WEIGHTED VERTICES OPTIMIZER (WVO): A NOVEL METAHEURISTIC OPTIMIZATION ALGORITHM

SOHEIL DOLATABADI
Department of electrical and computer engineering
University of Tabriz, Tabriz, Iran

(Communicated by Wenxing Zhu)

Abstract. This paper introduces a novel optimization algorithm that is based on the basic idea underlying the bisection root-finding method in mathematics. The bisection method is modified for use as an optimizer by weighting each agent or vertex, and the algorithm developed from this process is called the weighted vertices optimizer (WVO). For exploitation and exploration, both swarm intelligence and evolution strategy are used to improve the accuracy and speed of WVO, which is then compared with six other popular optimization algorithms. Results confirm the superiority of WVO in most of the test functions.

1. Introduction. Optimization algorithms are used to identify the minimum or maximum value of a function without using mathematical approaches. Such algorithms are generally classified into two major categories, namely, heuristic and metaheuristic algorithms [16]. Heuristic algorithms are used to determine an optimum solution by trial and testing within a limited computational time. These methods provide excellent results in terms of finding suboptimal solutions, but they do not guarantee the identification of a global optimum solution. Conversely, metaheuristic algorithms use trial and testing, higher-level techniques, and exploration and exploitation to ascertain a global optimum solution. They are divisible into many categories, among which two are important and are the focus of this study. The first category consists of evolutionary algorithms, which use genetic operators (selection, mutation, and crossover) to a certain degree. The most popular algorithm in this class is the genetic algorithm (GA) [7]. The second category comprises algorithms that are based on swarm intelligence and the collective behavior of animals. Particle swarm optimization (PSO) [5], artificial bee colony (ABC) [10], and ant colony optimization (ACO) [4] fall within this category. Although metaheuristic algorithms are not limited in terms of sources of inspiration, most of them are grounded in the characteristics of nature and are designed to mimic a natural phenomenon to find a solution. Examples of nature-inspired algorithms are GA, PSO, and invasive weed optimization (IWO). The advantage of these algorithms is that the nature behaviors and techniques that underlie them have been examined and improved over thousands of years and therefore serve as reliable bases for algorithmic development. A representative example is GA, which is based on Darwin's
evolution theory and defines the best solution by generating new solutions from the highest-performing agents and eliminating those that are unfit. Despite the benefits derived from using nature techniques, however, these also present limitations that prevent innovation. For example, GA should use only two parents to produce offspring; otherwise, the essence of the algorithm will completely change and become meaningless. Yet, the participation of more than two parents enables the production of more qualified offspring and guarantees improved diversity, which leads to more accurate solutions. Some other algorithms, such as the cultural algorithm (CA) [14] and imperialist competitive algorithm (ICA), are developed on the basis of human behavior [2]. In CA, agents are spread throughout a belief space, and the algorithm employs an evolutionary strategy to find an optimum solution. ICA also uses an evolutionary approach, but it adopts a mathematical model of human social evolution, contrary to GA, which uses the biological evolution of animals. The aim of this study is to introduce a new metaheuristic algorithm that is not inspired by nature but by a mathematical approach called the bisection root-finding method [3]. The proposed algorithm is called the weighted vertices optimizer (WVO), which simultaneously employs swarm intelligence and an evolutionary strategy [6] for exploration and exploitation purposes. Briefly, the algorithms and techniques from which the raw concept of the WVO algorithm is drawn are described below:

- Bisection root-finding method: This idea is employed to enable the algorithm to guide an agent toward the global best solution.
- Evolutionary strategy from GA: The strategy enables the generation of a new solution through genetic operators. This functionality is extended in WVO, which is designed to choose more than two parents and the transfer characteristics of each parent on the basis of their fitness.
- Swarm intelligence from swarm-based algorithms (PSO, ACO, etc.): Particles or agents come into contact with one another and guide other particles toward the best solution in each iteration.

The rest of the paper is organized as follows. Section 2 presents the definition of the bisection root-finding method and the raw concept of WVO. Subsection 2.1 explains the selection method used in WVO, and Subsection 2.2 describes the methods used by WVO to produce new generations. Section 3 discusses the results of a simulation, and Subsection 3.2 presents the solution to a practical example. Section 4 concludes the paper.

2. The proposed algorithm. The concept of the proposed algorithm is based on the bisection root-finding method where the roots are defined by guessing two initial points and using iterating trial and test (Figure 1-a). In each iteration of this method, three new possible solutions (C1 to C3) are generated and only one of these solutions along with one of primary points (G1 or G2) with the best fitness will be kept for next iteration. In rest of the paper, these primary points are called vertices. With some changes, the bisection method could be used for finding minimum of function (Figure 1-b).

In Figure 1-b, G1 and G2 represent two initial points and the C1 to C3 points are possible new vertices. Contrary to the bisection method, these points will not be used directly as a new solution but as a guide to move G1 or G2 toward them. In one hand, selecting C1 for moving G1 or selecting C3 for moving G2 causes more distance between G1 and G2 which in this paper, this move is known as backward movement. On the other hand, the C2, known as geometric center and
Figure 1. a) the bisection method b) the raw concept of WVO algorithm using two vertices

is placed in middle of G1 and G2, causes G1 and G2 come closer which known as forward movement. To reduce amount of computation, only the best vertex will be considered for generating new solution or to be moved. This strategy will be very helpful when number of vertices is high. In addition, fitness value of each vertex are taken in to account to calculate geometric center of vertices. The array of problem variables for $N_{Var}$-dimensional optimization for WVO is given in 1. Different optimization methods have used different terminology to name this array, for instance “chromosome” in GA or “particle position” is PSO. In this paper it is called “vertex” as mentioned earlier.

$$\text{Vertex}(i) = [x^i_1 \ x^i_2 \ \cdots \ x^i_{N_{Var}}]_{1 \times N_{Var}} \quad i = 1, \ldots, N_{Pop}$$  \hspace{1cm} (1)

Which $N_{Pop}$ indicate number of vertices. Each vertex is evaluated using cost function as follow:

$$\text{Profit}_i = f(\text{Vertex}(i)) = f([x^i_1 \ x^i_2 \ \cdots \ x^i_{N_{Var}}])$$  \hspace{1cm} (2)

As mentioned earlier, the two possible actions for creating new solutions are forward movement and backward movement which both of them are depicted in 2. In this figure, the white-colored circles represent the selected vertices which their sizes are related to their fitness values.

In the forward movement, two factors form new solution by moving the best vertex: 1) toward the geometrical center of selected vertices and 2) toward the global best. Similarly, the backward movement is formed by moving the best vertex: 1) against the geometric center as much as D-unit and 2) toward the global best. The whole processes of selection and movement are formulated and are explained in below subsections.

2.1. Selecting vertices. As mentioned earlier, the WVO lets operator to define number of participant vertices for producing new generation of solutions which is a new concept in evolutionary strategy. The vertices are selected randomly using integer unified random function which produces integer values between 1 to $N_{Pop}$.

$$P_i \sim [U(1, N_{Pop})] \quad i = 1, \ldots, N_V$$  \hspace{1cm} (3)
where the \( N_V \) represents number of desired vertices. This number is equal to two for genetic algorithm because only two parents are involved to produce new offspring. Unlike the GA, the WVO is able to use two or more vertices to produce new solutions. The \( P \) is a random integer number between 1 to \( N_{pop} \) that defines which agents should be used as vertices to produce a new vertex. The set of selected vertices is given in 4.

\[
S_{Sel} = \{ \text{Vertex}(P_1), \text{Vertex}(P_2), \ldots, \text{Vertex}(P_{N_V}) \} 
\]

The \( S_{Sel} \) represents the set of selected vertices. The \( V_{sel} \) shows the matrix form of \( S_{sel} \).

\[
V_{sel} = \begin{bmatrix}
    x_{1,1} & x_{1,2} & \cdots & x_{1,N_{Var}} \\
    x_{2,1} & x_{2,2} & \cdots & x_{2,N_{Var}} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{N_V,1} & x_{N_V,2} & \cdots & x_{N_V,N_{Var}}
\end{bmatrix}_{N_V \times N_{Var}} 
\]

2.2. Producing new generation. The forward movement and backward movement are explained in this subsection.

2.2.1. Forward movement. The forward movement is consisted of three terms which are presented in 6.

\[
\text{NewVertex}(j) = \text{Vertex}_{BV}(j) + C_G \times (\text{Vertex}_{GB}(j) - \text{Vertex}_{BV}(j)) + C_F \times (\text{Geo}(j) - \text{Vertex}_{BV}(j)) 
\]

where \( C_F \) and \( C_G \) are coefficient of forward movement and coefficient of moving toward the global best. The \( V_{BV} \) and \( \text{Vertex}_{GB} \) are the best vertex from selected vertices set \( (S_{sel}) \) and the global best. \( \text{Geo} \) represents geometrical center or weighted sum of selected vertices which is calculated using 7.
The $CV$ represents weight of vertex. The $CV$ for $i^{th}$ vertex is calculated using 8.

$$CV(i) = \frac{W_{GB} - W_{GW}}{Profit_{GB} - Profit_{GW}} \times (Profit_i - Profit_{GB}) + W_{GB} , i = 1, \ldots, N_V$$

Here $W_{GB}$, $W_{GW}$, $Profit_{GB}$, $Profit_{GW}$ and $Profit_i$ are assumed weight for the global best, assumed weight for the global worst, cost value of global best, cost value of global worst and cost value of $i^{th}$ vertex.

2.2.2. Moving backward. The backward movement is calculated by

$$NewVertex(j) = Vertex_{BV}(j) + C_G \times (Vertex_{GB}(j) - Vertex_{BV}(j)) - C_B \times (Geo_c(j) - Vertex_{BV}(j))$$

The CB is coefficient for backward movement.

2.3. Speed and accuracy. In WVO, operator is able to define a value between zero to one for $V_{Speed}$ which is responsible to make a balance between accuracy and speed of the algorithm. In other words, the operator is able to choose whether to find a very optimum solution in expense of long computation time and numerous function evaluations or finding near optimum solution in shorter time and fewer function evaluations. The steps of WVO algorithm is given in below:

I Define objective function: $f(\vec{X}), \vec{X} = (x_1, x_2, \ldots, x_{N_{Var}})$

II Initialize the WVO algorithm with generating random values and calculating cost value

III Selecting some vertices, calculating geometric center and generating new agents by moving the best selected vertex: 1) forward or backward 2) toward the global best.

IV Function evaluation of each movement according to $V_{Speed}$ and forming a primary new generation

V Producing the main new generation by mixing primary and old generations and selecting best agents of them as many as $N_{Pop}$.

VI If number of iteration reached its maximum stop the algorithm else go to step II.

VII End

The whole procedure also is shown in 3.

3. Simulation results. The proposed algorithm is tested on six standard function and the obtained results are compared with those driven from IWO [13], CA, harmony search (HS) [9], PSO and GA. In addition, the numeric results for modified versions of IWO [11], CA [1] and HS [15] are also presented. The results are obtained by running over 10 times independent simulations on each optimization algorithm. The number of function evaluations is used as termination criteria and it is set to 150000 for all algorithms. The parameters of WVO algorithm for these
Figure 3. the flowchart of WVO algorithm

tests are provided in Table 1. To avoid complexity and for a better comparison, only 2 vertices are considered for WVO algorithm.

A number of well-known benchmark functions are used for simulation and are given in Table 2 which are extracted from [17]. The 2D plot of each test function is shown in Figure 4. The simulations are implemented in MATLAB™ computer program.
Table 1. parameters of WVO

| C_F | C_B | C_G | N_V | V_Speed | W_GB | W_GW |
|-----|-----|-----|------|---------|------|------|
| 0.6 | 0.3 | 0.085 | 2    | 0.6     | 10   | 1    |

Table 2. benchmark optimization functions

| ID | Name | Function | Bound | Global Min |
|----|------|----------|-------|------------|
| F1 | Shubert | \( \left( \sum_{i=1}^{5} \cos((i + 1)x_1 + 1) \right) \) | \([-2.12, 2.12]^2\) | -186.7309 |
| F2 | Six-hump camel back | \( \left( 4 - 2.1x_1^2 + \frac{x_1^4}{4} \right)x_1^2 + x_1x_2 + \left( -4 + 4x_2^2 \right)x_2^2 \) | \([-5.5]^2\) | -1.0316285 |
| F3 | Sphere | \( \sqrt{\sum_{i=1}^{D} x_i^2} \) | \([-32.32]^A_{10}\) | 0 |
| F4 | Ackley | \( -A \times \exp \left( -0.02 \sqrt{\sum_{i=1}^{D} x_i^2} \right) - \exp \left( \sum_{i=1}^{D} \cos(2\pi x_i) \right) + A; A = 20 \) | \([-100.100]^A_{10}\) | 0 |
| F5 | Griewank | \( 1 + \frac{1}{3000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right) \) | \([-600.600]^A_{10}\) | 0 |
| F6 | Rastrigin | \( 10D + \sum_{i=1}^{D} \left( x_i^2 - 10 \cos(2\pi x_i) \right) \) | \([-5.12, 5.12]^A_{10}\) | 0 |

The plot of F1 for two dimensional variables is shown in Figure 5. The global minimum of this function is -186.7309 at (-0.8003, -1.4251). The position of vertices in first iteration of WVO algorithm is shown in Figure 6 which shows that most of vertices have found a local minimum or are near to the global minimum. The position of vertices in next iterations are depicted in Figures 7a to 7d. From Figure 7a it can be seen that all of vertices had found the global minimum or the second best optimum point. In Figure 7b the vertices are trying to find the global minimum and finally in 9th iteration the global minimum have been found by all agents and therefore the vertices in local minimum are eliminated (Figure 7c). In 13th iteration the algorithm have found the accurate place of global minimum (Figure 7d). Figure 8 depicts the cost value versus iteration of the WVO along with other algorithms.

The cost value of F2 functions for each iteration is presented in Figure 9. As it can be seen, again the WVO algorithm has incredibly faster convergence speed and it has come near to solution in 9th iteration and has reached it in 19th iteration.

The Figure 10 shows performance of each optimization method for the F3. Although the WVO algorithm reaches the stopping criteria (maximum number of function evaluation) in 78th iteration, it still has the highest decline rate among other algorithms. For a better insight, the cost value for each method in the 78th iteration is provided in Table 3.

The performance of WVO for F4 is superior to other methods. It has the highest decline rate and reaches the lowest cost value in the 67th iteration (Figure 11).

The WVO algorithm reaches the optimum cost value (0) in 41st iteration and also faster than other optimization methods (Figure 12).
Figure 4. 2D plot of test functions

Figure 5. 3D sketch of Shurb’s function (F1)

Table 3. Cost value of each optimization algorithm in 78th iteration

| Method | Cost value |
|--------|------------|
| WVO    | 1.78 E-15  |
| PSO    | 7.36 E-4   |
| GA     | 9.17 E-5   |
| IWO    | 2.157      |
| HS     | 2.56E-3    |
| CA     | 5.93E-6    |
| mIWO   | 1.23 E-5   |
| mHS    | 8.69 E-9   |
| mCA    | 5.12 E-10  |
WEIGHTED VERTICES OPTIMIZER (WVO)...

Figure 6. the positions of WVO vertices in first iteration

Figure 7. A) the positions of vertices in second iteration B) the positions of vertices in 5th iteration C) the positions of vertices in 9th iteration D) the positions of vertices in 13th iteration
Figure 8. the cost value versus iteration

Figure 9. cost value of F2 function in each iteration

The performance of optimization methods for the last function is shown in Figure 13 and again the WVO algorithm has the highest decline rate and reaches the zero cost value in 30th iteration.

For a better comparison between WVO and other optimization methods, all information are gathered in Table 4. The ranking is based on considering two aspects, first priority is given to cost value and second priority to number of iteration.
3.1. **Composition functions.** In addition to the six chosen traditional test functions, the performance of WVO is also measured by using three novel composition functions (CF) [12]. These CFs are formed from number of traditional test functions which were introduced in Table 2 except Weierstrass function which is given in 10. The understudy CFs are shown in Table 5. Due to lack of space, the details of CF is not presented here and can be found in [12].
Figure 12. cost value of F5 function in each iteration

Figure 13. cost value of F6 function in each iteration

\[ F_7 = f(x) = \sum_{i=1}^{D} \left( \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(\pi b^k)] \]  

(10)

\[ a = 0.5, \ b = 3, \ k_{\max} = 20, \ x \in [-0.5, 0.5]^{10} \]

The numeric results of each algorithm for 20 times independent runs are given in 6. the stopping criteria is number of function evaluation which is set at 50000.
Table 4. The performance of each optimization method

|   | WVO | PSO | GA | DWO | HS | CA | mWVO | mHS | mCA |
|---|-----|-----|----|-----|----|----|------|-----|-----|
| F1 | N   | 15  | 27 | 184 | 88 | 47 | 132  | 44  | 24  |
|    | B   | 176.7309 | 176.7309 | 176.7309 | 176.7309 | 176.7309 | 176.7309 | 176.7309 | 176.7309 |
|    | R   | 1   | 5  | 3  | 9  | 7  | 6   | 8   | 4   |
| F2 | B   | -176.7309 | -176.7309 | -176.7309 | -176.7309 | -176.7309 | -176.7309 | -176.7309 | -176.7309 |
|    | R   | 4   | 7  | 9  | 8  | 8  | 9   | 9   | 2   |
| F3 | B   | 1.711 E-58 | 7.643 E-128 | 2.432 E-6 | 2.19 E-10 | 1.891 E-143 | 1.312 E-8 | 1.33 E-2 | 0    |
|    | R   | 5   | 1  | 4  | 9  | 7  | 8   | 9   | 2   |
| F4 | B   | 8.881 E-16 | 4.441 E-15 | 20    | 6.21 E-5 | 20.29 | 6.132 E-8 | 2.52 E-4 | 1.23 E-4 |
|    | R   | 4   | 7  | 8  | 4  | 9  | 8   | 3   | 1   |
| F5 | B   | 0.009747 | 0.90071 | 1.52 E-8 | 20.25 | 2.38 E-3 | 1.58 E-32 | 9.741 E-2 | 0    |
|    | R   | 1   | 7  | 2  | 8  | 4  | 9   | 5   | 3   |
| F6 | B   | 0    | 9.6907 | 8.9552 | 4.061 E-10 | 22.94947 | 3.253 E-9 | 3.155 E-21 | 4.653 E-3 |
|    | R   | 1   | 7  | 2  | 8  | 4  | 9   | 5   | 3   |
| Σ  | R   | 1   | 5  | 2  | 9  | 6  | 8   | 6   | 3   |

1 N: Number of iteration - 2 B: Best cost value - 3 R: Rank

Table 5. The understudy composition functions [12]

|     | CF1 | CF2 | CF3 |
|-----|-----|-----|-----|
| f₁, f₂, ..., f₁₀ = F₅ |

3.2. Practical example. In this subsection, it is targeted to find optimum gains of PID controller for automatic voltage regulator (AVR) system using WVO which lately it was done in [8] using PSO. In synchronous generators, the AVR is responsible to maintain the generator voltage and control reactive power flow. The linearized model of AVR system is consisted of four major parts: amplifier, exciter, generator and sensor which along with a PID controller are shown in Figure 14. The intent of using optimization algorithm is that to find the best values for $K_I$, $K_P$.
Table 6. results of optimization algorithms for three CFs

|       | PSO [12]     | DE [12]     | GA           | WVO          |
|-------|---------------|-------------|--------------|--------------|
| CF1   | 1.7203 E2     | 1.4441 E2   | 1.3451 E2    | 1.1121 E2    |
|       | 3.2869 E1     | 1.9401 E1   | 1.9142 E1    | 1.4232 E1    |
| CF2   | 3.1430 E2     | 3.2486 E2   | 3.2314 E2    | 3.0021 E2    |
|       | 2.0006 E1     | 1.4784 E1   | 1.8154 E1    | 1.6823 E1    |
| CF3   | 8.3450 E1     | 1.0789 E1   | 7.5421 E1    | 3.8124 E1    |
|       | 1.0111 E2     | 2.6640 E0   | **1.0512 E1**| 8.5412 E1    |

Figure 14. block diagram of AVR along with PID controller [17]

and $K_D$ to improve time-domain characteristics of AVR such as overshoot, output error, rise time, settling time and etc. Different approaches regarding calculation of output error are introduced but in this study the integral absolute error (IAE) is used (Eq. 11). In addition, rise time (RT) and overshoot (OS) also are considered in cost function. The cost function is given in 12.

\[
\text{IAE} = \int_{0}^{\infty} |V_{ref} - V_{out}| \, dt = \int_{0}^{\infty} |e(t)| \, dt \tag{11}
\]

\[
\text{Cost Function} = 5 \times RT + 3 \times IAE + OS \tag{12}
\]

where the $V_{ref}$, $V_{out}$ and $e(t)$ are reference voltage, output voltage and difference between reference and output voltage. In this study, according to desired characteristics, the coefficients of rise time and error are chosen 5 and 3 respectively. The parameters’ value of AVR system are given in Table 7 [8].

The parameters for WVO algorithm are similar to values which are provided in Table 1. Figure 15 shows cost value versus iteration of WVO along with GA and PSO algorithms which proves good performance of WVO. In addition, Table 8 provides obtained $K_I$, $K_P$ and $K_D$ by each optimization method along with the obtained numeric results. The number of population and number of iteration are
Table 7. value of AVR’s parameters [17]

| Parameter | Value |
|-----------|-------|
| $K_A$     | 10    |
| $\tau_A$ | 0.1   |
| $K_E$     | 1     |
| $\tau_E$ | 0.4   |
| $K_G$     | 1     |
| $\tau_G$ | 1     |
| $K_R$     | 1     |
| $\tau_R$ | 0.01  |

Table 8. obtained values and the result for each optimization method

| Method | KP     | KI     | KD     | RT     | ST (sec) | OS (%)  | Final error (%) | Cost value |
|--------|--------|--------|--------|--------|---------|---------|-----------------|------------|
| WVO    | 0.600518 | 0.41376 | 0.20136 | 0.3101 | 0.5013  | 0.0003  | 0               | 3.11706    |
| PSO    | 0.600532 | 0.41386 | 0.20137 | 0.3141 | 0.5013  | 0.0017  | 0               | 3.11752    |
| GA     | 0.610065 | 0.42965 | 0.20784 | 0.3226 | 0.5605  | 0.1522  | 0.018           | 3.17785    |

The search domain for each variable is in range of [0, 2]. The step response of each optimization method is presented in Figure 16.

3.3. Effect of WVO’s parameters. The proposed optimization algorithm uses independent parameters to achieve efficient performance. Here, the effect of WVO’s parameters, namely $C_F$, $C_B$, $C_G$, $W_{GB}$ and $W_{GW}$ on optimization results are studied but the $N_V$ is extensively studied in a separate subsection. The results of variation of the mentioned parameters are presented in Table 9.

3.3.1. Number of vertices. In above subsections, the number of vertices ($N_V$) was set to two and it was constant to avoid chaos. Here, it is aimed to observe the
Figure 16. The step response of without PID controller and with optimized gains

Table 9. Effect of $C_F$, $C_B$, $C_G$, $W_{GB}$ and $W_{GW}$ on performance of WVO

| Function | $C_F$ | The best cost | Iteration | $C_B$ | The best cost | Iteration | $C_G$ | The best cost | Iteration | $W_{GB}$ | The best cost | Iteration | $W_{GW}$ | The best cost | Iteration |
|----------|-------|---------------|-----------|-------|---------------|-----------|-------|---------------|-----------|----------|---------------|-----------|----------|---------------|-----------|
| F5       | 0.2   | 3.12E-12      | 46        | 0.2   | 0             | 43        | 0.02  | 0             | 73        | 2        | 0             | 47        | 2        | 0             | 41        |
|          | 0.4   | 2.02E-19      | 42        | 0.4   | 0             | 41        | 0.04  | 0             | 45        | 5        | 0             | 43        | 5        | 0             | 45        |
|          | 0.6   | 0             | 42        | 0.6   | 0             | 42        | 0.06  | 0             | 45        | 10       | 0             | 42        | 10       | 0             | 47        |
|          | 0.8   | 0             | 42        | 0.8   | 2.31E-26      | 43        | 0.08  | 0             | 41        | 15       | 0             | 42        | 15       | 0             | 47        |
|          | 1     | 3.12E-30      | 44        | 1     | 8.64E-23      | 45        | 0.1   | 0             | 53        | 20       | 0             | 43        | 20       | 0             | 48        |
| F6       | 0.2   | 2.31E-6       | 45        | 0.2   | 0             | 41        | 0.02  | 0             | 2.31E-26  | 45       | 2             | 0         | 45       | 2             | 0         |
|          | 0.4   | 0             | 42        | 0.4   | 0             | 42        | 0.04  | 0             | 43        | 5        | 0             | 43        | 5        | 0             | 43        |
|          | 0.6   | 0             | 42        | 0.6   | 1.12E-28      | 45        | 0.06  | 0             | 43        | 10       | 0             | 43        | 10       | 0             | 43        |
|          | 0.8   | 0             | 45        | 0.8   | 6.78E-25      | 49        | 0.08  | 0             | 41        | 15       | 0             | 44        | 15       | 0             | 45        |
|          | 1     | 0             | 45        | 1     | 1.32E-24      | 51        | 0.1   | 0             | 44        | 20       | 0             | 44        | 20       | 0             | 48        |
Table 10. the effect of $N_V$ value on speed of algorithm

| Function | $N_V$ | The best cost | Iteration |
|----------|-------|---------------|-----------|
| F5       | 2     | 0             | 47        |
|          | 3     | 0             | 43        |
|          | 4     | 0             | 42        |
|          | 5     | 0             | 41        |
|          | 10    | 0             | 43        |
|          | 15    | 4.55E-8       | 116       |
| F5       | 2     | 0             | 45        |
|          | 3     | 0             | 43        |
|          | 4     | 0             | 43        |
|          | 5     | 0             | 44        |
|          | 10    | 0             | 44        |
|          | 15    | 0.406497      | 116       |

effect of $N_V$ variation on performance of algorithm, and therefore a comparison is done among WVOs with different $N_V$. The parameters of WVO is similar to pervious simulations, except $V_{Speed}$ is set to 0.6 to reduce number of function evaluation. The presented results in Figure 17, Figure 18 and Table 10 are average of 10 times simulation on F5 and F6 test functions. As it is observable from the results, $N_V$ should be chosen carefully to increase speed and accuracy of algorithm otherwise the optimization may not lead to a proper solution.

Figure 17. the logarithmic plot of cost function versus iteration for F5 function and different $N_V$
4. Conclusion. In this paper, a novel heuristic optimization algorithm based on the bisection root-finding method is introduced. The proposed algorithm, called WVO, uses a number of parents (vertices) and calculates the geometric center of vertices to produce new generations. Similar to other optimization algorithms, WVO takes into account the best global agent for producing new generations. To investigate the performance of the algorithm, a simulation is carried out on 9 test functions, and the results are compared with those derived for PSO, GA, IWO, HS, and CA and also modified versions of IWO, HS and CA. In addition to traditional test functions, three composition test functions are employed to evaluate performance of WVO. The findings indicate that WVO exhibits very good performance in a number of test functions and reasonable performance in others. WVO is then applied in a practical example, in which the optimum PID’s gains of an AVR system require identification. The comparison of the results derived by WVO, GA, and PSO confirm the superiority of the proposed algorithm.

REFERENCES

[1] M. Z. Ali, N. H. Awad, P. N. Suganthan and R. G. Reynolds, A modified cultural algorithm with a balanced performance for the differential evolution frameworks, Knowledge-Based Systems, 111 (2016), 73–86.

[2] E. Atashpaz-Gargari and C. Lucas, Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition, IEEE Congress on Evolutionary Computation, Singapore, 2007.

[3] R. L. Burden and J. D. Faires, Numerical Analysis, 3rd edition, Prindle, Weber and Schmidt, 1985.

[4] M. Dorigo, V. Maniezzo and A. Colomn, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26 (1996), 29–41.

[5] R. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995.
[6] L. J. Fogel, A. J. Owens and M. J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley and Sons, 1966.

[7] D. E. Goldberg and J. H. Holland, Genetic algorithms and machine learning, Machine Learning, 3 (1988), 95–99.

[8] Z.-L. Gaing, A particle swarm optimization approach for optimum design of PID controller in AVR system, IEEE Transactions on Energy Conversion, 19 (2004), 384–391.

[9] Z. W. Geem, J. H. Kim and G. Loganathan, A New Heuristic Optimization Algorithm: Harmony Search, Simulation, 76 (2001), 60–68.

[10] D. Karaboga and B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, International Fuzzy Systems Association World Congress, 2007.

[11] E.-H. Kenane, F. Djahli and C. Dumoud, A novel Modified Invasive Weeds Optimization for linear array antennas nulls control, 4th International Conference on Electrical Engineering (ICEE), Boumerdes, Algeria, 2015.

[12] J. Liang, P. Suganthan and K. Deb, Novel composition test functions for numerical global optimization, Swarm Intelligence Symposium, Pasadena, CA, USA, 2005.

[13] A. Mehrabian and C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecological Informatics, 1 (2006), 355–366.

[14] R. G. Reynolds, An Introduction to Cultural Algorithms, 3rd Annual Conference on Evolutionary Programming, 1994.

[15] W. Xiang, M. An, Y. Li, R. He and J. Zhang, An improved global-best harmony search algorithm for faster optimization, Expert Systems with Applications, 41 (2014), 788–803.

[16] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms: Second Edition, Luniver press, 2010.

[17] A. E. M. Zavala, A. H. Aguirre and E. R. V. Diharce, Constrained optimization via particle evolutionary swarm optimization algorithm (PESO), 7th annual conference on Genetic and evolutionary computation, Washington DC, USA, 2005.

Received August 2017; 1st revision January 2018; final revision March 2018.

E-mail address: s.dolatabadi93@ms.tabrizu.ac.ir