

# Jaya a Novel Optimization Algorithm: What, How and Why?

Hari Mohan Pandey

Department of Computer Science and Engineering  
Amity University Uttar Pradesh, Noida, India  
profharimohanpandey@gmail.com, hmpandey@amity.edu

**Abstract**— This paper presents a comprehensive discussion of the Jaya algorithm, a novel approach for the optimization. There exist two broad categories of heuristic algorithms are: evolutionary algorithms and swarm intelligence. These algorithms' performance vastly depends on the parameters used need extensive tuning during the computational experiments to achieve the superior performance. The Jaya algorithm is a new optimization algorithm has been proposed recently is parameter less and therefore parameters tuning is not needed for it. The primary aim of this paper is to discuss the Jaya algorithm based on the rational aspects outlined as: (a) what is Jaya algorithm; (b) How it works and (c) why one should use it. The author believes that this discussion might be useful to explore the potential of the Jaya to the general audience working for the optimization.

**Keywords**— Evolutionary Algorithm, Jaya Algorithm, Swarm Intelligence, Nature Inspired Algorithm, Optimization

## I. INTRODUCTION

Jaya, a novel algorithm has been illustrated in this paper considering the three key rational aspects are: what is the Jaya algorithm, how it works and why to use it. There exist several different types of population based heuristic algorithms have been proposed, were classified into two broad categories are: evolutionary algorithms (EAs) and swarm intelligence (SI) algorithms. The EA and SI have been successfully applied to solve many typical real life problems were most of the time difficult to solve employing traditional search algorithms. The genetic algorithm (GA), evolutionary strategy (ES), evolutionary programming (EP), genetic programming (GP), differential evolution (DE), bacteria foraging optimization (BFO), artificial immune algorithm (AIA) are few belongs to the EA, whilst particle swarm optimization (PSO), ant colony optimization (ACO), artificial bee colony (ABC), firefly (FF), shuffled frog leaping (SFL) algorithms fall into the SI. There exist some other algorithms are based on the nature, such as: Harmony Search (HS), Lion Search (LS), Gravitational Search (GS), Grenade Explosion Method (GEM), and Biogeography-Based Optimization (BBO) and others.

There is no question about the exploration capabilities, these algorithms holds, has been successfully applied to solve several different types of search and optimization problems. The successes of these algorithms are greatly based on the parameters they used basically guides the searching and are majorly contributing in the exploration and exploitation of the search space. For example, the GA uses crossover operator is

mainly an exploration operator, whilst the mutation operator has been considered as exploitation operator. These GA operators work based on the specific probabilities defined in the beginning of the search. It has been observed that if the probabilities of the crossover and mutation operator has been tuned appropriately the GA's search get trapped in a local optimum. This situation is referred as premature convergence, a situation when the diversity of the population is decreased leads to unavoidable local optimum convergence. Pandey et al. [1] has discussed a lot about this and presented a comprehensive review of the approaches have been proposed to address the premature convergence. Like GA's crossover and mutation there are other factors also presents are population size, maximum number of generations, elite size, inertia weight (in PSO), acceleration rate (in PSO), onlooker bees, employed bees, scout bees (in ABC), harmony memory (in HS), number of improvisation (in HS), pitch adjusting rate (in HS). These parameters play a significant role in the success of the algorithms: GA, ES, EP, GP, DE, BFO, AIA, PSO, HS, ACO, ABC and others. The controlling and proper tuning of the algorithm specific parameter is a crucial factor, shows a greater impact in the success in terms of finding the global optimum.

Several literatures exist reports the advocacy of controlling the algorithm specific parameters and how these parameters help in balancing the exploration and exploitation. There exist several approaches that have been attempted to balance the exploration and exploitation employing fine tuning are: trial and error (consume more time and is performed in ad-hoc manner), following general guidelines (use recommended parameter setting from the existing literatures), use parameter less algorithms (robust, but less efficient), use experience from the similar applications (not fit in a situation, where experience do not exist), use of mathematical models (sometimes simple but most of the times have been found difficult), etc. Additionally, some literatures have reported the importance of orthogonal array approach, Taguchi approach and full factorial approach in identifying the tuned value of the parameters. The above discussion shows a clear indication and importance of the tuning and controlling of the parameters, which is undoubtedly a tedious and time consuming process. The lack of knowledge in conducting and preparing the correct computational experimental setup makes the whole process even more complex.

Considering the above discussed fact, a new approach was proposed known as *Teaching Learning Based Optimization* (TLBO). The TLBO algorithm does not need any algorithm

specific parameters; rather it needs only common controlling parameters (population size and maximum number of generations) to function. The TLBO algorithm works in two phases' are: *teacher phase* and *learner phase*.

This paper is focused towards the Jaya algorithm, a novel approach to the optimization. The Jaya algorithm is very similar to the TLBO, as it falls in the category of algorithm specific parameter less algorithm. The Jaya algorithm is new, is proposed by R. Venkata Rao in 2015 [4], has been discussed in this paper in a significant depth, considering the following rational aspects: *what is Jaya algorithm, how it works, and why to use it*.

The rest of the paper is organized as follows: Section II has reported the Jaya algorithm and its working, whilst the importance of the Jaya algorithm over the other algorithms has been discussed in Section III. The concluding remarks and future assessment have been drawn in Section IV.

## II. THE JAYA ALGORITHM AND ITS WORKING

The Jaya is a simple and powerful global optimization algorithm has been successfully applied to the benchmark function of constraint and unconstrained problems. All though it is parameter less algorithm as the TLBO is, however, it is different since it does not require learner phase, i.e. it uses only one phase is teacher phase, whereas the TLBO performs its action in two phases. It is based on the fact that the solution can be obtained for a given problem moving towards the best solution, avoid the worst solution. It is the beauty of this algorithm that it requires only few control parameters like maximum number of generations and population size, and number of design variables, most of the time common for any algorithms. It does not need any algorithm specific control parameters require extensive tuning before conducting the actual computational experiments. The work procedure of the Jaya algorithm is outlined in Algorithm-1 is simple to understand.

---

### Algorithm-1: Jaya Algorithm

---

S1 Initialize  
 $PS \leftarrow Population\_Size$   
 $NDV \leftarrow Number\_of\_Design\_Variables$   
 $TER\_COD \leftarrow Termination\_Condition$   
S2 Until the termination condition, not satisfied Repeat S3 to S5  
S3 Evaluate the best and worst solution  
Set  $best \leftarrow Best\_Solution\_Population$   
Set  $worst \leftarrow Worst\_Solution\_Population$   
S4 Modify the solution  
 $X'_{j,k,i} = X_{j,k,i} + r_{1,j,i} (X_{j,best,i} - |X_{j,k,i}|) - r_{2,j,i} (X_{j,worst,i} - |X_{j,k,i}|)$   
S5 If  $(X'_{j,k,i} > X_{j,k,i})$  Then  
    Update the previous solution  
Else  
    No update in the previous solution  
S6 Display the optimum result

---

It is must discuss the basic terminologies used in the Algorithm-1. The Jaya algorithm starts initializing the basic parameters such as: Population size ( $PS$ ), Number of design variables ( $NDV$ ), and Termination condition ( $TER\_COD$ ) (most of the time maximum number of iterations are considered as the termination condition). The primary objective is to minimize/maximize an objective function  $f(x)$ . Let us assume a situation, where at  $i^{th}$  iteration there are  $m$  design variables (i.e.  $j = 1, 2, \dots, m$ ) and  $n$  number of candidate solution exist (i.e.  $k = 1, 2, \dots, n$ ). Suppose that  $best$  indicate the best candidate solution obtained for the  $f(x)$ , is represented as  $f(x)_{best}$  from the whole candidate solution. Similarly, if  $worst$  represent the worst solution obtained in the entire worst candidate solution for the  $f(x)$ , is shown as  $f(x)_{worst}$ . The term  $X_{j,k,i}$  represents the  $j^{th}$  design variable for the  $k^{th}$  candidate at  $i^{th}$  iteration. Let  $X'_{j,k,i}$  is the modification done in  $X_{j,k,i}$ , is calculated via S4 of the Algorithm-1. The terms  $X_{j,best,i}$  and  $X_{j,worst,i}$  respectively represent the value of  $j$  variable for the best and worst candidate. Two random numbers  $r_{1,j,i}$  and  $r_{2,j,i}$  in the range  $[0,1]$  have been used, represents the  $j^{th}$  variable at  $i^{th}$  iteration. The expression  $r_{1,j,i} (X_{j,best,i} - |X_{j,k,i}|)$  and  $-r_{2,j,i} (X_{j,worst,i} - |X_{j,k,i}|)$  respectively shows that the search is moving towards the best solution and avoids the worst solution. If  $(X'_{j,k,i} > X_{j,k,i})$  is true, then update the previous solution otherwise no (S4). A loop has been implemented at S2 of Algorithm-1 repeats S3 to S5 maintains a list of all the accepted values for  $f(x)$ , is used as an input for the  $(i+1)^{th}$  iteration.

The S4 has clearly represented that the Jaya algorithm has the tendency to move to the best, i.e. closer to the success and avoids the worst solution obtained in the iteration. This nature makes this algorithm victorious, hence the name Jaya is defined, is derived from a Sanskrit word meaning "victory".

## III. WHY JAVA ALGORITHM?

This section illustrates the importance of the Jaya algorithm over the other algorithms. The detailed discussion presented in the Section I and II has concluded three main reasons are outlined here:

- The Jaya algorithm does need any algorithm specific parameters, which requires extensive tuning before conducting the actual computational experiments, if not done correctly, leads unavoidable and unwanted convergence.
- The controlling of the algorithm specific parameters is not as easy as it looks. Also, controlling the parameters in each

iteration is most of the time difficult and time consuming. This entire process does not belong to the Jaya algorithm.

- c) The most important reason is the victorious nature of the Jaya algorithm makes it more powerful as compare to any other algorithms.

#### IV. CONCLUSIONS

This research successfully meets what it aimed for. The author has strongly presented that the algorithms either belongs to EAs or SI or nature based needs algorithm specific parameters greatly contribute to the success demands extensive tuning before the computational experiments. This process is time consuming and most of the times were found difficult to achieve by the beginner or intermediate level researchers. Therefore, they miss the actual melody of these algorithms. In addition, the author has discussed the importance of the TLBO algorithm for the optimization, works in two different phases: teacher phase and learner phase.

In this paper, the author has discussed a new optimization algorithm has been presented recently known as “Jaya algorithm”, has shown the victorious nature, does not require any algorithm specific parameters and delivers the optimized result in a time effective manner. Previous research conducted has compared the Jaya algorithm with several other algorithms shows the computational capabilities, the discussed algorithm holds. It is not the aim of this paper is to prove Jaya algorithm is the “best” algorithm over the algorithms discussed in the literatures. In fact, no such the “best” algorithm exists, it all depends on the problem type and the environment where the problem is solved. Rather, the aim is to increase the awareness of the Jaya algorithm among the researchers working for optimization.

So far, the Jaya has shown the computational capability when applied to the constraints and unconstrained problem, but not applied to solve any real life problems. Therefore, it would be interesting to see the impact of the Jaya algorithm to solve the real world optimization problems. The author strongly believes that this paper will give an edge to the Jaya algorithm, which will lead to an increase the use of it in the near future.

#### REFERENCES

- [1] H.M. Pandey, A. Chaudhary, and D. Mehrotra. "A comparative review of approaches to prevent premature convergence in GA." *Applied Soft Computing* 24 (2014): 1047-1077.
- [2] Črepinšek, Matej, Shih-Hsi Liu, and Marjan Mernik. "Exploration and exploitation in evolutionary algorithms: a survey." *ACM Computing Surveys (CSUR)* 45.3 (2013): 35.
- [3] H.M. Pandey, A. Chaudhary, and D. Mehrotra. "Grammar induction using bit masking oriented genetic algorithm and comparative analysis." *Applied Soft Computing* 38 (2016): 453-468.
- [4] Venkata Rao Jaya Algorithm. Rao, R. "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems." *International Journal of Industrial Engineering Computations* 7.1 (2016): 19-34.