

Adaptive Wind Driven Optimization Algorithm

Zikri Bayraktar
Schlumberger-Doll Research Center
One Hampshire Street
Cambridge, MA 02139
zikribayraktar@gmail.com

Muge Komurcu
Earth Systems Research Center
University of New Hampshire
8 College Road, Morse Hall
UNH, Durham, NH 03824
muge.komurcu@unh.edu

ABSTRACT

In this paper, we propose two new methods to create an adaptive Wind Driven Optimization (WDO) algorithm, both of which are shown to outperform the classical WDO method while eliminating the need for fine-tuning the coefficients of the update equations. While the classical WDO offers a simple and efficient metaheuristic optimization algorithm, the coefficients that are inherent to the workings of the algorithm create an undesired level of complexity especially for the novice users. To alleviate this complexity and automate the coefficient selection, two adaptive Wind Driven Optimization (AWDO) methods are proposed in this paper. First method is to replace the fixed values of the coefficients with randomly generated numbers from a uniform distribution at each iteration and the second method is to optimize the selection of the coefficients with the Covariance Matrix Adaptation Evolution Strategy (CMAES). To evaluate the performance of the proposed methods for AWDO, four well-known numerical benchmark functions from the literature are utilized and results are compared against the classical WDO. Both of the new methods outperform the classical WDO while the AWDO using CMAES performs the best among of all.

Categories and Subject Descriptors

• Computing methodologies

General Terms

Algorithms.

Keywords

Wind driven optimization, adaptive wind driven optimization, covariance matrix adaptation evolutionary strategy, metaheuristic.

1. INTRODUCTION

Metaheuristic algorithms constitute one of the many branches of the stochastic optimization algorithms, which do not depend on derivatives and perform as well as global optimization methods. Compared to deterministic methods, metaheuristics tend to be slower but they do not require prior information on the search

space. The workings of these algorithms are either inspired from nature or based on statistics and probability.

Some of the nature inspired algorithms are Genetic Algorithms (GA) [1], Particle Swarm Optimization (PSO) [2], Ant Colony optimization [3], Artificial Immune System algorithms [4], Differential Evolution (DE) [5] and more recently the Wind Driven Optimization (WDO) method [6]. The WDO, is inspired from the motion of wind in the atmosphere and provides a simple and efficient global optimization method. The algorithm inherently employs multiple variables, which allows the user to tune it if the problem at hand requires fine-tuning. On the other hand, such flexibility also leads to a challenge for the novice user on picking the suitable values for these variables since most users utilize these optimization algorithms as black box solvers for their applications. To alleviate this issue, we are proposing two new methods to adaptively select the values of the variables inside the WDO so that the users should not worry about the algorithm performance and can easily use the AWDO as a black box solver.

Section 2 briefly introduces the workings of the WDO and the update equations along with the tuning parameters. Section 3 provides a short overview of the Covariance Matrix Adaptation Evolutionary Strategy and Section 4 presents the adaptive WDO methods. Section 5 illustrates numerical benchmark results of the proposed methods and Section 6 presents the conclusions.

2. WIND DRIVEN OPTIMIZATION (WDO)

The Wind Driven Optimization algorithm, which was introduced in [7], is a nature inspired population based iterative heuristic global optimization method and can be implemented in any field and application where GA, PSO or any type of evolutionary strategy are utilized [8,9]. It was inspired by the physical equations describing the trajectory of an individual air parcel under the influence of various natural forces in our atmosphere in hydrostatic balance. In this high level of abstraction of wind description, the horizontal movement of air is stronger than the vertical movement hence equations are derived accordingly where certain level of simplifications are made to achieve computational efficiency in an N -dimensional optimization problem.

Atmospheric motion can be represented by the Eulerian description of an assumed infinitesimally small air parcel where its motion follows the Newton's second law of motion. Then, we can compute the velocity and position of the air parcel within the N -dimensional search space. The position and the velocity of the air parcel are iteratively updated in the search space until it converges to an optimum point or maximum number of iterations is reached. A detailed description of the algorithm and the

parameter study can be found in [6] and [7], hence we will only present the velocity and position update rules below. The velocity update equation is expressed as,

$$\vec{u}_{new} = (1-\alpha)\vec{u}_{cur} - g(\vec{x}_{cur}) + \left|1 - \frac{1}{i}\right| RT(x_{max} - x_{cur}) + \frac{c}{i} u_{cur}^{otherdim} \quad (1)$$

where i represents the rank of the air parcel among all population members based on the pressure value at its location in the search space. Other terms of the velocity update equation include α as the friction coefficient, g as the gravitational constant, R as the universal gas constant, T as the temperature and c as a constant that represents the rotation of the Earth. Each of these terms is set to a constant value and do not change over time. As seen in equation (1), the updated velocity of each air parcel for the next iteration (u_{new}) is computed for each dimension and is affected by its velocity at the current iteration (u_{cur}), its current location in the search space (x_{cur}), its distance from the highest pressure point that has ever been found (x_{max}) as well as a velocity at one of the other dimensions ($u_{cur}^{otherdim}$). The pressure in the WDO is analogous to the cost function in general, which is used to determine how well each air parcel satisfies the desired design goals. Low pressure indicates a good solution and high pressure indicates a bad solution. The position of each air parcel is limited to hard boundaries of $[-1, 1]$ by the search algorithm before linearly rescaling to the design limits, and the parcel velocity per iteration is limited to $V_{max} = \pm [0.3]$. Once the velocity of the parcel is updated by equation (1), then the position is updated by,

$$\vec{x}_{new} = \vec{x}_{cur} + (\vec{u}_{new} \times \Delta t) \quad (2)$$

where x_{new} represents the updated position for each air parcel for the next iteration. For all iterations, a unity time step, $\Delta t = 1$, is assumed. At each iteration, the velocity and the position of each parcel are updated based on their pressure values and parcels move from high pressure points to lower pressure point just like the wind movement in the atmosphere. One can terminate the iterations by either setting the maximum number of iterations or simply by exiting if a predefined pressure value is achieved.

3. COVARIANCE MATRIX ADAPTIVE EVOLUTIONARY STRATEGY (CMAES)

As one of the many evolutionary algorithms, the Covariance Matrix Adaptation Evolution Strategy (CMAES) is a population based iterative global optimization method for continuous space optimization problems [10,11]. One of the unique properties of the CMAES is that it does not require parameter tuning, which is taken care of internally without needing input from the user other than the population size. This makes it a suitable candidate for our application, since the WDO defines the population size and CMAES only optimizes the four terms inherently important for the WDO.

The CMAES starts with a population that is sampled from a distribution in the search space with a standard deviation of one-third of the parameter range. Then, new populations are formed from the modified distributions over each iteration which are reshaped Gaussian distributions. The distribution can take a hyperellipse shape in any direction since it is defined in part by a covariance matrix. User defined cost function is used to evaluate the population performance and they are ranked based on the cost function. Then, the covariance matrix is updated and the shape of

the distribution is determined. Finally, the step size is updated and the iterations are repeated until the desired cost is achieved. Further details of the algorithm can be found in [10,11] and freely available sample codes on the Internet, hence will not be repeated here.

4. ADAPTIVE WIND DRIVEN OPTIMIZATION (AWDO)

The velocity update equation of the WDO algorithm requires preselected values for four different terms, namely α , g , c , and RT that are constant throughout the iterations. These terms allow users to tune the algorithm based on the problem at hand, but it can also lead to challenges for the novice users who treat the WDO algorithm as a black-box solver. In such a case, setting these terms properly becomes a burden on the user and can lead to inefficiency if they are not selected correctly. In [6], we performed numerical studies to identify the most suitable values for these terms and obtained a range of different combinations of values that could work for various optimization problems.

To alleviate the burden of choosing the right values for these terms, we propose two new methods in this work. The first method is to randomize the values of these terms at each iteration with a value selected from a uniform distribution in the range of $[0, 1]$. Similar approach is taken in [12], where the values of these four terms are replaced with new terms utilizing a heavy tailed Levy distribution (WDOLE). However, their approach simply replaces the original four terms in WDO with new four terms that need to be manually set for the Levy distribution, which does not alleviate the original problem. From [6], we can conclude that the values of the four WDO terms can be simply picked from the $[0, 1]$ range and using a simple uniform distribution, WDO can perform better than the heavy tailed, more complicated distributions as we will demonstrate in the next section with numerical benchmark functions.

While we propose to choose the value of the terms randomly in the first approach, a better way would be to optimize these values as the iterations progress. To this end, we propose a second method, where we utilize the CMAES algorithm to optimize the values of these four terms at each iteration and suggest a new set of values for the next iteration. The population size for the CMAES would be the same size as the WDO population size and the dimensions would be limited to four since CMAES would only work on these four terms. Other internal parameters of the CMAES are internally selected by CMAES by design, which makes this implementation of WDO a parameter free adaptive optimization algorithm. The flowchart of the AWDO is shown in Figure 1. It starts with the initialization of the algorithms, such as randomly generating position and velocity. Then, pressure at the location of each population member is evaluated and velocity along with the position is updated for the next iteration. At this point, the pressure and values for the four terms, α , g , c , and RT , are passed on to the CMAES algorithm. CMAES then selects a new set of values for the four terms based on the pressure and returns the control to WDO. This iterative process of updating the position and velocity by WDO and optimizing four terms by CMAES continues until the maximum number of iterations is exhausted.

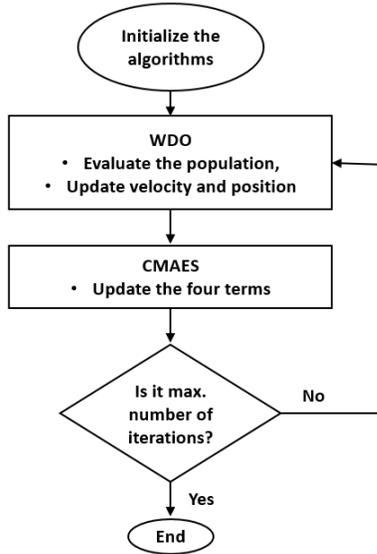


Figure 1. Flowchart of the adaptive WDO algorithm.

5. COMPUTATIONAL RESULTS

The performance of the AWDO is tested and compared against the classical WDO, utilizing four well-known numerical benchmark functions: *Sphere*, *Rastrigin*, *Griewank* and *Rosenbrock*. The *Sphere* function is a separable unimodal function, where as *Rastrigin* and *Griewank* are multimodal. *Rosenbrock* is a non-separable, non-convex quadratic function.

Each algorithm ran for maximum of 3000 iterations with a population size of 100 members. The problem dimensions were limited to $N=10$, with different upper and lower bounds per function as shown in the tables, below. We ran each function 50 times and the best, worst, mean and standard deviations are presented in Tables 2-5.

Table 1 illustrates the average number of iterations that each algorithm took to reach convergence. Since the maximum number of iterations is limited to 3000, neither of the algorithms reached the global optimum for *Rosenbrock* while for *Sphere*, *Rastrigin* and *Griewank*, AWDO needed the least number of iterations to find the global optimum.

Table 1. Average number of iterations for convergence

	Sphere	Rastrigin	Griewank	Rosenbrock
cWDO	445	2367	3000	3000
WDOu	105.14	23.72	63.68	3000
AWDO	35	18.14	28.04	3000

On the other hand Tables 2-5, illustrate the best, mean and worst pressure values for classical WDO (cWDO), and newly proposed WDO with uniform distribution for the terms (WDOu) and adaptive WDO with CMAES (AWDO) optimization achieved at the maximum number of iterations over the 50 runs.

Table 2. Results of Sphere in search space of $[-100, 100]^{N=10}$

	cWDO	WDO with uniform	AWDO
Best	0	0	0
Mean	0	0	0
Worst	0	0	0
Stdev	0	0	0

Table 3. Results of Rastrigin in search space of $[-5.12, 5.12]^{N=10}$

	cWDO	WDO with uniform	AWDO
Best	0	0	0
Mean	2.9884	0	0
Worst	20.1973	0	0
Stdev	3.6242	0	0

Table 4. Results of Griewank in search space of $[-600, 600]^{N=10}$

	cWDO	WDO with uniform	AWDO
Best	0.0128	0	0
Mean	0.1957	0	0
Worst	0.6214	0	0
Stdev	0.1263	0	0

Table 5. Results of Rosenbrock in search space of $[-30, 30]^{N=10}$

	cWDO	WDO with uniform	AWDO
Best	5.9906	3.2219	1.5926e-9
Mean	7.1079	4.1390	7.1087e-7
Worst	7.8595	4.7889	6.2096e-6
Stdev	0.3701	0.2739	1.11917e-6

From Tables 1-5, it can be observed that the AWDO achieves the best pressure performance with the least number of iterations needed. Then comes the WDOu, where the WDO performance can be improved by randomly selecting the inherent terms and achieve better results than fixing these terms as in classical WDO [6,7]. In addition, both of the proposed algorithms perform better than the WDO with Levy flight algorithm proposed in [12], where same study conducted with these benchmark functions. Figures 2-5 illustrate the mean pressure value comparison of the algorithms over the number of iterations. The pressure values are averaged over the 50 runs and plotted at each iteration for the first 1000 iterations or less. AWDO provides the best mean pressure performance and reaches the global optimum the fastest. Specifically, for the *Rosenbrock* function, AWDO is the only algorithm that is not stuck at a local minima and continues the search until the maximum number of iterations is exhausted.

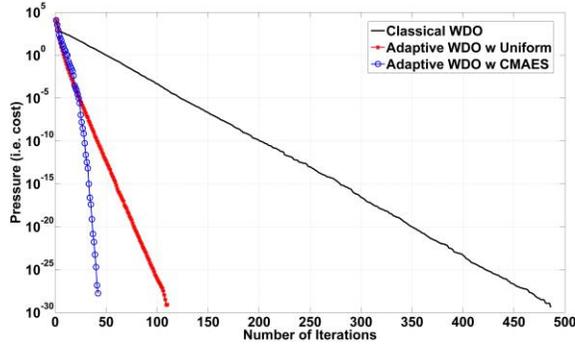


Figure 2. Comparison of mean best pressure for Sphere.

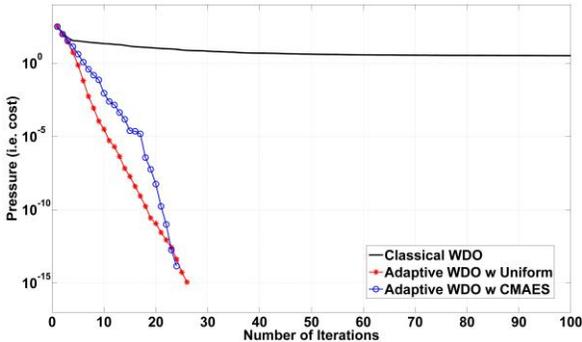


Figure 3. Comparison of the mean best pressure for Rastrigin.

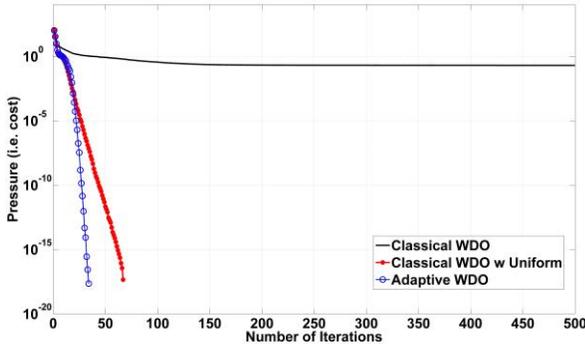


Figure 4. Comparison of the mean best pressure for Griewank.

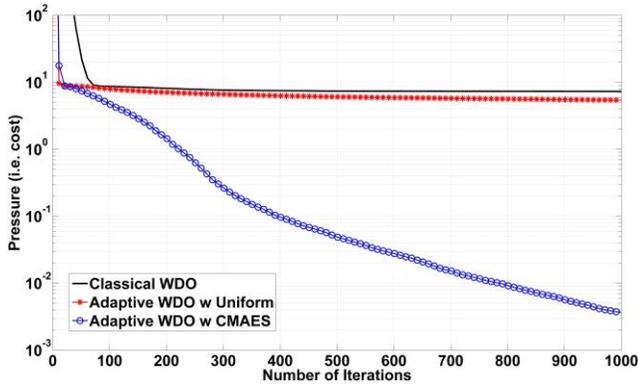


Figure 5. Comparison of the mean best pressure for Rosenbrock.

6. CONCLUSIONS

In this paper, we proposed two new methods to create adaptive WDO algorithms and demonstrated their performances against the classical WDO and WDOLE. Numerical test results show that the proposed algorithms improve performance by achieving faster convergence and better results. These adaptive methods also eliminate the need for tuning the internal WDO terms, alleviating the burden to select the optimum values for the problem at hand without requiring additional function calls.

7. REFERENCES

- [1] Haupt, R. L. and Werner, D.H. 2007. *Genetic Algorithms in Electromagnetics*. Wiley-IEEE Press, New Jersey.
- [2] Kennedy, J. and Eberhart R. 1995. *Particle Swarm Optimization*. In *Proceedings of the Ninth Int. Conf. on Neural Networks*, Nov. 1995, vol. 4, 1942-1948.
- [3] Dorigo, M. and Stutzle, T. 2004 *Ant Colony Optimization*, MIT Press, Cambridge, MA.
- [4] Bayraktar, Z., Bossard J., Wang, X. and Werner, D. H. 2012. *A real-valued parallel clonal selection algorithm and its application to the design optimization of multi-layered frequency selective surfaces*. *IEEE Trans Antennas Propag.* 4/2012, 1831-1843.
- [5] Price, K., Storn, R. M. and Lampinen, J. A. 2005. *Differential Evolution: A Practical Approach to Global Optimization*. Springer-Verlag, Berlin.
- [6] Bayraktar, Z., Komurcu, M., Bossard J. A., and Werner, D. H. 2003. *The Wind Driven Optimization Technique and its Application in Electromagnetics*. *IEEE Trans Antennas Propag.*
- [7] Bayraktar, Z., Komurcu, M., and Werner, D. H. 2010. *Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics*. In *Proceedings of the IEEE Antenna and Propagation Society International Symposium*.
- [8] Bayraktar, Z., Komurcu, M., Jiang, Z., Werner, D. H., and Werner, P. L. 2011. *Stub-Loaded Inverted-F Antenna Synthesis via Wind Driven Optimization*, In *Proceedings of the IEEE Int. Symp. on Antennas and Propagation and USNC/URSI National Radio Science Meeting*.
- [9] Bayraktar, Z., Turpin, J. P., and Werner, D. H., *Nature-Inspired Optimization of High-Impedance Metasurfaces with Ultra-Small Interwoven Unit Cells*, *IEEE Antennas Wireless Propag Lett*, 2011, vol. 10, pp 1563-1566.
- [10] Hansen, N. and Ostermeier, A. 1996. *Adaptive arbitrary normal mutation distribution in evolution strategies: The covariance matrix adaptation*. *Proceedings of the IEEE Int. Conf. on Evolutionary Computation*, 312-317.
- [11] Gregory, M. D., Bayraktar, Z. and Werner, D. H. 2011. *Fast optimization of electromagnetic design problems using the covariance matrix adaptation evolutionary strategy*. *IEEE Trans Antennas Propag.* 4/2011, vol. 59, no. 4, 1275-1285.
- [12] Segundo, E. H. V., Amoroso, A. L., Mariani, V. C., and Coelho, L. S. 2014. *A Wind Driven Approach Using Levy Flights for Global Continuous Optimization*. *IEEE 2nd Int. Conf. on Artificial Intelligence, Modeling and Simulation*.