IOP Conf. Series: Journal of Physics: Conf. Series **1455** (2020) 012007 doi:10.1088/1742-6596/1455/1/012007

Best-Worst-Play (BWP): A metaphor-less Optimization Algorithm

Ramanpreet Singh[a*], Kumar Gaurav[a], Vimal Kumar Pathak[a], Prem Singh[b], Himanshu Chaudhary[c]

[a]Department of Mechanical Engineering, Manipal University Jaipur, Jaipur, 303007, India
[b]Department of Mechanical Engineering Swami Keshvanand Institute of Technology Management & Gramothan, Jaipur, 302017, India
[c]Department of Mechanical Engineering, Malaviya National Institute of Technology Jaipur, Jaipur 302017, India

Email: ramanpreet.singh@jaipur.manipal.edu

Abstract : A novel algorithm which is an ensemble of two metaphor-less algorithms is presented in this paper. The algorithm is inspired by Rao-1 and Jaya algorithms. Since the algorithm always plays around with the best and worst solutions; the algorithm is named as Best-Worst-Play (BWP) algorithm. The algorithm does not require any algorithm specific parameters, however, algorithm control parameters are required. To test the effectiveness and performance of the proposed algorithm, a number of unconstrained and constrained benchmark functions are considered. It is found that proposed algorithm has outperformed several well-established metaphor based algorithms. The proposed BWP algorithm may be used by researchers to solve the unconstrained and constrained optimization problems
Keywords: Jaya, Rao-1; Best-Worst Play; BWP; Optimization algorithm; hybrid;

1. Introduction

The meta-heuristic optimization algorithms are categorized as: evolutionary, physics-based, swarm-based, and human-based algorithms. For an instance, Genetic Algorithm (GA), Evolution Strategy (ES), Genetic Programming (GP), Biogeography based Optimizer (BBO), etc., these algorithms are inspired by the phenomenon of natural evolution, and hence they come under evolutionary algorithms. Likewise, the category that is inspired from the social behaviour of animals comes under nature-inspired or swarm intelligence algorithms [1,6, 11]. Some of the examples include: Particle Swarm Optimization (PSO), Ant-Colony Optimization (ACO), Marriage in Honey Bees Optimization (MBO), Monkey Search (MS), Firefly Algorithm, etc. The algorithms such as Big-Bang Big-Crunch (BBBC), Simulated Annealing (SA), Black Hole (BH), etc., fall in the realm of physics-based algorithms. The fourth category of the optimization algorithms is human-based algorithms which include algorithms like teaching-learning based optimization (TLBO), Harmony Search (HS), Tabu Search (TS), League Championship Algorithm etc. [1, 2, 4].

These algorithms are population-based meta-heuristics which shares a common tendency. Due to this tendency, the algorithm is portioned as exploration and exploitation phases [3]. Exploration phases allows the random disturbance of design variables to the highest possible extent. This phase is followed by the exploitation phase in which the most promising areas are identified. To avoid entrapping in a local optima, a right balance between exploration and exploitation is needed. Because of the stochastic nature of these algorithms, it becomes a challenging task [5, 10].

In the realm of meta-heuristic population-based optimization algorithms, the algorithms are based on the swarm behaviour, physics laws, or natural phenomenon. Therefore, these are also known as metaphor-based algorithms. Lately, many researchers are proposing new algorithms and they all prove that their algorithm outperform other algorithms. Many of those algorithms are not used for the research while some gained some popularity. Nevertheless, the researchers may focus on developing simpler and metaphor-less algorithms. Therefore, in this paper, a new metaphor-less algorithm named as "Best-Worst-Play (BWP)" algorithm is proposed. The algorithm makes use of two popular metaphor-less algorithms, namely, Jaya, and Rao-1. Due to their hybridization, the algorithms contains two operators, besides, the algorithm has a right balance between exploration and exploitation.

2. Proposed Algorithm

In this section, BWP algorithms is proposed. The algorithm makes use of the Jaya and Rao-1 operators. Therefore, a little explanation for Jaya and Rao-1 is as follows:

2.1. Jaya Algorithm

Jaya word belongs to Sanskrit language meaning 'victory.' The algorithm has an operator that brings the candidate solution closer to the best solution and avoids the worst solution. The operator for performing this function is as follows:

$$d_{ji,iter}^{t+1} = d_{ji}^{t} + rand_1\left(d_{can_{best}}^{t} - \left|d_{ji}^{t}\right|\right) - rand_2\left(d_{can_{worst}}^{t} - \left|d_{ji}^{t}\right|\right) \quad (1)$$

in which, d, stands for the design variables for $i = 1, \ldots, n$. $n$is the total number of design variables, $j = 1, \ldots, p$, where p is the total number of population size (or number of candidate solutions), $t$represents the t^{th} list of candidates, $iter$ represents iteration number, can_{best} represents the best candidate solution, and can_{worst} represents the worst candidate solution. The pseudo code for implementation of Jaya algorithm is as follows:

Table 1. An example of a table.

	Jaya Algorithm
1	*Initialize controlling parameters such as:* *Population size, number of iterations, number of design variables, etc.*
2	*Search for the Candidate Best and Candidate Worst solution in the entire population*
3	*Using Eq. (1) update the candidates*
4	*If the candidate solution corresponding to* $\boldsymbol{d^t_{ji}}^{+1}$*is better in comparison with* $\boldsymbol{d^t_{ji}}$
5	*If Yes: then update the candidate by replacing*
6	*Else: Keep the previous candidate*
7	*Is the number of generations/ termination criterion satisfied?*
8	*If Yes: Record the candidate solution as optimum; Else GoTo Step 2*

2.2. Rao-1 Algorithm

Rao-1 is a metaphor-less algorithm, which is entirely dependent on the difference between the $\boldsymbol{can_{best}}$ and $\boldsymbol{can_{worst}}$. Variables which are considered for Rao-1 have same meaning as for Jaya algorithm. The algorithm uses only one operator which is as follows:

$$d_{ji,iter}^{t+1} = d_{ji}^{t} + rand\left(d_{can_{best}}^{t} - d_{can_{worst}}^{t}\right) \quad (2)$$

The pseudo code for implementation of Rao-1 algorithm is same as used for Jaya algorithm; except the fact that Eq. (2) is used in Step 3 instead of Eq. (1).

2.3. Best-Worst-Play (BWP) Algorithm

The BWP algorithm has a right balance of exploration and exploitation. It merges two metaphor-less algorithms to fully utilizing the best-worst candidate solutions for updating the candidates. The algorithm uses two operators as used in Jaya and Rao-1 which are as follows:

$$d_{ji,iter}^{t+1} = d_{ji}^{t} + rand_1\left(d_{can_{best}}^{t} - \left|d_{ji}^{t}\right|\right) - rand_2\left(d_{can_{worst}}^{t} - \left|d_{ji}^{t}\right|\right) \quad (3)$$

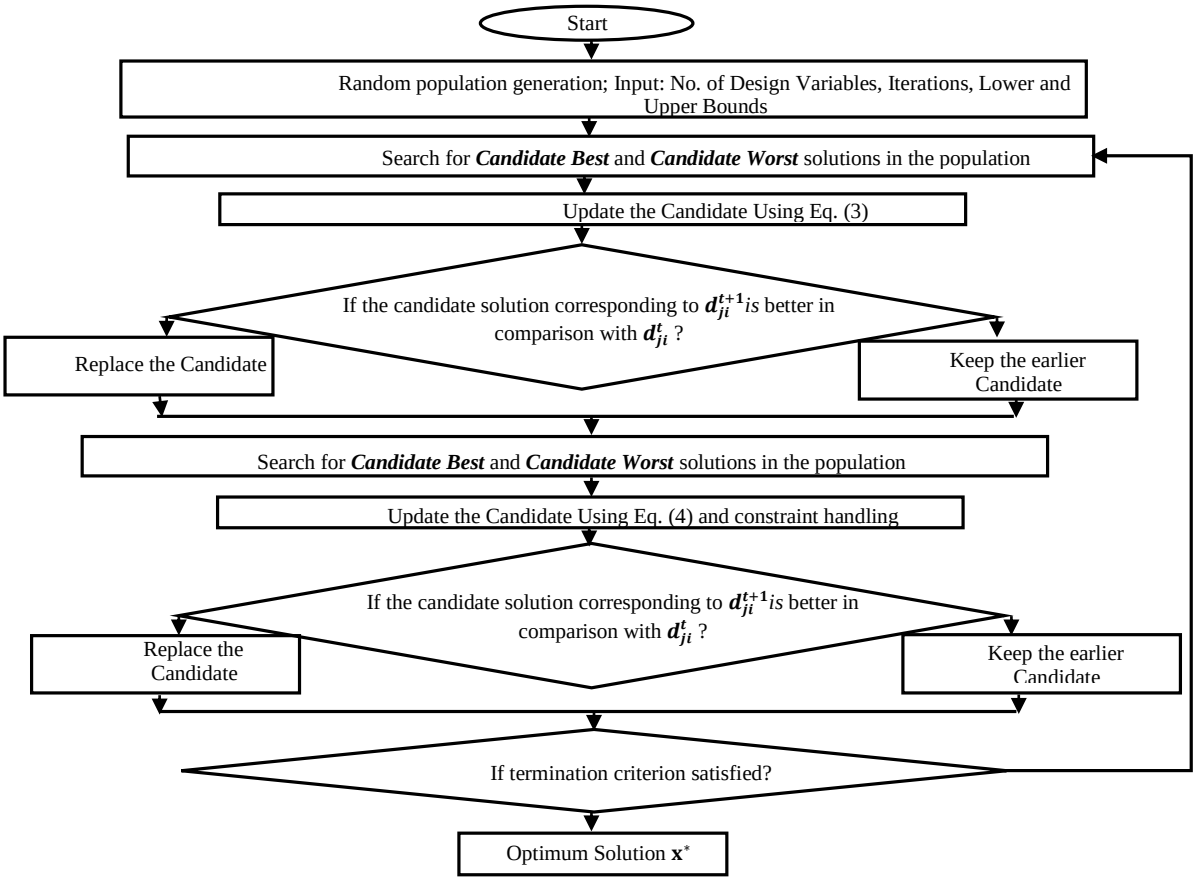
$$d_{ji,iter}^{t+1} = d_{ji}^{t} + rand\left(d_{can_{best}}^{t} - \left|d_{can_{worst}}^{t}\right|\right) \quad (4)$$

in which, the variables have the same definitions as used here for Jaya and Rao-1 algorithms. The schematic diagram for the BWP algorithm is shown in Fig. 1. The algorithm begins with the initialization of the population randomly followed by prescribing the number of design variables,

maximum number of generations, lower bounds, and upper bounds. From the generated population, the algorithm search for the *candidate best* and *candidate worst* solutions. Then, entire population of candidates is updates using the Eq. (3). This will create an updated list of candidates. The updated list is compared with the previous list of candidates for creating a new list of candidates through *Greedy Selection* (exploitation phase). Thereafter, the algorithm identifies the *candidate best* and *candidate worst* solutions to updating the list of candidates using Eq. (4). Eventually, a final list is created through greedy selection which completes first iteration of BWP algorithm.

3. Computational experiments for unconstrained and constrained optimization problem

In this section, the BWP algorithm is applied on 5 unconstrained and 5 constrained benchmark functions. The unconstrained functions are shown in Table VI, in which *Dn* represent the dimensions of the function, *Limits* indicate the lower and upper limits for the design variables in the search space, and



$f(\mathbf{x}^*)$, indicates the optimum value for objective function.

Table 2. .Unconstrained benchmark objective function

Sr. No.	Benchmark Function	*Dn*	*Limits*	$f(\mathbf{x}^*)$
1	$\sum_{i=1}^{n} x_i^2$	30	[-100 100]	0
2	$\sum_{i=1}^{n-1}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$	30	[-30 30]	0
3	$\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10 10]	0
4	$\sum_{i=1}^{n}[ix_i^4+random[0,1]]$	30	[-1.28 1.28]	0

To demonstrate the effectiveness of the proposed BWP algorithm, 5 unimodal unconstrained benchmark functions (in Table VI) and 5 constrained functions are considered. The algorithm is run 20 times for

each benchmark function for different number of function evaluations for each function. The results of the BWP algorithm are compared with other metaphor-less and nature inspired algorithms which are shown in Table VII. It is found that the BWP algorithm is competitive to other well established algorithms such as GWO, Rao-(1-3), etc.

Table 3. .Unconstrained benchmark objective function

Benchmark Function		**Rao-1**	**Rao-2**	**Rao-3**	**GWO [6]**	**BWP**	**Benchmark Function**		**Rao-1**	**Rao-2**	**Rao-3**	**GWO [6]**	**BWP**
1	Best	4.84E25	1.40E15	1.58E50		2.095E35	3	Best	2.04E-15	0.00012	6.32E24		0.573
	Mean	3.59E22	3.57E12	6.71E42	6.59E28	**1.156E32**		Mean	4.07E-12	0.678	9.33E21	7.18E17	1.4679
	SD	7.33E22	7.95E12	1.56E41	6.34E5	0		SD	1.40E-11	2.53	3.84E20	0.029	1.1160
	Function Evaluations					12000		Function Evaluations					30000
2	Best	0.4038	0.00287	0.00648		4.74E-04	4	Best	25.868	68.121	29.88		33.0428
	Mean	31.60	11.474	29.206	26.812	**0.0203**		Mean	87.013	148.94	84.122	0.310	110.09
	SD	28.40	16.683	29.093	69.90	0.0358		SD	32.317	41.526	38.179	47.356	49.988
	Function Evaluations					54000		Function Evaluations					30000

4. Conclusions

The paper presented a novel metaphor-less algorithm, BWP, which is based on the notion of best and worst candidate solutions. The algorithm is a careful ensemble of two metaphor-less algorithms, namely, Jaya and Rao-1 algorithms. The algorithm's effectiveness is tested by considering 5 unconstrained benchmark functions. It is found that BWP algorithms converges faster than the contemporary and other nature-inspired algorithms. For an instance, the algorithm reported more than 90% of reduction in the number of functions evaluations required for 1st constrained function. Also, more than 66%, and 83% reductions in the required number of function evaluations are reported for benchmark functions 2 and 3, respectively. However, the algorithm took more than 20% of the number of function evaluations for convergence. The algorithm is proved to be competitive with other natureinspired algorithms. It may also be tried for several realistic applications.

References

[1] S. Mirjalili, . and A. Lewis,"The whale optimization algorithm. Advances in engineering software," vol 95, pp.51-67, 2016.

[2] R.V. Rao, V.J. Savsani, and D.P. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems." Computer-Aided Design, vol 43(3), pp.303-315, 2011.

[3] R. Singh, H. Chaudhary,and A.K. Singh, "A new hybrid teaching–learning particle swarm optimization algorithm for synthesis of linkages to generate path." Sādhanā, vol42(11), pp.18511870, 2017.

[4] R. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems." International Journal of Industrial Engineering Computations, vol 7(1), pp.19-34, 2016.

[5] R. Singh, H. Chaudhary, and A.K. Singh, "a novel gait-inspired four-bar lower limb exoskeleton to guide the walking movement. "Journal of Mechanics in Medicine and Biology", p.1950020, 2019. [6] S. Mirjalili, S. M. Mirjalili, A. Lewis. "Grey wolf optimizer." Advances in engineering software. Mar 1;69:pp 46-61, 2014.

[7] R.V. Rao, V.J. Savasani."Mechanical design optimization using advanced optimization techniques." Springer Science & Business Media, London, 2012

[8] R.V. Rao, and V. Patel,"An elitist teaching-learning optimization algorithm for solving complex constrained optimization problems." International Journal of Industrial Engineering Computations. 3 (2012) 535-560

[9] R.V. Rao, "Teaching Learning Based Optimization Algorithm and its Engineering Applications," Springer, 2015

[10] V.K. Pathak, A.K. Singh, R. Singh,and H. Chaudhary,"A modified algorithm of Particle Swarm Optimization for form error evaluation." tm-Technisches Messen. Apr 20;84(4):272-92, 2017

[11] N. Jain, G.K. Badhotiya, A.S. Chauhan and J.K. Purohit,"Reliability-Based Design Optimization Using Evolutionary Algorithm." InAmbient Communications and Computer Systems 2018 (pp. 393-402). Springer, Singapore.