot \left(L+x_2\right) \tag{3}$$

subjected to

$$\begin{aligned}
&g_1(x_1,x_2,x_3,x_4) = \tau_{\max} - \tau\left(x_1,x_2,x_3,x_4\right) \geq 0 \\
&g_2(x_1,x_2,x_3,x_4) = \sigma_{\max} - \sigma\left(x_1,x_2,x_3,x_4\right) \geq 0 \\
&g_3(x_1,x_2,x_3,x_4) = x_4 - x_1 \geq 0 \\
&g_4(x_1,x_2,x_3,x_4) = 5 - c_1 \cdot x_1^2 - c_2 \cdot x_3 \cdot x_4 \cdot \left(L+x_2\right) \geq 0 \\
&g_5(x_1,x_2,x_3,x_4) = x_1 - 0.125 \geq 0 \\
&g_6(x_1,x_2,x_3,x_4) = \delta_{\max} - \delta\left(x_1,x_2,x_3,x_4\right) \geq 0 \\
&g_7(x_1,x_2,x_3,x_4) = Pc\left(x_1,x_2,x_3,x_4\right) - P \geq 0
\end{aligned} \tag{4}$$

where L is the length of the suspended part of the beam, c_1 is the cost per unit of weld material, c_3 is the labor cost per unit of weld material, c_2 is the cost per volume unit of the beam, τ is the weld stress, σ is the beam bending stress, δ is the deflection when the load P is applied at the free end of the beam, Pc is the beam bucking load and $x_1, x_2 \geq 0.1$, $x_3 \leq 10, x_4 \leq 2.0$.

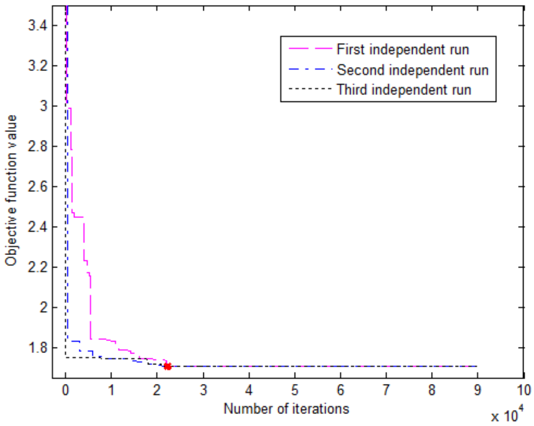

Fig. 4. Evolution of objective function value vs. number of iterations for the welded beam optimal design benchmark problem

Using the values given in the literature for costs, applied load and maximum values of τ, σ, δ, the best solution is: $x_1 = 0.205729631744914, x_2 = 3.347214236041922$, $x_3 = 9.036623910679769, x_4 = 0.205729639976771$. It was found after approximately 21100 objective function evaluations (see Fig. 4) which is less than the ones reported in the open literature [12].

3.3 Speed reducer optimal design

The speed reducer is used in many other types of engineering applications. Its optimal design is a more challenging benchmark engineering problem, due to the fact that it has seven design variables. These variables are: the gear face width (x_1), the teeth module (x_2), the number of pinion teeth (x_3), the length of the first shaft between bearings (x_4), the length of the second shaft between bearings (x_5), the diameter of the first shaft (x_6) and the diameter of the second shaft (x_7). A schematic view of the speed reducer is presented in Fig. 5 together with the seven design variables.

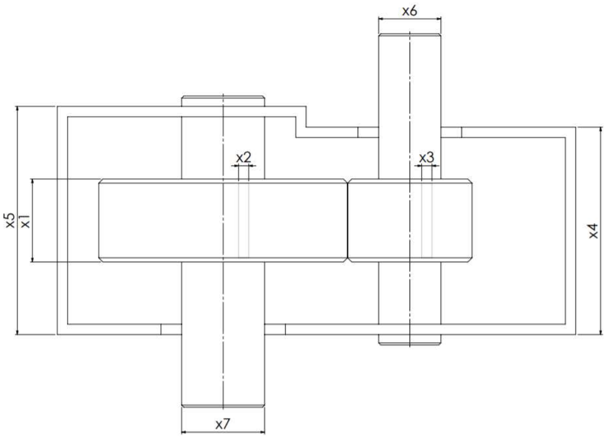

Fig. 5. A schematic view of the speed reducer

This second engineering design benchmark problem consists in minimizing the weight of the speed reducer while respecting the constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shaft. The problem is formulated as follows:

$$\min_{x_1,x_2,\ldots,x_7} 0.7854 x_1 x_2^2 \left(3.3333 x_3^2 + 14.9334 x_3 - 43.0934\right) - 1.508 x_1 \left(x_6^2 + x_7^2\right) + 7.4777 \left(x_6^3 + x_7^3\right) + 0.7854 \left(x_4 x_6^2 + x_5 x_7^2\right) \quad (5)$$

subjected to

$$\begin{aligned}
g_1(x) &= 1 - \frac{27}{x_1 x_2^2 x_3} \geq 0 \\
g_2(x) &= 1 - \frac{397.5}{x_1 x_2^2 x_3^2} \geq 0 \\
g_3(x) &= 1 - \frac{1.93 x_4^3}{x_2 x_3 x_6^4} \geq 0 \\
g_4(x) &= 1 - \frac{1.93 x_5^3}{x_2 x_3 x_7^4} \geq 0 \\
g_5(x) &= 1 - \frac{1}{110 x_6^3} \sqrt{\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16.9 \cdot 10^6} \geq 0 \\
g_6(x) &= 1 - \frac{1}{85 x_7^3} \sqrt{\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 157.5 \cdot 10^6} \geq 0 \\
g_7(x) &= 1 - \frac{x_2 x_3}{40} \geq 0 \\
g_8(x) &= 1 - \frac{5 x_2}{x_1} \geq 0 \\
g_9(x) &= 1 - \frac{x_1}{12 x_2} \geq 0 \\
g_{10}(x) &= 1 - \frac{1.5 x_6 + 1.9}{x_4} \geq 0 \\
g_{11}(x) &= 1 - \frac{1.1 x_7 + 1.9}{x_5} \geq 0
\end{aligned} \quad (6)$$

where x is the vector of the seven design variables and $2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4 \le 8.3$, $7.8 \le x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5.0 \le x_7 \le 5.5$. It has to mentioned here variables are real, except x_3 which is integer.

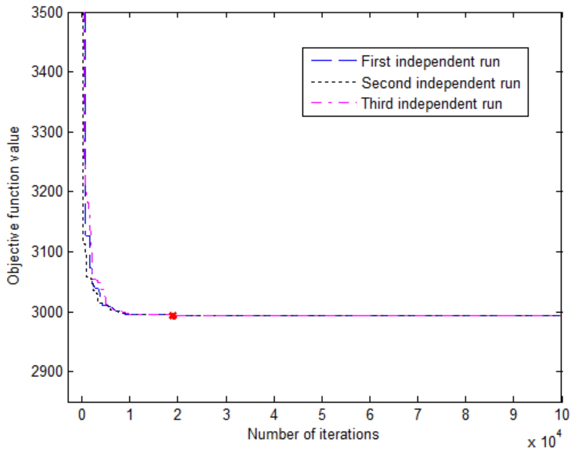

Fig. 6. Evolution of objective function value vs. number of iterations for the speed reducer optimal design benchmark problem

Regardless of the fact that this problem has almost twice as many design variables than the previous one, the best solution was found after less than 20000 objective function evaluations (see Fig. 6) which is less than the ones reported in the open literature [12].

3.4 Four bar truss optimal design

The third design engineering benchmark problem is the one of a four bar truss. The purpose is to determine design with minimum weight of the four bar truss presented in Fig. 7.

The following assumptions were made: members 1 through 3 have the same length l (and the same area) and member 4 has the length $\sqrt{3}l$ (and different area). The constraints concern the stresses in the members and on the vertical displacement at the right end of the truss.

According to [10], the problem can be formulated as follows:

$$\min_{x_1, x_2} 3x_1 + \sqrt{3}x_2 \qquad (7)$$

subjected to

$$g_1(x) = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \ge 0$$
$$g_2(x) = x_1 - 5.73 \ge 0 \qquad (8)$$
$$g_3(x) = x_2 - 7.17 \ge 0$$

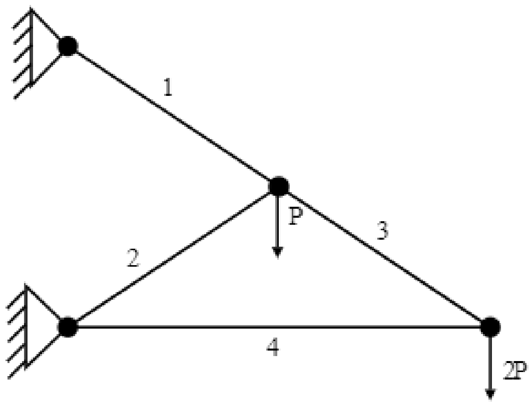

Fig. 7. Four bar statically determinate truss

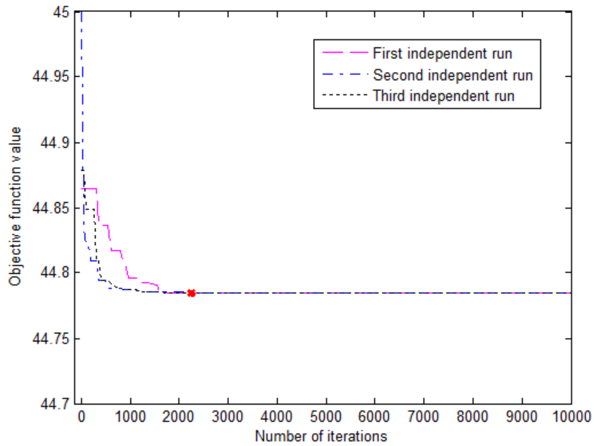

Fig. 8. Evolution of objective function value vs. number of iterations for the four bar truss optimal design benchmark problem

The simplicity of the problem and the modest number of variables is reflected in the small number of objective function evaluations (see Fig. 8) required to find the optimal solution.

4. CONCLUSIONS

The standard Cuckoo search algorithm was enhanced by implementing the Knowledge Gradient policy in Matlab and innovatively use it to evaluate and predict the knowledge of the cuckoo populations. The obtained KGCS algorithm was presented in this paper together with the proof that it performed better when tested on mathematical benchmark functions.

The algorithm was also tested on three engineering design benchmark problems. The results of the performed optimizations showed that KGCS is a fast and reliable optimization algorithm suitable to be used for solving engineering design problems.

5. REFERENCES

[1] Parkinson, A. R., Balling, R. J., Hedengren, J. D., *Optimization methods for engineering design. Applications and Theory*, Brigham Young University, 2013

[2] Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q., *Multiobjective evolutionary algorithms: a survey of the state of the art,* Swarm and Evolutionary Computation, Vol. 1, No. 1, pp. 32-49, 2011

[3] Hassan, S., Kumar, K., Raj, C. D., Sridhar, K., *Design and Optimisation of Pressure Vessel Using Metaheuristic Approach*, Applied Mechanics and Materials, Vols. 465-466, pp. 401-406, 2014

[4] Lin, M.-H., Tsai, J.-F., Hu, N.-Z., Chang, S.-C., *Design Optimization of a Speed Reducer Using Deterministic Techniques*, Mathematical Problems in Engineering, vol. 2013, 7 pages, 2013

[5] Cagnina, L.C., Esquivel, S.C., Coello, C.A.C., *Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer,* Informatica, Vol. 32, pp. 319–326, 2008

[6] Coello Coello, C., Pulido, G., *Multiobjective structural optimization using a microgenetic algorithm*, Structural and Multidisciplinary Optimization, Vol. 30, Issue 5, pp. 388-403, 2005

[7] Yang, X.-S., and Deb, S., *Engineering Optimisation by Cuckoo Search*, Int. J. Mathematical Modelling and Numerical Optimisation, Vol. 1, No. 4, 330–343, 2010

[8] Rusu, M.-F., Optimizations under uncertainty with applications in rolling bearing industry, Faculty of Machine Building, Technical University of Cluj-Napoca, 2014 (PhD. Thesis)

[9] Tudose, L., Rusu, F., Tudose, C., Optimal design under uncertainty of bearing arrangements, Mechanism and Machine Theory, 98, 2016, pp. 164-179

[10] Haftka., R. T., *Approximation and Optimization in Engineering, Chapter 5: Constrained Optimization,* http://www2.mae.ufl.edu/nkim/eas6939/ConstrainedOpt.pdf

[11] De, K., *Optimal design of a welded beam via genetic algorithms*, AIAA Journal, Vol. 29, pp. 2013-2015, 1991

[12] Cagnina, L. C., Esquivel, S. C., *Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer*, Informatica, Vol. 32, pp. 319-326, 2008

ALGORITMUL ÎMBUNĂTĂȚIT KGCS PENTRU PROIECTARE ÎN INGINERIE

Această lucrare prezintă un algoritm de optimizare numit KGCS potrivit pentru a fi folosit la diferite probleme de inginerie. Eficiența acestui algoritm este data de îmbinarea dintre particularitățile algoritmului Cuckoo Search (CS) și ale metodei Knowledge Gradient (KG). Lucrarea prezintă deasemenea testarea agoritmului KGCS pe o serie de probleme de test din inginerie, cum ar fi proiectarea optimală a grinzilor sudate, a reductoarelor de viteză sau a grinzilor cu zăbrele. Rezultatele testelor sunt comparate cu cele raportate în literatură pentru alți algoritmi evolutivi cu scopul de a dovedi performanța ridicată a algoritmului KGCS.

Florina ȘERDEAN, PhD. Math., Lecturer, Technical University of Cluj-Napoca, Department of Mechanical Systems Engineering, Florina.Rusu@omt.utcluj.ro.

Radu MORARIU-GLIGOR, PhD. Eng., Senior Lecturer, Technical University of Cluj-Napoca, Department of Mechanical Systems Engineering, rmogli70@yahoo.com.

Adina CRIȘAN, PhD. Eng., Lecturer, Technical University of Cluj-Napoca, Department of Mechanical Systems Engineering, ducaadina@yahoo.com.