A Dynamic Step-size Adaptation Roach Infestation Algorithm for Constrained Engineering Optimization Problems

Obagbuwa I.C [*]
School of Management, Information Technology and Governance
University of KwaZulu-Natal
Westville, Durban, 4000
Obagbuwa@ukzn.ac.za

Maharaj M.S[†]
School of Management, Information Technology and Governance
University of KwaZulu-Natal
Westville, Durban, 4000
maharajms@ukzn.ac.za

ABSTRACT

Engineering problems belong to a large and complex category of optimization problems with non-linear and non-convex functions; conventional methods are no longer sufficient to handle such problems. Meta-heuristic optimization algorithms have been proved in literature for being able to tackle complex problems. A new meta-heuristic algorithm called dynamic step-size roach infestation optimization algorithm based on searching behaviour of cockroaches was published recently. A Simple Euler method was introduced into a roach infestation optimization algorithm for the enhancement of swarm stability and to allow a balance of exploitation and exploration. The results of the experiments, show its superiority over the existing algorithms. In this work the same method was applied; and modified to solve a constrained engineering problem. The results obtained from simulation processes are close to those obtained by other meta-heuristic methods.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Swarm intelligence and Optimization**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

Swarm Intelligence, Meta-heuristic, Engineering Optimization, Constrained problem

1. INTRODUCTION

Many real world industrial engineering optimization problems are complex, and can be formulated as non-linear programming problems. Traditional solutions are not appropriate for these problems due to complex objective functions and constraints to be satisfied. Constraints are usually very hard to satisfy, but very crucial in engineering optimization design problems. Meta-heuristic algorithms have been proved in literature to provide solutions to such complex problems. Intensification and diversification, or exploitation and exploration are the two major component of a meta-heuristic algorithm [1]. Generating diverse solutions so as to explore the search space on the global scale is referred to as diversification or exploration. While, concentrating on the search in a local region by utilizing the information that a current good solution is found in this region is referred to as intensification or exploitation [1]. The combination of these two components enhances the selection of the best solutions that will converge optimally [1].

Swarm intelligence(SI) algorithms are meta-heuristic algorithms that are population-based, and inspired by social behaviour of insects and animals. Examples of SI algorithms include: particle swarm optimization (PSO), inspired by the birds flocking and fish schooling [2]; ant colony optimization (ACO), inspired by the social behaviour of ants [3]. SI algorithms have been successfully applied to structural design engineering problem in the literature including: PSO was used to solve mixed-integer design optimization problems by He and Whang [4], He et al. [5], Shi and Eberhart [6]. Cagnina et al. [7] applied a simple constraints PSO (SiC-PSO) algorithm to solve constrained engineering optimization problems. Gandomi et al. [8, 9] solved mixed continuous/discrete structural optimization problems using a cuckoo search and Firefly algorithm. Harish Garg applied an artificial bee colony (ABC) algorithm to solve structural engineering problem [10].

Recent emerging techniques in SI paradigm includes Cockroach-based algorithms, inspired by the cockroaches social behaviour. Two recently introduced cockroach algorithms are Cockroach Swarm Optimization (CSO)[11] and Roach Infestation Optimization (RIO) [12] which have been applied to some optimization problems and achieve competitive results when compared to PSO. A dynamic step-size roach infestation optimization (DSARIO) algorithm is a variant of RIO. The simple Euler technique improves DSARIO performance

---

[*] Obagbuwa Ibidun Christiana

[†] Maharaj Manoj

BICT 2015, December 03-05, New York City, United States

DOI 10.4108/eai.3-12-2015.2262602

while searching; cockroach agents adjust their velocity based on their performance [13]. During exploration, if a better region is detected, it will make a local search or make a global search pattern [13]. DSARIO was applied to function optimization problems [13]; and its performance on a constrained problem called the welded beam design problem is shown in this paper. The step-length technique of Euler makes the algorithm effective for constraints handling. It helps to direct the search towards the feasible region where optimal solution can be found.

In Welded beam design optimization, a rigid member is welded onto a beam, and a load applied to the member end [10]. The total production cost is equal to the labour costs, i.e a function of the weld dimensions, in addition to the weld and beam material cost. The optimization of the welded beam design is done to minimize production cost by varying the weld and member dimensions; and the constraints include: limits on the shear stress, bending stress, buckling load and end deflection, and several size constraints [10].

The remaining part of this paper is organised as follows: section 2 describes the DSARIO algorithm, section 3 describes the welded beam constraint engineering optimization problem, section 4 shows DSARIO constraints-handling techniques, section 5 presents experimental settings and the results obtained from our experiments. Conclusions and possible future extensions are given in section 6.

2. DYNAMIC STEP-SIZE ROACH INFESTATION OPTIMIZATION ALGORITHM

Roach infestation optimization (RIO) Algorithm of Haven et al [12] was modified by Obagbuwa et al [13] using the simple Euler method with adaptive step–size h technique to improve swarm stability, convergence capability and maintain balance of exploitation and exploration. DSARIO was described with three components- Find dark, find friend and find food [13]. Cockroaches search for the darkest location in the search space. If a cockroach comes within a detection radius of another cockroach agent, it will stop, socialize and share information of the darkest known location. When a cockroach agent becomes hungry, it searches for food in random food locations [12, 13].

1. Find Dark: Cockroach search for the darkest location in the search space. Find dark model is described as:

   $$v_i = c_0 v_i + c_{max} R_1 (p_i - x_i) \qquad (1)$$

   where v_i is the velocity of ith agent, x_i is the current location found by the ith agent, p_i is the best location found by the ith agent, c_0, c_{max} are cockroach parameter, R_1 is a vector of uniform random numbers, and $(p_i - x_i)$ is a velocity change in the direction of the darkest known location for that agent.

2. Find Friend: If a cockroach comes within a detection radius of another cockroach agent, it will stop, socialize and share information of the darkest known location by setting local location l.

   $$l_i = l_j = arg_k min \left\{ F(p_k) \right\}, k = \{i, j\} \qquad (2)$$

   where i, j are the indices of the two socializing cockroach and p_k is the darkest known location. Find darkest model is extended to include find friend behaviour:

   $$v_i(t+1) = [1-h]v_i(t) - h\psi x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \qquad (3)$$

   $$x_i(t+1) = [1-h]x_i(t) + h(\psi_1 p_i(t) + \psi_2 l_i(t)) \qquad (4)$$

   where v_i denotes cockroach velocity, x_i denotes cockroach position at time t, p_i denotes local best position, l_i denotes global best position, $\psi_1 = c_0.R_1$, $\psi_2 = c_{max}.R_2$, $\psi = \psi_1 + \psi_2$ and h is step–size.

3. Find food: when a cockroach agent becomes hungry, it searches for food and being transported to a random food location b.

   $$x_i = b \qquad (5)$$

For stability of Euler method, Ascher and Petzold [14] recommended a small step size of $(h < 0.2)$ to obtain accuracy per step. In the test of this paper, step-size $h = 0.1$ is considered, the control parameters are taken from Obagbuwa et al [13]. For further information about DSARIO, refer to Obagbuwa et al [13]..

The constrained engineering benchmark problem considered in this paper is described in section 3.

3. WELDED BEAM CONSTRAINED ENGINEERING OPTIMIZATION PROBLEM

The welded beam problem is a practical design problem; a well known benchmark for evaluating optimization algorithms. Figure 1 depicts the welded beam system. The objective of the welded beam design problem is to minimize the cost subject to certain constraints such as shear stress (τ), bending stress in the beam σ, buckling load on the bar (pc), end reflection of the beam δ, and side constraints [10]. There are four design variables in the problem: $h(x_1)$, $l(x_2)$, $t(x_3)$, $b(x_4) = x_1, x_2, x_3, x_4$ [10]
Minimize $f(X) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$

Subject to:
$g_1(X) = \tau(x) - \tau_{max} \leq 0$
$g_2(X) = \sigma(x) - \sigma_{max} \leq 0$
$g_3(X) = x_1 - x_4 \leq 0$
$g_4(X) = 0.1047x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$
$g_5(X) = 0.125 - x_1 \leq 0$
$g_6(X) = \delta(x) - \delta_{max} \leq 0$
$g_7(X) = P - P_c(x) \leq 0$
where:
$\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$
$\tau' = \frac{P}{\sqrt{2x_1x_2}}$
$\tau'' = \frac{MR}{J}$
$M = P(L + \frac{x_2}{2})$
$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2}$
$J = 2\{\frac{x_1x_2}{\sqrt{2}}[\frac{x_2^2}{12} + (\frac{x_1+x_2}{2})^2]\}$
$\sigma(x) = \frac{6PL}{x_4x_3^2}$
$\delta(x) = \frac{4PL^3}{Ex_4x_3^3}$
$P_c = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$
with $P = 6000, L = 14in, \delta_{max} = 0.25in, E = 30{\times}10^6psi, G = 12{\times}10^6psi, \tau_{max} = 1,3600psi, \sigma_{max} = 30000psi, X = (x_1, x_2, x_3, x_4)^T$
$0.1 \leq x_1, x_4 \leq 2.0,\ and\ 0.1 \leq x_2, x_3 \leq 10.0.$

Best solution: $x^* = (0.205730, 3.470489, 9.036624, 0.205729)$ where $f(x^* = 1.724852)$

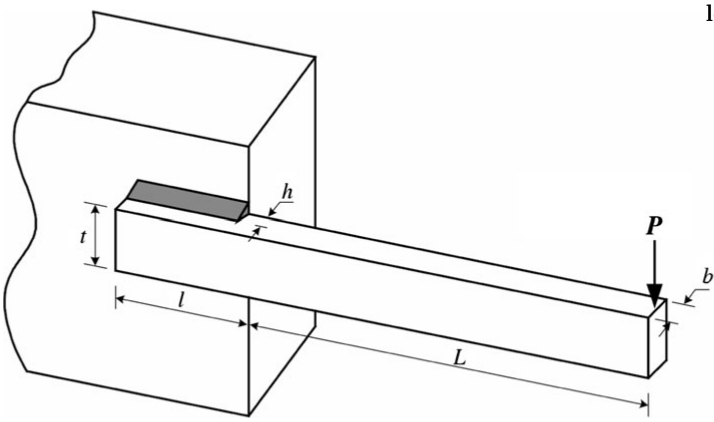

Figure 1: Welded Beam Design Diagram

The constrained DSARIO algorithm for solving the problem described above is shown in section 4.

4. CONSTRAINED DSARIO ALGORITHM

The algorithm in the simplest form, describing the computational steps of constrained DSARIO is shown below:

```
1: Choose cockroach swarm size, Randomly generate the initial cockroach position $x(0)$ and set all parameters with values.
2: Compute the constraints values $g_i(x(0))$ and$h_i(x(0))$
3: Update the cockroach initial personal best position
4: Update the initial global position
5: Enforce the variable bound constraints
6: Compute the analytic constraints $g_i(x(t))$ and $h_i(x(t))$
7: Perform find dark, find friend and find food habits described in section 2
8: Update the cockroach best position
9: Update global best position of cockroach swarm
10: Repeat the loop until stopping criteria is reached.
```

We tested the constrained DSARIO through simulation studies, in section 5, using standard benchmark problems that have been used by many researchers to test optimization algorithms on constrained problems.

5. EXPERIMENTS

The constrained DSARIO algorithm was implemented using MATLAB 7.14 (R2012a), and was simulated on a computer with 2.30 GHz processor with 4.00 GB of RAM. The performance of the algorithm was tested on a known benchmark-welded beam constrained optimization problem.

The control parameters: population size 20, number of iteration 150 were chosen after groundwork experiments; and hunger interval $t_{hunger} = 100$, step-size $h = 0.1$ were adopted from DSARIO [13]

Table 1 show the results of experiments conducted for 20 runs. The results are close to those reported in the literature for related meta-heuristic algorithm on the same problem. The obtained results were compared with the results of ABC [10] on the welded beam design problem. Table 2 show the best performance of the Constrained DSARIO and ABC [10], both algorithms have similar performance. Table 3 depicts the results of the statistical analysis on the obtained results. DSARIO have a very low standard deviation; which indicates that the algorithm is very consistent in solving the problem. The STD in Table 3 denotes standard deviation. The numerical results of ABC on welded beam design problem is as recorded in [10].

Table 1: Simulation Results of DSARIO Welded Beam Constrained Engineering Optimization Problem

Run	X1	X2	X3	X4	F(X)
1	0.2096	3.4769	8.8228	0.2158	1.7699
2	0.2021	3.5505	9.0374	0.2058	1.7303
3	0.1601	4.8253	9.0391	0.2057	1.8208
4	0.2112	3.4002	8.9208	0.2112	1.7451
5	0.1998	3.6465	8.9366	0.2104	1.7569
6	0.1544	5.0369	9.0867	0.0255	1.8428
7	0.2020	3.3298	9.6540	0.2029	1.7831
8	0.1411	5.7316	9.0368	0.2057	1.8910
9	0.1777	4.1983	9.0333	0.2059	1.7749
10	0.1891	3.8736	9.0393	0.2057	1.7522
11	0.1933	3.7607	9.0376	0.2057	1.7440
12	0.1582	4.9993	9.0369	0.2058	1.8263
13	0.2017	3.5592	9.0394	0.2057	1.7309
14	0.1848	3.9857	9.0383	0.2057	1.7593
15	0.3120	2.5272	7.3355	0.3122	2.0930
16	0.2022	3.5486	9.0384	0.2057	1.7302
17	0.2945	2.6388	7.5526	0.2946	2.0338
18	0.2277	3.2193	8.5719	0.2287	1.8081
19	0.1664	4.5797	9.0385	0.02057	1.8023
20	0.2035	3.5165	9.0434	0.2057	1.7287

6. CONCLUSION

This paper presented a constrained DSARIO algorithm, which was applied to solve a popular engineering optimization benchmark called welded beam design constrained problem. The dynamic step-size h of DSARIO allows swarm stability, and maintained a balance of exploitation and exploration within the search domain. This makes the algorithm effective for constraints handling, and also enhances searching by directing it towards the feasible region where an optimal solution can be found. The algorithm showed good results through simulation. Its results were compared with ABC algorithm results; the comparison results show that constrained DSARIO has similar performance with ABC on welded bean design optimization problem. DSARIO algorithm shall be applied to more constrained problems in our further studies.

Acknowledgement

This work was funded by the University of KwaZulu-Natal, South Africa, through a post-doctoral research scholarship.

7. REFERENCES

[1] Xin-She Yang, “Nature-inspired Meta-heuristic Algorithms,” *Second Edition, University of Cambridge, United Kingdom Published by Univer Press, Frome, BA116TT, www.luniver.com*, (2010).

Table 2: Comparison of Best Solution Found for Welded Beam Design Problem.

Algorithm	X1	X2	X3	X4	F(X)
DSARIO	0.2035	3.5165	9.0434	0.2054	1.7287
ABC [10]	0.2057	3.2532	9.0366	0.2057	1.6952

Table 3: Comparison Statistical Result for Welded Beam Design Problem.

Algorithm	Best	Mean	Worst	STD	Median
DSARIO	1.7287	1.8062	2.0930	0.0985	1.7724
ABC [10]	1.6952	1.6953	1.6953	2.8362	1.6953

[2] J. Kennedy and R.C. Eberhart, “Particle Swarm Optimization,”*IEEE Neural Networks Proceedings*, **4**, 1942-1948 (1995)

[3] Marco Dorigo, “Optimization, Learning and Natural Algorithms,” *PhD Thesis, Politecnico di Milano, Italy*, (1992).

[4] Q. He and L. Wang, “An Effective Co-evolutionary Particle Swarm Optimization for Constrained Engineering Design Problems,” *Engineering Applications of Artificial Intelligence*, **20(2007)**, 89-99, (2007).

[5] S. He, E. Prempain and Q. H. Wu, “An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems,” *Engineering Optimization*, **36 (2004)**, 585-605, (2004).

[6] Y. Shi and R. C. Eberhart, “A Modified Particle Swarm Optimizer,” *IEEE International Conference on Evolutionary Computation, Piscataway, NJ: IEEE Press*, 69-73, (1998).

[7] L. C. Cagnina, S. C. Esquivel and C. A. C. Coello, “Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer,” *Informatica*, **32 (2008)**, 319-326, (2008).

[8] A. H. Gandomi, X. S. Yang, and A. H. Alavi, “Mixed Variable Structural Optimization Using Firefly Algorithm,”*Computers and Structures*, **89 (2011a)**, 2325-2336, (2011)

[9] A. H. Gandomi, X. S. Yang and A. Alavi, “Cuckoo Search Algorithm: A Metaheuristic Approach to Solve Structural Optimization Problems,” *Engineering with Computers*, **29 (2003)**, 17-35, (2003)

[10] H. Garg, “Solving Structural Engineering Design Problems using an Artificial Bee Colony Optimization Algorithm,” *Journal of Industrial and Management optimization*, **10(3)**, (2014).

[11] C. ZhaoHui and T. HaiYan, “Cockroach Swarm Optimization” *Computer Engineering and Technology (ICCET) Second International Conference, IEEE*, **6**, (2010).

[12] T. Havens, C. Spain, N. Salmon and J. Keller, ”Roach Infestation Optimization,” *IEEE Swarm Intelligence Symposium*, (2008).

[13] I.C. Obagbuwa, A. O. Adewumi, A. A. Adebiyi, “A Dynamic Step-Size Adaptation Roach Infestation Optimization”, In Proceedings of IEEE International Conference on Advance Computing (IACC 2014), Gurgaon, Indian, (2014).

[14] UM Ascher, L. Petzold, “Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations”*SIAM, Philadelphia*, (1998).