Swarm Intelligence Optimization Algorithms: A Review

Julius Beneoluchi Odili[1], Mohd Nizam Mohmad Kahar[1] and A. Noraziah[1, 2]
[1]*Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang, Kuantan 26300, Malaysia.*
[2]*IBM Centre of Excellence, Universiti Malaysia Pahang, Kuantan 26300, Malaysia.*
odili_julest@yahoo.com

***Abstract*— Swarm Intelligence, of late, has gradually become an exciting area of research interest to many researchers in science and engineering. The primary reason for this interest is because swarm intelligence exploits the miraculous cum harmonious working of nature in ensuring order, preservation, conservation, longevity and sustenance of plants and animals in the ecosystem. As a result, researchers that believe that mimicking nature is key to solving diverse problems in engineering, technology and science has developed a number of swarm intelligence techniques. This paper presents a review of some recently developed swarm intelligence algorithms that have been successfully applied to solve a number of optimization problems with special emphasis on their application areas, strengths and observable weaknesses. This study aims to assist researchers in their choice of algorithm to solve optimization problems.**

***Index Terms*—Swarm Intelligence; Optimization Problems; Algorithms; Application Areas; Algorithm's Strengths and Weaknesses.**

I. Introduction

Swarm Intelligence is simply a reference to a set of mobile agents that through direct or indirect communications among themselves collectively solve some fundamental problems that hitherto would have been impossible if the agents are working individually. Some experts [1-3] define Swarm Intelligence as a discipline that deals with natural and artificial systems of a group of individuals who organize themselves through decentralization using very basic rules that lead to the emergence of intelligent behavior unknown to single agents. Examples of algorithms in this class include Cuckoo Search (CS) [4], African Buffalo Optimization (ABO) [5], Teaching Learning Based Optimization [6], Bat Algorithm [7], Jaya Algorithms [8] etc.

In Computer Science, Swarm Intelligence is used to refer to a set of techniques or methods that derive their inspiration from the collective social behavior cum interactions of organisms such as ants, animals, plants and other elements in our ecosystem that harness the collective intelligence of the entire swarm/herd towards solving some difficult problems that would have been difficult or even impossible for a single agent. [9]. The features of Swarm-based techniques are:

- They are usually population-based. They use multi agents in their search processes
- The agents that make up the population are usually homogenous
- The general attitude of the system results from each individual interacting with one another and with the environment
- The individual agents are always mobile, moving in a haphazard manner
- Their actions, including movements are in response to the environment
- There is no centralized control structure, leaders only emerge according to their performances in each iteration [10].

Some of the most popular Swarm Intelligence techniques are the Genetic Algorithm, Ant Colony Optimization, Particle Swarm Optimization, Bee Colony Optimization and the Firefly Algorithm etc. Swarm-based techniques that were developed in the last few years are getting increasingly popular. Examples of these emerging techniques include Cuckoo Search, Jaya Algorithm, Teaching Learning Based Optimization and Bat Algorithm etc. Some common areas of application of the above-listed swarm intelligence techniques are the Travelling Salesman's Problem (TSP) [11], numerical function optimization [12], vehicle routing [13], job scheduling [14], tuning of PID Parameters in Automatic Voltage Regulators [15] etc.

In view of the popularity of swarm intelligence algorithms coupled with the subsequent development of a number of them in recent times, researchers are confronted with the problem of choice whenever they require a swarm intelligence technique to solve a given problem. This problem is further given impetus by the No Free Lunch Theorem which states that no particular algorithm is capable of solving all types of problems optimally. That is to say that algorithms' performances are rather specialized: that a particular algorithm obtained optimal solution in a particular problem is not a guarantee that it will do the same in other problems. It is, therefore, necessary to examine algorithms' performances in order to guide researchers in making informed choices.

This rest of this paper is organized in the following way: Section II discusses the Cuckoo Search; Section III examines Bat Algorithm; Section IV, the Teaching Learning Based Optimization; Section V, Jaya Algorithm and Section VI draws conclusion on the study.

II. Cuckoo Search (CS)

The CS which is a simulation of the subtle attitude of the cuckoo bird was designed by Yang and Deb [16]. The cuckoo lays her eggs in the nests of other birds, sometimes of other species with the hope that those other birds will execute the maternal function of incubating those eggs. Whenever the host birds discover the cuckoo's prank, it either abandons the nest or throw such strange eggs away. Otherwise, it goes ahead to incubate the eggs. The cuckoo bird, on its own, perfects the act of subtlety by imitating the egg of its host in order to perpetuate its kind by such fraud. In CS, the hosts' eggs in a nest represents an optimization solution while the cuckoo egg is a representation of a newer solution with the objective of using the newer solution to replace the existing one.

In CS, it is assumed that a cuckoo bird lays an egg at a time in any nest chosen randomly. Next, the nest with the best quality/number of eggs carries on to the next generation. Moreover, there are a fixed number of nests and the cuckoo egg is discovered by the host bird with a given probability, usually between 0 and 1. The CS pseudocode [17] is presented in Figure 1.

```
Begin
Initialize the algorithm parameters
Determine initial population of nests
While termination not reached
    Randomly generate a cuckoo by Levy flight:
    X_ij(t + 1) = X_ij(t)+ α ⊕ Levy (λ)
    Evaluate the cuckoo fitness
    Randomly select a nest among available host nests
    If f_i>f_k) replace k with the new outcome
    End if
    Abandon a fraction of worse nests and replace
    with new nests
    Retain the best outcomes
    Rank the best outcomes and obtain the current best
End while
Output the best result
End
```

Figure 1: Cuckoo Search pseudocode

The CS though relatively young has enjoyed wide applications in different technical areas such as travelling salesman problems, document clustering, wireless sensor networks, speaker recognition, flood forecasting, shortest path in distributed system, image processing, classification task in health sector, job scheduling etc. with competitive outcomes [18].

The CS is quite an effective algorithm obtaining results where other algorithms struggle [18, 19]. In spite of its seeming wide applications, the Cuckoo search has the problem of speed due to its use of several parameters, and sometimes, falling into local optima [19] hence the emergence of several varieties of the algorithms such as discrete CS, Improved CS etc. [20].

III. Bat Algorithm (BA)

The BA which is inspired by the echolocation attitudes of micro-bats that employ different pulse rates of loudness and emission was designed by Xin-She Yang in 2010. In this algorithm, individual artificial bats fly randomly employing a velocity vi at and at position (solution) xi, at a given dynamic frequency (wavelength) and loudness Ai. As the search process progresses, the micro-bats find a prey and changes its frequency, pulse emission r and loudness Ai. The local search component of the search is done via a random walk. At each iteration, the algorithm selects the position of the best performing bat as its solution until the stopping condition is reached. The pseudocode of the BA [21] is presented in Figure 2.

```
Begin
State Objective function
Initialize the population of bats, velocity Vi and position Xi
Determine the pulse frequency fI at position Xi
Define the pulse rates rl and the loudness Ai
While (not termination) do
    Obtain new outcomes by adjusting frequency,
    updating velocities and positions using the
    frequency, velocity, position equations
    respectively:
    f_i = f_min +(f_max − f_min)β
    v_i(t + 1) = v_i(t) +(x_i(t)-x*) f_i
    x_i(t + 1)= x_i(t)+ v_i(t + 1)
    If (rand > rl)
    Choose an outcome among the best outcomes
    Select a local outcome (position) around the
    selected best position (outcome)
    End if
    Select another outcome by getting the micro-bats
    to fly randomly
    If (rand < Ai and f(Xi) < f(x*))
        Accept the new outcomes
        Increase the value of rl and reduce Ai
    End if
End while
Rank the bats and determine the current best outcome X*
Output the position of the best micro-bat
End
```

Figure 2: Pseudocode of Bat Algorithm

So far, the BA has been very successful since its design and has enjoyed a considerable application such as in combinatorial optimization problems, parameter estimation and classifications, image processing, data mining [22] etc.

BA's effectiveness is traceable to its simplicity, flexibility and straightforward implementation strategy. Moreover, the algorithm provides quick convergence at the initial stages through switching from exploration to exploitation [23]. The pulse emission and loudness components of the algorithm helps the algorithm to control and zoom into new regions (exploration). However, there is inherent danger in allowing BA to switch to exploitation stage too early by varying rl and Ai too fast: the algorithm settles into premature convergence. Moreover, the BA speed is bogged down by several parameters since it appears to be a simplification of the PSO. Invariably, it inherits the basic parameters of the PSO in addition to some of its own algorithm-specific parameters [7].

IV. Teaching Learning Based Optimization (TLBO)

TLBO is a recently developed population-based optimization algorithm that searches for solutions through each learner's deliberate effort to attain the knowledge height of the teacher [24, 25]. The algorithm approximates the societal view of teachers as the most knowledgeable individuals in the society. In TLBO, therefore, the teacher represents the optimum solution and a group of learners constitutes the population. The design variables in TLBO are the different subjects being offered to the learners and the learners' results represent the fitness. The search process of TLBO is divided into two: the first, being "Teacher phase" and the next: "Learner phase". In the "Teacher phase" the learners learn from the teacher but during the "Learner phase", learners interact among themselves. Therefore, a learner with the minimum objective function value is deemed a Teacher for the subsequent iteration. The algorithm obtains good results through a careful interplay of the Teacher and Learner phases. The pseudocode of the TLBO [26] is presented in Figure 3.

Begin
Initialize random population of learners X and evaluate all learners X
While (Until stopping criteria) do
For $i = population\ size$, do
/*Teacher Phase (Exploration)*/
Select $X_{teacher}$ and calculate X_{mean}
T_F= round (1+rand (0,1))
$X_i(t+1) = X_i(t) + X_{teacher} - T_F X_{mean}$
If $X_i(t+1) > f(X_i(t))$, $X_i(t) = X_i(t+1)$
End if
/* Learners Phase (Exploitation)*/
Randomly select one learner $X_j(t)$ from the population X such that $i \neq j$
If $f(X_i(t)) < f(X_j(t))$, $X_i(t+1) = X_i(t) + r(X_j(t) - X_i(t))$
Else $X_i(t+1) = X_i(t) + r(X_i(t) - X_j(t))$
End if
If $f(X_j(t)) < f(X_i(t))$, $X_i(t) = X_i(t+1)$
End if
End for
End while
Output the best result
End

Figure 3: TLBO pseudocode

The TLBO has been applied to solve the travelling salesman's problems, constrained and unconstrained global optimization test functions, economic dispatch, robotics, mechanical engineering, motif discovery problems, structural engineering [27], vehicle routing, data clustering [28], structural engineering, parameter optimization and etc. [29] with good results.

TLBO has proven to be efficient and has enjoyed wide application since its development [30-32]. The algorithm has input capacities that ensure wide exploration of the search space [33]. However, in spite of the huge acceptance of the TLBO, TLBO has been found to be good at exploration but can perform very poorly in exploitation resulting in premature convergence in complex problems [34, 35].

Another major concern with the use of TLBO is the issue of speed. It is believed that the speed of TLBO is a major shortcoming. To address this, Satapathy and Naik developed what they termed modified TLBO. Their effort majorly changed the learner phase equation thus inculcating extra tutorials [36]. Another effort to address the speed problem of TLBO was by Rao and Patel in what they called Improved TLBO [37].

V. Jaya Algorithm (JA)

The JA which was developed by Rao is in furtherance of the new paradigm in algorithm development, namely, parameter-less algorithms following the successful design of the Teaching Learning Based Optimization (TLBO), a pioneer parameter-less approach to optimization. JA operates on the premise that solutions are obtainable for any problem moving towards the best results and deliberately, avoiding the worst solutions. Unlike other optimization algorithms, JA requires few control parameters such as population size, maximum number of generations and number of design variables. JA does not have need of any problem-specific parameters that requires to be tuned in order to get appropriate results. The pseudocode of Jaya Algorithm is presented in Figure 4.

Begin
Initialize population size, number of design variables and termination criterion
Identify best and worst solutions in the population
Modify the solutions based on the best and worst solutions:
$X'j,k,i = Xj,k,i + r1,j,i\ (Xj,best,i - \ldots$
$|Xj,k,i|) - r2,j,i\ (Xj,worst,i - |Xj,k,i|$
If the solution, $X'j,k,i < Xj,k,i$ **then**
Accept and replace the previous solution
Else
Keep the previous solution
End if
Repeat until no more improvement
Output best solution
End

Figure 4: Jaya Algorithm pseudocode

The Jaya is actually a simple to implement algorithm and has been successfully applied to solve benchmark global optimization functions, travelling salesman's problem as well as power flow problems [38].

The biggest strength of the JA is that it is parameter-less. This single attribute makes it user-friendly because a user does not have to bother with parameter-tuning in order to obtain acceptable outcomes. Although JA is parameter-less the tuning of the algorithm-specific controlling parameters is not as easy as it looks. For instance, controlling the parameters in each iteration was found to be very difficult and time consuming [39]. Moreover, because it is a recently developed algorithm, its application is not widespread. It remains to be seen how the algorithm will perform in diverse search landscapes [40].

VI. Findings and Conclusions

This paper examined the four recently-developed algorithms, otherwise called the 21st century algorithms and these are the CS, BA, TLBO and Jaya Algorithm. The choice of these algorithms are borne out of the fact that aside they all being swarm intelligence techniques and are less than a decade old, they have been proven to be quite successful. Moreover, while two of the algorithms: CS and TLBO are parameterized algorithms, the other two: TLBO and Jaya Algorithm are parameter-less. That is to say that TLBO and Jaya Algorithm do not require special tuning of parameters to solve optimization problems.

From the foregoing discussion, it is obvious that the parameter-less techniques have the advantage of user-friendliness as non-experts can easily solve scientific and engineering optimization problems without having to bother about rigorous and time-consuming parameter-tuning. However, it has been observed that obtaining the appropriate controlling parameters for Jaya, in particular, is still a matter of concern to many researchers. This possibly explains why the TLBO appears to be more popular.

In terms of the parameterized algorithms, the BA has a slight edge over the CS because the BA is famous for its simplicity, flexibility and straightforward implementation strategy. In addition, the BA's quick convergence at the initial stages through switching from exploration to exploitation further recommends the algorithm [23]. In conclusion, however, the findings from this study further agrees with the No Free Lunch Theorem that asserts that whatever is of interest to a researcher determines his choice of a swarm intelligence technique.

ACKNOWLEDGMENT

The authors are grateful to the Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang for funding under Grant GRS 1403118 and to FRGS for additional funding under Grant RDU 140101.

REFERENCES

[1] S. Binitha and S. S. Sathya, "A survey of bio inspired optimization algorithms," *International Journal of Soft Computing and Engineering,* vol. 2, pp. 137-151, 2012.
[2] J. Kennedy, J. F. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm intelligence*: Morgan Kaufmann, 2001.
[3] R. A. Mahale and S. Chavan, "A survey: evolutionary and swarm based bio-inspired optimization algorithms," *International Journal of Scientific and Research Publications,* vol. 2, pp. 1-6, 2012.
[4] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with computers,* vol. 29, pp. 17-35, 2013.
[5] J. B. Odili and M. N. M. Kahar, "African buffalo optimization (ABO): a new meta-heuristic algorithm," *Journal of Advanced & Applied Sciences,* vol. 03, pp. 101-106, , 2015.
[6] R. Rao and V. Kalyankar, "Parameters optimization of advanced machining processes using TLBO algorithm," *EPPM, Singapore,* vol. 20, pp. 21-31, 2011.
[7] X.-S. Yang and X. He, "Bat algorithm: literature review and applications," *International Journal of Bio-Inspired Computation,* vol. 5, pp. 141-149, 2013.
[8] R. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations,* vol. 7, pp. 19-34, 2016.
[9] V. Pandiri and A. Singh, "Swarm intelligence approaches for multidepot salesmen problems with load balancing," *Applied Intelligence,* pp. 1-13, 2015.
[10] R. S. Parpinelli and H. S. Lopes, "New inspirations in swarm intelligence: a survey," *International Journal of Bio-Inspired Computation,* vol. 3, pp. 1-16, 2011.
[11] J. B. Odili, "Application of ant colony optimization to solving the traveling salesman's problem," *Science Journal of Electrical & Electronic Engineering,* vol. 2013, 2013.
[12] J. B. Odili and M. N. M. Kahar, "Numerical function optimization solutions using the african buffalo optimization algorithm (ABO)," *British Journal of Mathematics & Computer Science,* vol. 10, pp. 1-12, 2015.
[13] S. Erdoğan and E. Miller-Hooks, "A green vehicle routing problem," *Transportation research part E: logistics and transportation review,* vol. 48, pp. 100-114, 2012.
[14] S. Ren, Y. He, and F. Xu, "Provably-efficient job scheduling for energy and fairness in geographically distributed data centers," in *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, 2012, pp. 22-31.
[15] J. B. Odili and M. N. Mohmad Kahar, "African buffalo optimization approach to the design of PID controller in automatic voltage regulator system," *National Conference for Postgraduate Research, Universiti Malaysia Pahang,* vol. September, 2016, pp. 641-648, 2016.
[16] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 210-214.
[17] S. Agrawal, R. Panda, S. Bhuyan, and B. K. Panigrahi, "Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm," *Swarm and Evolutionary Computation,* vol. 11, pp. 16-30, 2013.
[18] S. Kamat and A. Karegowda, "A brief survey on cuckoo search applications," *Int. J. Innovative Res. Comput. Commun. Eng,* vol. 2, 2014.
[19] A. Ouaarab, B. Ahiod, and X.-S. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem," *Neural Computing and Applications,* vol. 24, pp. 1659-1669, 2014.
[20] M. Marichelvam, T. Prabaharan, and X.-S. Yang, "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan," *Applied Soft Computing,* vol. 19, pp. 93-101, 2014.
[21] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO 2010),* ed: Springer, 2010, pp. 65-74.
[22] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu, *Swarm intelligence and bio-inspired computation: theory and applications*: Newnes, 2013.
[23] I. Fister, S. Rauter, X.-S. Yang, and K. Ljubič, "Planning the sports training sessions with the bat algorithm," *Neurocomputing,* vol. 149, pp. 993-1002, 2015.
[24] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design,* vol. 43, pp. 303-315, 2011.
[25] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences,* vol. 183, pp. 1-15, 2012.
[26] K. Z. Zamli, F. Din, S. Baharom, and B. S. Ahmed, "Fuzzy adaptive teaching learning-based optimization strategy for the problem of generating mixed strength t-way test suites," *Engineering Applications of Artificial Intelligence,* vol. 59, pp. 35-50, 2017.
[27] S. Degertekin and M. Hayalioglu, "Sizing truss structures using teaching-learning-based optimization," *Computers & Structures,* vol. 119, pp. 177-188, 2013.
[28] S. C. Satapathy and A. Naik, "Data clustering based on teaching-learning-based optimization," in *International Conference on Swarm, Evolutionary, and Memetic Computing*, 2011, pp. 148-156.
[29] R. Rao, "Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems," *Decision Science Letters,* vol. 5, pp. 1-30, 2016.
[30] R. V. Rao and V. Patel, "Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm," *Applied Mathematical Modelling,* vol. 37, pp. 1147-1162, 2013.
[31] A. Shabanpour-Haghighi, A. R. Seifi, and T. Niknam, "A modified teaching–learning based optimization for multi-objective optimal power flow problem," *Energy Conversion and Management,* vol. 77, pp. 597-607, 2014.
[32] M. Singh, B. Panigrahi, and A. Abhyankar, "Optimal coordination of directional over-current relays using Teaching Learning-Based Optimization (TLBO) algorithm," *International Journal of Electrical Power & Energy Systems,* vol. 50, pp. 33-41, 2013.
[33] K. Yu, X. Wang, and Z. Wang, "An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems," *Journal of Intelligent Manufacturing,* pp. 1-13, 2014.
[34] M.-Y. Cheng and D. Prayogo, "Fuzzy adaptive teaching–learning-based optimization for global numerical optimization," *Neural Computing and Applications,* pp. 1-19, 2016.
[35] F. Ge, L. Hong, and L. Shi, "An autonomous teaching-learning based optimization algorithm for single objective global optimization," *International Journal of Computational Intelligence Systems,* vol. 9, pp. 506-524, 2016.
[36] S. C. Satapathy and A. Naik, "Modified teaching–learning-based optimization algorithm for global numerical optimization—A comparative study," *Swarm and Evolutionary Computation,* vol. 16, pp. 28-37, 2014.
[37] R. V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica,* vol. 20, pp. 710-720, 2013.
[38] W. Warid, H. Hizam, N. Mariun, and N. I. Abdul-Wahab, "Optimal power flow using the Jaya algorithm," *Energies,* vol. 9, p. 678, 2016.
[39] H. M. Pandey, "Jaya a novel optimization algorithm: What, how and why?," in *Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference*, 2016, pp. 728-730.
[40] R. Venkata Rao and G. Waghmare, "A new optimization algorithm for solving complex constrained design optimization problems," *Engineering Optimization*, pp. 1-24, 2016.