Discrete Multi-Valued Search Space Algorithm based on Cockroach Swarms

Ibidun C. Obagbuwa

Discipline of Information Systems and Technology
School of Management, Information Technology, and Governance
University of KwaZulu-Natal
Westville, Durban, 4000, South Africa.

Abstract - **Many real-life applications today are discrete-valued in nature, and no algorithm has been confirmed to offer solutions to all optimization problems. There is a need for more improved discrete valued algorithms to offer solutions to real life discrete-valued complex optimization problems. A discrete multi-valued search space cockroach swarm optimization algorithm is proposed in this paper which is designed to provide solutions to optimization problems with any discrete range value. The proposed algorithm model was tested with different discrete value ranges using standard benchmarks to prove its performance. The results obtained show that the proposed algorithm would be efficient in solving optimization problems with any discrete value.**

Keywords - Swarm Intelligence, Cockroach swarm, Sigmoid function, Global Optimization.

I. INTRODUCTION

The interest of industry and research community is to develop more improved discrete valued optimization algorithms to solve real life problems which are mainly discrete in nature [1, 2, 3, 4, 5]. Many real world optimization problems today such as signal design, scheduling, and power management have discrete multi-valued variables. Swarm intelligence (SI) techniques are population based algorithms that are inspired by the social behaviour of animal, and one of the recent SI techniques is the cockroach swarm optimization (CSO) that imitate cockroach behaviours [6, 7, 8, 9].

The original CSO was primarily designed to provide solution for optimization problems with continuous values [6, 7].

A number of popular swarm intelligence algorithms have discrete versions that enable them to function in discrete-values spaces; these include particle swarm optimization [1, 2, 3, 4, 5], Harmony search algorithm [10], Cuckoo search algorithm [11], and firefly meta-heuristic algorithm [12]. This paper proposes a discrete multi-valued space cockroach swarm optimization (DCSO) algorithm to handle any discrete range valued optimization problem.

The proposed algorithm can optimize any discrete valued problem without converting the problem into equivalent binary representation. Although, it may be disputed that discrete variable can be converted into an equivalent binary format [2, 3]. For instance a three bit binary representation of range $(0 - 7)$ is needed for a discrete variable range (0,1,2,3,4,5); therefore, special conditions are needed to manage the values past the main range of the discrete variable [2, 3]. In addition, complexity may occur in a search process because the Hamming distance between two discrete values undergoes a linear transformation by the adoption of an equivalent binary representation technique [2, 3]. The binary representation can also increase the dimensions of cockroach. Therefore, the design of a multivalued CSO is essential.

DCSO was designed to work on discrete variable between the range of $[0, D - 1]$, where D denotes discrete values from 2 upward. DCSO searches from one state to another, for instance in binary space, it searches between states (0,1); ternary space (0,1,2); quaternary space (0,1,2,3). Cockroach position is translated into number between $[0, D - 1]$.

Methods on minimization of a set of standard benchmarks that have been widely used by researchers for algorithm performance testing and comparison is investigated in this paper; the benchmarks are described in Table 1. The considered benchmarks characteristics include uni-modal, multi-modal, shifted, rotated, separable, non-separable, scalable, noise in fitness.

The rest of the paper is organized as follows: the proposed DCSO algorithm model is presented in section , the investigation of the ability of the proposed algorithm on standard benchmarks to prove its performance is shown in section , the paper is concluded in section .

II. DISCRETE MULTI-VALUED SPACE COCKROACH SWARM OPTIMIZATION

Some existing meta-heuristics in the literature that were originally developed for continuous optimization problems were transformed to discrete spaces using some methods which include sigmoid function, random-key, smallest position value, modified position equation, great value priority, and nearest integer [13]. Sigmoid function is the most popular among the methods, Kennedy and Eberhart, Veeramacheneni et.al. used sigmoid function to transform continuous PSO to discrete binary and multivalued discrete spaces respectively [1, 2, 3]. Similarly, Sayadia et al, Kwiencien and Filipowicz developed discrete firefly meta-heuristic with the employment of sigmoid function, and solved permutation flow shop scheduling problems [12, 14]. Kwiencien and Filipowicz compared firefly and CSO algorithms for permutation flow shop scheduling problems [14].

Sigmoid function was adopted in this paper to construct a discrete multi-valued space algorithm to cater for optimization problems with any discrete range.

A sigmoid S i = 1+eD−x(i) and (D−1)μ is used to generate solution that is rounded to the nearest discrete variable, and cockroach position is translated to discrete values between [0, D − 1]. The discrete value ranges is between [0, D − 1], where *D* is the discrete value.

$$xi = round((D - 1)\mu + S\ i) \quad 1)$$

Cockroach position is updated as:

$$xi = D - 1,\ 0,\ if\ xifi > xDi <- 01 \quad (2)$$

where xi is the current position, D is the discrete value considered and μ is an operator which enhances the performance of the algorithm. Cockroach positions are now discrete values between [0, D − 1]. There is a possibility of selecting a number between [0, D − 1].

The computational steps of DCSO algorithm is given as follows:

1. Initialize cockroach swarm with uniform distributed random numbers in the range [0, D − 1] and set all parameters with values.
2. Perform cockroach operations (see [6, 7, 9])
3. Limit cockroach agent i magnitude with constant xmax:
   x_i < xmax If xi >
   xmax Then xi =
   xmax
   ElseIf xi < −(xmax) Then xi
   =−(xmax)

4. Update cockroach position using sigmoid function to transform position to discrete values:
   S i = 1+eD−x(i)
   xi = round((D − 1)μ + S i)
   (
   xi = D − 1, 0, if xifi > xDi <− 10

5. Repeat the loop until stopping criteria is reached.

III. SIMULATION STUDIES

This section presents performance of DCSO. The proposed algorithm was implemented in MATLAB 7.14 (R2012a) and run on a computer with 2.30 GHz processor with 4.00 GB of RAM. Methods on minimization of 11 standard benchmarks that have been widely used by researchers for algorithm performance testing and comparison is investigated in this paper; the benchmarks are described in Table 1.

Experiments were conducted in testing the performance of DSCO with different number bases: binary, ternary and quaternary (Bases 2, 3 and 4) respectively. Each number base has length of numerical character string, binary is 2 digits (0,1); ternary is 3 digits (0,1,2); quaternary is 4 digits (0,1,2,3,4). The algorithm can be used for any number based system.

Experimental settings considered in this paper are: 10 dimensions for the benchmarks, swarm size N = 20, xmax = 4, number of runs =25, and 5000 iterations each. Algorithm settings of CSO [6, 7] is used; inertial weight w = 0.618, perception distance visual = 5, the largest step size step = 2. Parameter τ is an operator of DSCO that controls its convergence. The setting of parameter τ is crucial to the performance of the algorithm, low τ values yield better result. τ is set to a value less or equals to 0.2 in our experiments; τ value is stochastically generated, this allows random generation of different values between 0.0 and 0.2 in each iteration. The operator τ effects the convergence of the proposed algorithm, optimal results is obtained in each experiment when τ is set to rand[0.0, 0.2].

The numerical result of DCSO for binary, ternary and quaternary number bases were recorded in this paper after implementing the algorithm; the result is depicted in Table 2. The proposed algorithm has consistence performance in each iteration, this is proved by low standard deviation of the optimal mean in Table 2. The proposed algorithm solves optimally in most cases the benchmark problems for binary, ternary and quaternary discrete values as shown in Table 2. The expected optima value for each benchmark problem is shown in the Table 1 as fbias. The results of our experiments when compare with the expected optima value show that the proposed algorithm solves the benchmark problems optimally, except for Schwefel 2.6 problem for binary and ternary. The algorithm will be efficient for solving optimization problems with any discrete range value.

TABLE 1: BENCHMARK TEST FUNCTIONS.

	Range	f_{bias}	Properties	Functions	Description
F1	[-32, 32]	-140	M,Rt,Sh,N,Sc	Shifted Rotated Ackley	$f_1(x) = -20exp(-0{\cdot}2\sqrt{\sum_{i=1}^{D} \frac{z_i^2}{D}}) - exp(\sum_{i=1}^{D} \frac{cos\, 2\pi z_i}{D})$ $+20 + e + f_{bias}, z = (x-o) * M$
F2	[-5,5]	-330	M,Sh,S,Sc	Shifted Rastrigin	$f_2(x) = \sum_{i=1}^{D}(z_i^2 - 10cos(2\pi z_i) + 10) + f_{bias}, z = (x-o)$
F3	[-5,5]	-330	M,Sh,Rt,N,Sc	Shifted Rotated Rastrigin	$f_3(x) = \sum_{i=1}^{D}(z_i^2 - 10cos(2\pi z_i) + 10) + f_{bias}$ $z = (x-o) * M$
F4	[-0.5,0.5]	90	M,Sh,Rt,N,Sc	Shifted Rotated Weierstrass	$f_4(x) = \sum_{i=1}^{D}(\sum_{k=0}^{kmax}[a^k cos(2\pi b^k(z_i + 0.5))])$ $-D\sum_{k=0}^{kmax}[a^k cos(2\pi b^k 0.5)] + f_{bias}$ $a - 0.5, b = 3, k_{max} = 20, z = (x-o) * M$
F5	[−π,π]	-460	M,Sh,N,Sc	Schwefel 2.13	$f_5(x) = \sum_{i=1}^{D}(A_i - B_i(x))^2 + f_{bias}, A_i = \sum_{j=1}^{D}(a_{ij}\, sin\alpha_j + b_{ij} cos\alpha_j), B_i = \sum_{j=1}^{D}(a_{ij} sinx_j + b_{ij} cosx_j)$ $\alpha = [\alpha_1, \alpha_2, ..., \alpha_D], \alpha_j, [-\pi, \pi]$
F6	[-100,100]	-450	U,Sh,S,Sc	Shifted Sphere	$f_6(x) = \sum_{i=1}^{D} z_i^2 + f_{bias}, z = x - o$
F7	[-100,100]	-450	U,Sh,N,Sc	Shifted Schwefel 1.2	$f_7(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2 + f_{bias}, z = x - o$
F8	[-100,100]	-450	U,Sh,N,Sc,Ns	Shifted Schwefel 1.2 with noise	$f_8(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2 * (1 + 0.4N(0,1)) + f_{bias}$ $z = x - o$
F9	[-100,100]	-310	U,N,Sc	Schwefel 2.6	$f_9(x) = max A\_i\text{-}B\_i + f_{bias}, i = 1,..., D$ $f(x) = max x\_1\text{+}2x\_2\text{-}7, 2x\_1\text{+}x\_2\text{-}5, i = 1,..,n$ A is D*D matrix, $B_i = A_i * o$, o is a D*1 vector
F10	[-100,100]	390	M, Sh, N, Sc	Shifted Rosenbrock	$f_10(x) = \sum^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{bias}$
F11	[0,600]	-180	M, Rt, Sh, N,Sc	Shifted Rotated Griewangk	$z = x - o + 1$ $f_11(x) = \sum_{i=1}^{D}\frac{z_i^2}{4000} - \prod_{i=1}^{D} cos(\frac{z_i}{\sqrt{i}}) + 1 + f_{bias}$ $z = (x-o) * M$

U= Unimodal, M=Multimodal, S=Separable, N=Non-Separable, Sc=Scalable, Sh=Shifted, Rt=Rotated, Ns=Noise in fitness. D=Dimension, Global optimum $x* = 0$, $f(x*) = fbias$. $x = [x1, x2,..., xD]$, $o = [o1,o2,...oD]$, M=D*D orthogonal matrix.

TABLE 2: PERFORMANCE OF DCSO.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
Binary Best	-140	-330	-330	90	-460	-450	-450	-450	-310	390	-180
Average	-140	-330	-330	90	-446.84	-450	-450	-450	294.04	390	-180
STD	0	0	0	0	18.10	0	0	0	532.94	0	0
Ternary Best	-140	-330	-330	90	-460	-450	-450	-450	-310	390	-180
Average	-140	-330	-330	90	-456.17	-450	-450	-450	-5.12	390	-180
STD	0	0	0	0	5.77	0	0	0	346.20	0	0
Quaternary Best	-140	-330	-330	90	-460	-450	-450	-450	-310	390	-180
Average	-139.85	-329.96	-330	90	-454.89	-450	-450	-450	-352.92	390	-180
STD	0.73	0.20	0	0	13.10	0	0	0	512.01	0	0
EOV	-140	-330	-330	90	-460	-450	-450	-450	-310	390	-180

STD denotes standard deviation; EOV denotes expected optimal value include uni-modal, multi-modal, shifted, rotated, separable, non-separable, scalable, noise in fitness. Further details about the benchmark test functions can be found [15].

IV. CONCLUSION

This paper presented a multi-valued discrete space cockroach swarm optimization algorithm that can be used to evaluate optimization problems with any discrete range value. Its performance was tested with binary, ternary and quaternary discrete systems on standard benchmark test functions. The proposed algorithm solves the benchmark problems optimally for binary ternary and quaternary in most cases, and this is an indication that the algorithm would be useful in solving discrete valued optimization problems. Application of proposed algorithm to real life discrete optimization problems, and performance comparison with similar algorithms will be focused upon in further research.

REFERENCES

[1] J. Kennedy and R. Eberhart, “A Discrete Binary Version of the Particle Swarm Algorithm”, IEEE International Conference on System, Man, and Cybernetics, (1997).

[2] L. Osadciw and K. Veeramacheneni, “Particle swarm optimization for continuous, binary and discrete search spaces”, Particle swarm optimization, edited by Aleksander Lazinica, published by In-tech, printed in Croatia, 451-460, (2009).

[3] K. Veeramacheneni, L. Osadciw and G. Kamath, “Probabilistically Driven Particle Swarms for Optimization of Multi-valued Discrete Problems: Design and Analysis”, In IEEE swarm intelligence symposium, 141-149 (2007).

[4] J. Pugh and A. Martinoli, “Discrete Multi-valued Particle Swarm Optimization”, IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, (2006).

[5] B. Al-Kazemi and C. K. Mohan, “Multi-Phase Discrete Particle Swarm Optimization”, Proceedings of the Fourth International Workshop on Frontiers in Evolutionary Algorithms, (2002).

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

[7] C. ZhaoHui, “ A modified Cockroach Swarm Optimization”, SciVerse ScienceDirect, Elsevier Ltd, (2011).

[8] I.C. Obagbuwa, A. O. Adewumi and A. A. Adebiyi, “Stochastic Constriction Cockroach Swarm Optimization for Multidimensional Space Function Problems”, Mathematical Problems in Engineering, 2014, (2014).

[9] I.C Obagbuwa and A.O. Adewumi, “An Improved Cockroach Swarm Optimization”, The Scientific World Journal, 2014, (2014).

[10] W. Ling, Y. Xu, Y. Mao and M. Fei, “A Discrete Harmony Search Algorithm”, Life System Modelling and Intelligent Computing, Kang Li, Xin Li, Shiwei Ma George W. Irwin, (Eds.), Springer Berlin Heidelberg, 98, 37-43, (2010).

[11] A. Ouaarab, B. Ahiod, XS Yang, “Discrete Cuckoo Search Algorithm for the Travelling Salesman Problem”, Neural Computing and Applications, 1;24(7-8), 1659-1669, (2014).

[12] M. K. Sayadia, R Ramezaniana and N. Ghaffari-Nasaba, “A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems”, International Journal of Industrial Engineering Computations, 1 (2010), (2010).

[13] J. Krause, J. Cordeiro, R.S. Parpinelli, and H.S. Lopes, “A Survey of Swarm Algorithms Applied to Discrete Optimization”, In Swarm Intelligence and Bio-inspired Computation: Theory and Applications. Elsevier Science and Technology Books; Elsevier Inc.: Amsterdam, The Netherlands, pp. 169-191, (2013).

[14] J. Kwiencien and B. Filipowicz,“Comparison of firefly and cockroach algorithms in selected discrete and combinatorial problems”, Bulletin of the Polish Academy of Sciences Technical sciences, 62(4), (2014).

[15] P.N. Suganthan et al., “Problem definitions and evaluation criteria for the CEC 2005, special session on realparameter optimization,” Technical report, Nanyang technology university, Singapore and KanGAL report number 2005005(Kanpur genetic algorithms laboratory, IIT Kanpur), (2005).