

Volcano Eruption Algorithm for Solving Optimization Problems

Eghbal Hosseini, Ali Safaa, Kayhan Zrar Ghafoor, Danda B. Rawat, Mehrdad Saif, Xinan Yang

Abstract—Due to certain attractive features of meta-heuristics, these algorithms have gained popularity in solving optimization problems and have been considered by many researchers. Because meta-heuristics solve just some of optimization problems, so proposing a meta-heuristic to solve all optimization problems in both kinds of discrete and continuous problems could be significant. This paper proposes an algorithm based on simulation of volcano eruption process, which solves NP-hard problems such as non-linear and multi-level programming problems. The algorithm's feasibility and efficiency is then evaluated by applying it to solve a number of test problems reported in the literature. Based on the solutions and number of required iterations, it is concluded that the proposed meta-heuristic algorithm performs remarkably well. Finally, the proposed algorithm solves some large size benchmark linear programming and also the routing problems in the Internet of Vehicles (IoV).

Index Terms—Volcano Eruption Algorithm, Meta-Heuristic Approaches, Internet of Vehicles Problem, Optimization Problems.

I. INTRODUCTION

Meta-heuristic algorithms can be used to train neural network in solving real-life problems, though every approach has its own limitations. Some of the prominent meta-heuristic algorithms include Particle Swarm Optimization (PSO) [1] [22] and Autonomous Particles Groups for PSO, (AGPSO) [23] Bat Algorithm (BA) [6] and its recent application in optimizing beamforming for mmWave in 5G communication [24], Fire Fly (FF) [9]. On the other hand, there are some more nature creature inspired algorithms were proposed recently to solve optimization problems, such as: Whale Optimization Algorithm (WOA) [25], Ions Motion Optimization (IMO) [26] and Grey wolf optimizer (GWO) [27]. While other are inspired from nature phenomena, such as: Chaotic Gravitational Search Algorithm (CGSA) [28] and the recent application of Multi-Verse algorithm in optimizing the accuracy of fraud

detections in smart e-commerce ecosystem [29]. However, no heuristic algorithm is the best suited to solve all optimization problems. Moreover, limitations of expensive computational cost, occurrence of premature convergence, mutation rate, crossover rate, time consumingness evaluation leads to enhance existing algorithm or propose new one. In machine learning, classification is a supervised learning process to determine appropriate dataset for a new observation based on the performance through training set. In machine learning, classification in a supervised learning process refers to the process of computer learning to which class of data a new set of observation belongs. This is based on a prior learning conducted on a labeled training set. Evolutionary or nature-inspired metaheuristic algorithms can be a good option in the process of designing/training a classification system. As an example, Support Vector Machine (SVM) is an efficient supervised learning algorithm that can be applied for classification [8]. The optimization of SVM parameters is possible through algorithms like PSO or FF. Feature selection plays a vital part of the classification. It turns out that feature selection can be achieved through parameter optimization of SVM using a meta-heuristic algorithm [8]. Feature selection through this process is another example of an application area where a metaheuristic approach could be effective. It should be noted however that there are certain challenges with SVM such as: its high algorithmic complexity which leads to higher computational cost, extensive memory requirements, and selection of appropriate kernel parameters which may be tricky [10]. As a result, success of a meta-heuristic approach in one instance, may not guarantee a similar success in another instance. Nevertheless, optimization researchers have proposed heuristic approaches for solving some specific problems (e.g. see [1-3] and references cited therein). Additionally, researchers have tried to solve optimization problems by simulating several algorithms based on behavior of animals and insects, natural phenomena, or scientific theories [4-13]. Some of these proposed algorithms are: artificial bee colony algorithm [4], krill herd algorithm [5], social spider optimization [7], chicken swarm optimization (CSO) [8], big bang algorithm (BBA) [10], laying chicken algorithm (LCA) [11,18], modified genetic algorithm [12], [30], combined meta-heuristic and classic algorithm [13]. Almost all previous meta-heuristics have been inspired from behavior of animals or insects and only one of them has been simulated from a scientific theory [10]. Other related works about state-of-the-art machine learning and optimization algorithms: Collaborative Filtering Bandits [31], Distributed Clustering of Linear Bandits in Peer to Peer Networks [32], Mining λ -Maximal Cliques

Manuscript received Day Month Year.

Eghbal Hosseini is with Department of Mathematics, Faculty of Basic Education, Salahaddin University, Kurdistan Region, Iraq. E-mail: kseghbal-hosseini@gmail.com.

Ali Safaa is with Wolverhampton Cyber Research Institute, School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton, WV1 1LY, UK. E-mail:ali.sadiq@wlv.ac.uk

Kayhan Zrar Ghafoor is with Department of Computer Science, Faculty of Science, Cihan University-Erbil, Erbil, Kurdistan, E-mail: kayhan@iee.org.

Danda B. Rawat is with the Department of Electrical Engineering and Computer Science at Howard University, Washington, DC 20059, USA. E-mail: db.rawat@iee.org.

Mehrdad Saif is with Department of Electrical and Computer Engineering, University of Windsor, Windsor, N9B 3P4 Canada. E-mail: msaf@uwindsor.ca.

Xinan Yang is with Department of Mathematics, University of Essex. E-mail: lxyangk@essex.ac.uk

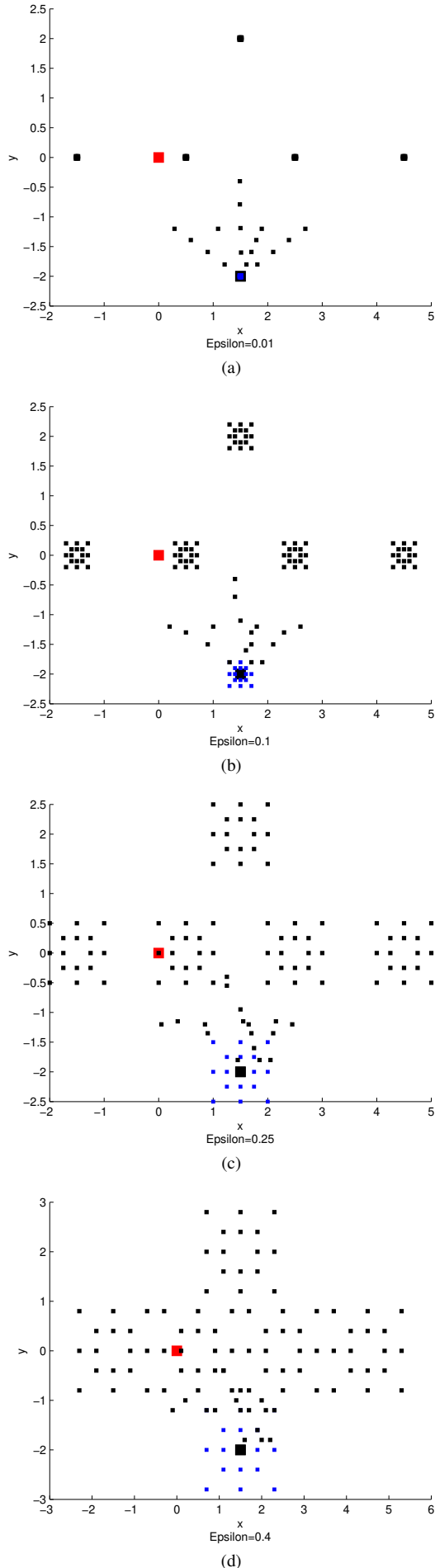


Fig. 1. Movement of initial population and generation of the next population by different epsilons.

from a Fuzzy Graph [34], Stochastic Optimization Techniques for Quantification Performance Measures [35].

A. Main Contribution

This paper proposes for the first time an algorithm which has been inspired from a natural event, that is, volcano eruption. The algorithm is a new optimizer for solving various types of continuous and discrete optimization problems like linear and non-linear, multi-level and multi-objective, transportation and Internet of Vehicles (IOT), and so on. In this article, we first present the underlying concept of Volcano Eruption Algorithm (VEA) and will then present various steps, which converts this natural process into an optimizer. VEA is then implemented through few simple MATLAB codes which are based on initial population, movement of solutions, explosion and eruption in the space. Therefore, the algorithm has an acceptable computational complexity. The proposed algorithm originates from a scientific process, involves simple steps, and implementation. Finally, VEA provides acceptable best solution in relation to other meta-heuristics because it uses different directions and large region of feasible space. On the other hand, the algorithm requires a high number of solution in each iteration because of changing of all solutions in different directions.

II. MOTIVATION

An opening or a hole on the surface of earth that acts as a vent for release of pressurized gases, ashes, and molten rock or magma deep beneath the surface of earth is referred to as a volcano. Deep under ground, pressurized magma is passed through a passage way or a conduit, called the volcanic pipe. Magma is referred to as lava when it reaches the hole on the surface of earth and erupts out of it [14]. There are a number of stages leading to formation of a volcano that can be summarized as follows:

- 1) Rise of magma through cracks in the earth.
- 2) Build up of pressure.
- 3) Volcanic eruption and rise of magma to earth's surface.
- 4) Formation of a crust as a result of lava's cooling down.
- 5) Repetition of this process over time leading to several layers of rock that builds up over time resulting in a volcano.

In this section, volcano simulation as a new meta-heuristic and volcano eruption algorithm (VEA), is discussed. In the process of volcano eruption, mass of magma is needed at the first step of this process, so VEA starts with some solutions as initial population. In the volcano eruption process, magma rises through pipes, hence, similar idea is used to move some of solutions in different directions for certain determined distances. In the next step, all solutions will come down and move again in different directions just like the process of volcano eruption. Finally some of solutions in the "pipes", and points near the surface of earth, are exploded in the region of optimization programming problem. This steps comes from eruption of volcano at the top of the mountain into the space. Best solution of all populations will be found and the algorithm

is continued using this solution as the initial solution for the next iteration.

In general, the presented algorithm has been simulated based on the process of volcanic eruption from the formation of magma until explosion to the space. VEA is based on both solution and population. In the other words, the algorithm starts from a solution and tries to generate initial population near it, and as it progresses, it changes and modifies the population and set of solutions, in each iteration.

It should be noted, that each meta-heuristic is inspired from either the behavior of animals, insects, or certain natural events. Chemical pheromone of ants is the fundamental concept used for ant colony optimization; it is the direction and global best for particle swarm optimization; and light of fireflies for fire fly algorithm. Similarly, laying chicken algorithm is simulated based on warming of eggs as its main concept, and big bang algorithm focuses on explosion of particles. In the VEA algorithm presented below, the movement of magma from inside the ground to top of mountain and its explosion is the main concept for simulation of VEA.

III. THE VOLCANO ERUPTION ALGORITHM (VEA)

Details of VEA is discussed in this section. Mathematical equations; details of volcano eruption simulation; and various steps in the process of the algorithm to find the optimal solution in an optimization problem are outlined in an orderly fashion.

A. The Solutions and populations

Initial solution is created randomly and initial population is generated nearly the same as magma in the volcano eruption process. In fact, initial population in VEA displays the mass of magma deep inside the ground. In volcano eruption process magma is distributed in different directions through pipes (points near surface of the earth). This natural phenomenon point to an appealing approach as to how to distribute solution of initial population in different directions.

Each solution x_i in initial population is created randomly very near to initial solution x_0 using one of following probability distribution functions, also provided in MATLAB: a) Probability function of the binomial distribution (binomialPF); b) probability function of the geometric distribution (geometricPF); c) probability function of the hypergeometric distribution (hypergeometricPF); d) probability function of the Poisson distribution (poissonPF), and according to the following formula:

$$\|x_i - x_0\| \leq \epsilon \quad (1)$$

In R^n , $i=1,2,\dots,n$, ϵ is a small positive number.

In Figure 1, feasible solutions in the initial population (small blue points) and their movement in different directions have been shown for a given problem, the red large point on the figure is the optimal solution. Next population (black points), is distributed by random directions and formula (1) with $\epsilon=1$, $\epsilon=0.01$. Movement in different directions is based on the following equation:

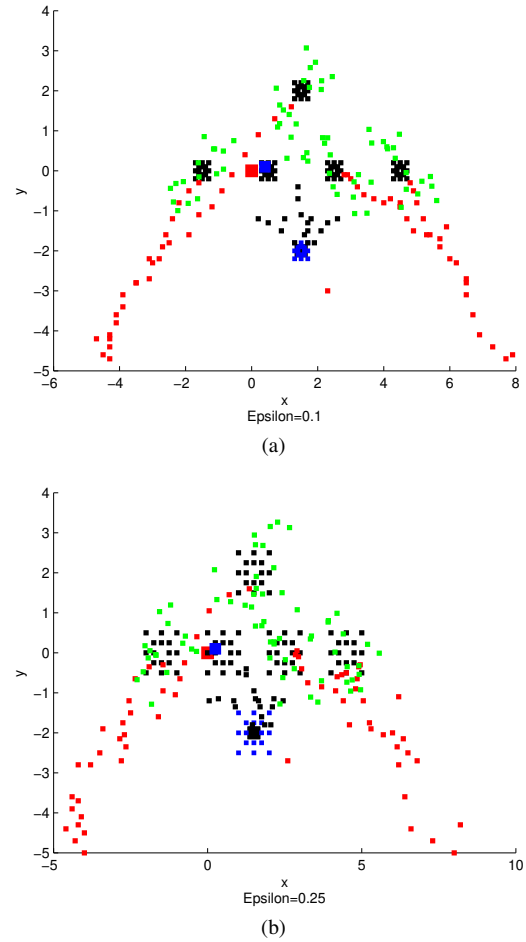


Fig. 2. Eruption and explosion of the population by different epsilon.

$$x_{j+1} = x_j + \lambda d_{rj} \quad (2)$$

d_{rj} is the j th random direction to up of the solution.

B. Explosion and eruption

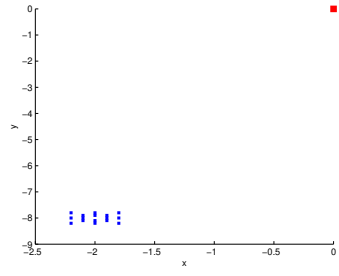
Solutions of the current population (black points in Figure 2) will be exploded (green points) and erupted (red points) in feasible space. This has been simulated based on explosion and eruption of volcano at the top of the mountain. In fact, solutions are changed in direction of the vector which connects solutions and the center solution of the population. These movements are according to the following equations:

$$x_{j+1} = x_i + \alpha d_{cj} \quad (3)$$

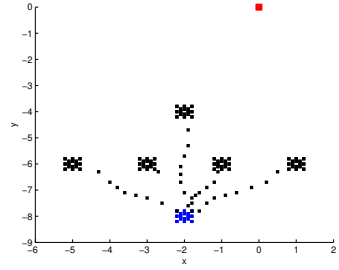
$$x_{j+1} = x_i - \beta d_{cj} \quad (4)$$

where d_{cj} is the vector which connects x_c , x_j and α, β are positive constants and x_c is the solution which comes from initial solution in previous direction to compose this population (center solution). Also, equation (3) represents the formulation of the explosion, and (4) represents the eruption.

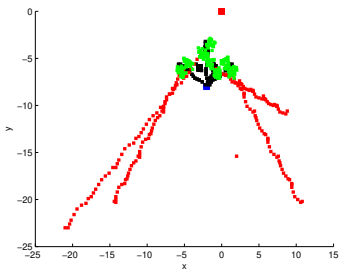
VEA finds the best solutions in all populations in this iteration. All population includes: initial population, second population which is created after movement of initial population in different directions (black points), third population which is generated after explosion using equation



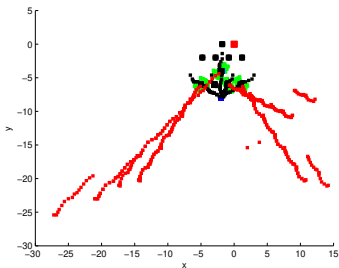
(a) Steps1,2 Initial Solutions and OS



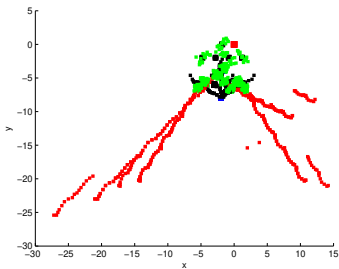
(b) Steps 3,4 Solutions Near Pipes



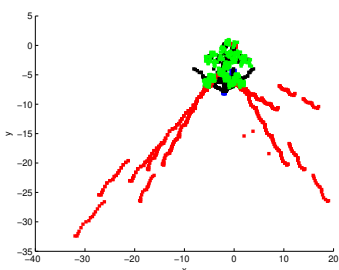
(c) Step 5 Eruption and Explosion



(d) Step 6 Best Solution, New Population



(e) Iteration 2 Explosion of Population



(f) Iteration 2 Optimal Solution

Fig. 3. Steps of the algorithm to solve a give problem.

(3) (green points), and finally the fourth population which is constructed after eruption and using equation (4) (red points). The best solution in this iteration is found and is shown by large blue point. The algorithm will be continued using the best solution as initial solution for the next iteration. This process in the first iteration has been shown in Figure 2 with $\epsilon=1$, $\epsilon=0.01$.

Algorithm 2 shows the pseudo code for this stage.

C. Algorithm

Pseudo code for this step is provided in Algorithm 1.

Algorithm 1 Pseudo code of solutions and populations

- 1: n: Number of solutions
 - 2: ϵ : A given small positive number
 - 3: Generate a random initial feasible solution
 - 4: Generate initial population near initial solution according to (1)
 - 5: **for** $i = 1$ **to** n **do**
 - 6: Generate solution of population based on (2)
 - 7: **end for**
-

Algorithm 2 Pseudo code of explosion and eruption

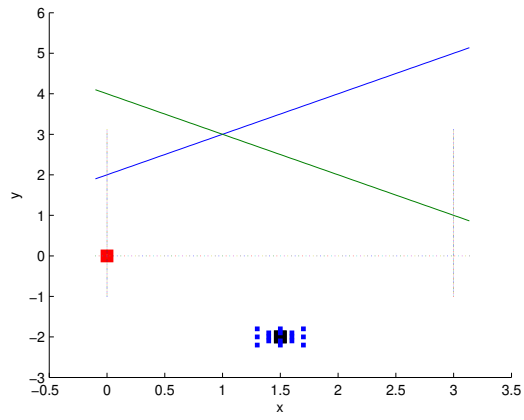
- 1: λ is given constant
 - 2: n is the number of initial population
 - 3: **for** $j = 1$ **to** n **do**
 - 4: $x_{j+1} = x_j + \lambda d$
 - 5: **end for**
 - 6: **for** $i = 1$ **to** n **do**
 - 7: $x_{i+1} = x_i + \lambda d$
 - 8: **end for**
-

The algorithm can be summarized in various steps as follows:

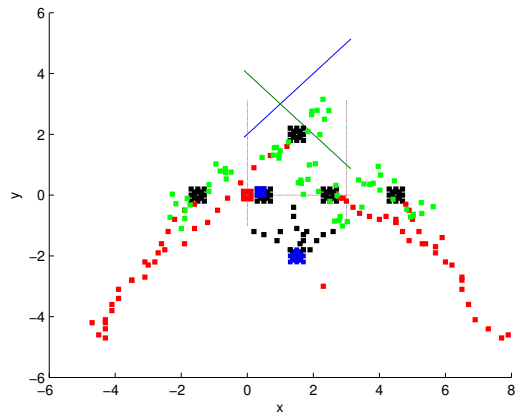
- 1) Initial solution will be generated randomly. It will be the origin for constrained problems (optional).
- 2) Initial population is generated near to the initial solution. Here ϵ is a given positive small number, $j=1$.
- 3) Solutions will be moved into different directions for a specific distance. (Until pipes)
- 4) New solutions near pipes will be generated.
- 5) a) Explosion of the solutions near pipes.
b) Coming down of solutions near pipes from different directions.
- 6) Find the best solution of the population. If $j < 2$, let $j=j+1$ and go to the step 2 with the best solution serving as the initial solution to the next population.
- 7) If $d(f(x_{j+1}), f(x_j)) < \epsilon$, then the algorithm will terminate and x_{j+1} is the best solution by VEA, x_j is the best solution in the j th iteration. Otherwise, let $j=j+1$ and go to step 2 where d is the following metric:

$$\max_i |f(x_{j+1}^i) - f(x_j^i)| = d(f(x_{j+1}), f(x_j)) \quad (5)$$

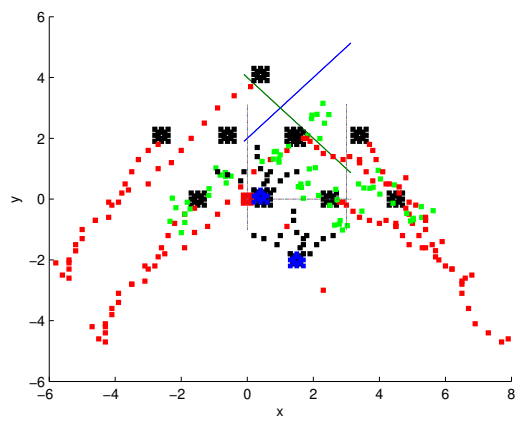
The following theorem proves that the proposed VEA algorithm as presented in the above steps is convergent.



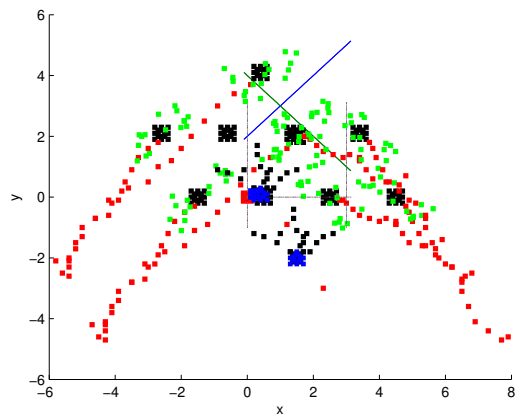
(a) Solutions near(1.5,-2) and optimal solution



(b) Generation 1

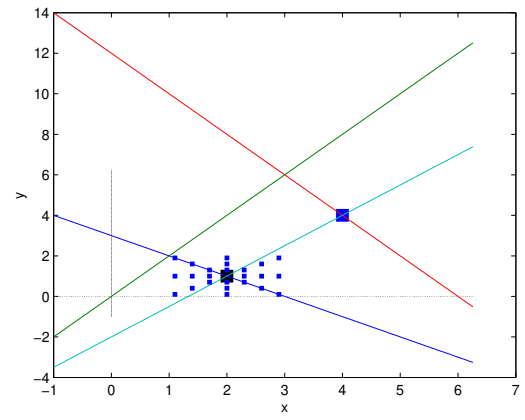


(c) Generation 2

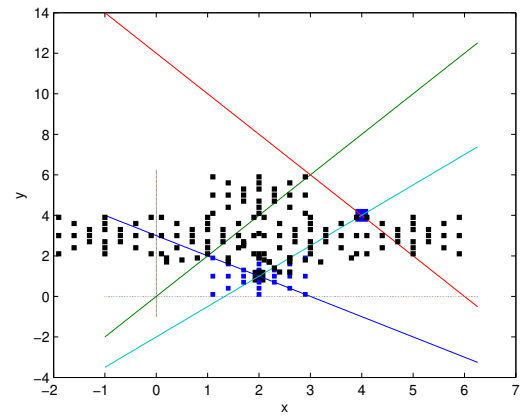


(d) Generation 2

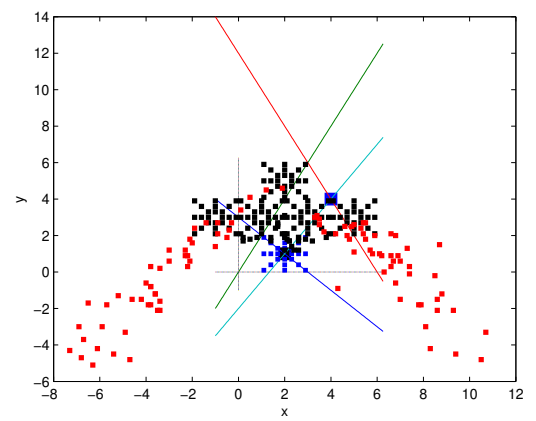
Fig. 4. Finding optimal solution by VEA- Example 1.



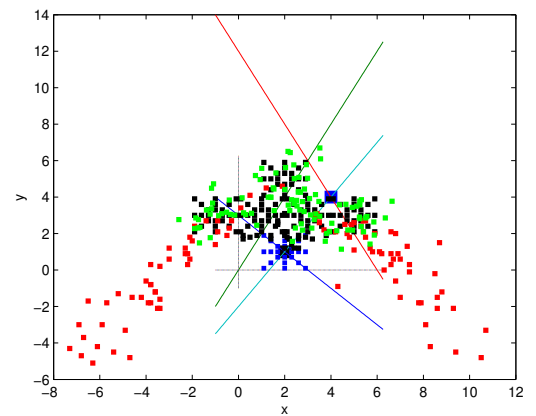
(a) Solutions near (2,1) and optimal solution



(b) Generation 1



(c) Generation 2



(d) Generation 2

Fig. 5. Process of finding optimal solution by VEA- Example 2.

Theorem 1: The sequence of F_k which was proposed in steps of VEA, [5], is convergent to the optimal solution.

Proof: Let

$$(F_v) = (F(t^v)) = (F(t_1^v), F(t_2^v), \dots, F(t_n^v)) = (F_1^{(v)}, F_2^{(v)}, \dots, F_n^{(v)})$$

According to step 6

$$\max |f(x_{j+1}^i) - f(x_j^i)| = d(f(x_{j+1}), f(x_j)) = d(F_{j+1}, F_j) < \epsilon_1$$

Therefore $|f(x_{j+1}^i) - f(x_j^i)|$ for each i . There is large number such as N which $k + 1 > k > N$ and $j = 1, 2, \dots, n$. Now we have:

$$|F_j^{(k+1)}, F_j^{(k)}| < \epsilon_1$$

Now let $m=k+1$, $r=k$ then we have:

$$|F_j^{(m)}, F_j^{(r)}| < \epsilon_1 \text{ For } m > r > n$$

This shows that for each fixed j , $1 \leq j \leq n$, the sequence $(F_j^{(1)}, F_j^{(2)}, \dots)$ is Cauchy of real numbers, then it converges, say to F_k . Using these n times, we define (F_1, F_2, \dots, F_n) and if $m=k+1$, $r=k$,

$$d(F_m, F_r) < \epsilon_1$$

Now if F_k we have

$$d(F_m, F_r) \leq \epsilon_1$$

This shows that F is the limit of F_m and the sequence is convergent, hence proving the Theorem.

Figure 3 shows the steps and progress of the algorithm in finding the optimal solution R^2 .

D. Mathematical nature of the algorithm

Mathematical nature of the algorithm is based on four phases:

- 1) Generation of feasible initial solution and population.
- 2) Movement of solutions to improve population and reaching better solutions.
- 3) Termination of the algorithm to propose the best solution.
- 4) Convergent of the algorithm.

Firstly, a feasible solution is created randomly in the feasible region. So to produce feasible initial population, it is generated near enough to initial solution based on the following formula: $\sqrt{(x_{i1} - x_{01})^2 + (x_{i2} - x_{02})^2 + \dots + (x_{in} - x_{0n})^2} \leq \epsilon$ In this phase, the algorithm tries to move solutions of initial population in different random directions to have high chance for finding better solutions. This movements is based on equation (2).

In the second phase, explosion and eruption of volcano is simulated by going up and then coming down based on equations (3) and (4).

In the third phase, termination of the algorithm is based on the following condition:

If $d(f(x_{j+1}), f(x_j)) = \text{Max}|f(x_{j+1}^i) - f(x_j^i)| < \epsilon$ then the algorithm will be finished and x_{j+1} is the best solution by VEA x_j is the best solution in j th iteration.

Finally, convergence feature of VEA has been proven by the above condition and theorem 1.

IV. COMPUTATIONAL RESULTS

To show efficiency of the algorithm, two classes of optimization problems are considered and solved: a) continuous problems with small size, and b) discrete and practical problems with large size. Then VEA is used to solve internet of vehicles problem.

A. Continuous Problems

In this subsection, almost all kinds of continuous optimization problems: constrained, unconstrained, linear, non-linear, multi-level and multi-objective will be solved.

Example 1 [15](Constrained - Non-linear)

Consider the following non-linear programming problem:

$$\begin{aligned} \min & -(x_1 - 4)^2 - (x_2 - 4)^2 \\ & \text{subject to} \\ & x_1 - 3 \leq 0 \\ & -x_1 + x_2 - 2 \leq 0 \\ & x_1 + x_2 - 4 \leq 0 \\ & x_1, x_2 \geq 0 \end{aligned} \quad (6)$$

Comparison between VEA and other classical methods in references is shown in Table I.

In Figure 4, the initial and optimal solutions and different populations of the algorithm for Example 1 is shown. The large blue point in Figure 4 is the optimal solution which has been found by solutions after 2 iterations.

Example 2 [16] (Multi-Level)

Consider the following linear bi-level programming problem:

$$\begin{aligned} \min & x - 4y \\ & \text{subject to} \\ & \min y \\ & \text{subject to} \\ & x + y \geq 3 \\ & -2x + y \leq 0 \\ & 2x + y \leq 12 \\ & 3x - 2y \leq 4 \\ & x, y \geq 0 \end{aligned} \quad (7)$$

Using KKT conditions the problem will be converted into the

TABLE I
COMPARISON OF VEA AND OTHER METHODS - EXAMPLE 1

| Algorithms | N. Agents | N. Iterations | Optimal Solution | F Min | ϵ | Initial Soltion |
|--------------------|-----------|---------------|------------------|--------|------------|-----------------|
| VEA | 16 | 1 | (0.4,0.09) | -28.17 | 1 | (1.5,-2) |
| VEA | 16 | 2 | (0.2,0.09) | -29.65 | 1 | (1.5,-2) |
| Other Methods [15] | | | (0,0) | -32 | | |

following problem:

$$\begin{aligned}
& \min x - 4y \\
& \text{subject to} \\
& -\lambda_1 + \lambda_2 + \lambda_3 - 2\lambda_4 = -1 \\
& \lambda_1(-x - y + 3) = 0 \\
& \lambda_2(-2x + y) = 0 \\
& \lambda_3(2x + y - 12) = 0 \\
& \lambda_4(3x - 2y - 4) = 0 \\
& -x - y + 3 \leq 0 \\
& -2x + y \leq 0 \\
& 2x + y - 12 \leq 0 \\
& 3x - 2y - 4 \leq 0 \\
& x, y, \lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0
\end{aligned} \tag{8}$$

The bi-level programming problem is NP-Hard because of its two objective functions. In fact, these two objective functions should be optimized in two different levels at the same time. Therefore, proposing a solution for this problem is significant. VEA proposed the optimal solution same as exact algorithms according to Table II. Number of iterations to find the optimal solution is completely down.

Behavior of solutions, constraints of the problem and optimal solution have been shown in Figure 5.

Example 3 [20] (Multi-Objective)

Here VEA is used for solving DTLZ benchmark problems [21]. Behavior of the algorithm in finding the pareto optimal for DTLZ1 problem is shown in Figure 6. The fact that some of solutions in the population have reached to pareto optimal solution, illustrates the feasibility of the algorithm as shown in Figures 6-c, 6-d, which also indicate the initial population. Also, efficiency of the algorithm is obvious by comparison of Figure 6-a, 6-f. Most of solutions are completely far from pareto optimal at first, but during the process, the algorithm solutions achieve pareto optimal. Figure 6-f shows that the last population has surrounded pareto optimal solutions.

Table III shows the comparison of best solutions to get Pareto optimal of DTLZ problems by VEA and the best method in [21].

Example 4

Consider Rastrigin Function (RF) which is a non-convex optimization problem.

$$\min 20 + (x^2 - 10\cos(2\pi x)) + (y^2 - 10\cos(2\pi y)) \tag{9}$$

Figure 7 shows the process of finding optimal solution by VEA for Example 4.

Example 5

Consider Ackley Function the (AF):

$$\min -20\exp(-0.2\sqrt{0.5(x^2 + y^2)}) - \exp(0.5(\cos(2\pi x) + \cos(2\pi y))) + e \tag{10}$$

Figure 8 shows the process of finding optimal solution and the results of it respectively.

Example 6

Consider Hölder Table Function (HTF):

$$\min -|\sin(x)\cos(y)\exp(|1 - \sqrt{(x^2 + y^2)}/\pi|)| \tag{11}$$

The problem has been solved by VEA, the process of the algorithm, initial population, optimal solution of generations and constraints of the problems have been shown for two iterations in Figure 9.

Example 7

Consider Mishra's Bird Function (MBF):

$$\min \sin(x)\exp((1 - \cos(y))^2) + \cos(y)\exp((1 - \sin(x))^2) + (x - y)^2 \tag{12}$$

The problem has been solved by MVA, the process of the algorithm, initial population, optimal solution of generations and constraints of the problems have been shown for two iterations in Figure 10.

B. Large size practical problems

To show efficiency of the algorithm for real life/size problems, in this section three kinds of practical problems have been solved: large size real linear programming problems, transportation problems and internet of vehicles problems.

Some benchmark linear programming have been solved and the results of Table IV indicates VEA can solve large size problems. Through a comparison with "Linprog", the Matlab tool for solving linear programming problems, the superiority of VEA has been demonstrated.

Absolute error of "Linprog" and VEA from the optimal solution in Table V indicates that the classic method is completely impractical and VEA is strongly efficient.

Finding a suitable feasible solution of transportation problem is remarkable, so VEA was applied to some random transportation problems [3]. The results are shown in Table VI.

In Table VII, comparison with Vogel algorithm, which is the best algorithm to find feasible solutions of transportation problem, confirms the superiority of the proposed VEA.

TABLE II
COMPARISON OF VEA AND OTHER METHODS (O.M)- EXAMPLE 2

| Algorithms | N. Agents | N. Iterations | Optimal Solution | Solu- tion | F Min | ϵ | Initial Soltion |
|--------------------|-----------|---------------|------------------|------------|-------|------------|-----------------|
| VEA | 24 | 1 | (3.4,3.1) | | -9 | 1 | (2,1) |
| VEA | 24 | 2 | (4,4) | | -12 | 1 | (2,1) |
| Other Methods [16] | | | (4,4) | | -12 | | |
| Other Methods [16] | | | (3.9,4) | | -12.1 | | |

TABLE III
COMPARISON OF VEA AND OTHER METHODS FOR DTLZ PROBLEMS

| Problems | k | ParEGO | | | VEA | | |
|----------|----|--------|-------|-------|-------|--------|--------|
| | | min | mean | max | min | mean | max |
| DTLZ1 | 3 | 13.42 | 52.47 | 112.7 | 9.13 | 31.24 | 78.18 |
| DTLZ1 | 4 | 18.63 | 45.45 | 87.76 | 11.57 | 32.21 | 59.32 |
| DTLZ1 | 10 | NA | NA | NA | 1.12 | 1.78 | 2.95 |
| DTLZ2 | 3 | 0.151 | 0.191 | 0.243 | 0.093 | 0.105 | 0.164 |
| DTLZ2 | 4 | 0.289 | 0.337 | 0.408 | 0.099 | 0.187 | 0.275 |
| DTLZ2 | 10 | NA | NA | NA | 0.081 | 0.123 | 0.187 |
| DTLZ3 | 3 | 81.15 | 145.5 | 261.6 | 52.56 | 123.26 | 213.77 |
| DTLZ3 | 4 | 66.93 | 138.1 | 209.4 | 43.32 | 107.41 | 186.24 |
| DTLZ3 | 10 | NA | NA | NA | 0.85 | 1.14 | 1.96 |

TABLE IV
RESULTS OF VEA FOR MORE TEST PROBLEMS

| Name | Size | Optimal | Linprog | VEA | N. Iterations |
|----------------|-----------|-------------------|---------------|---------------|---------------|
| agg | 489 163 | -3.5991767287E+07 | -3.9217e+16 | - 3.59917e+07 | 10 |
| qap8 | 913 1632 | 2.0350000000E+02 | -1.6987e+16 | 2.144e+02 | 20 |
| SC50A | 51 48 | -6.4575077059E+01 | -6.5313e+20 | -6.4879e+01 | 5 |
| AFIRO | 28 32 | -4.6475314286E+02 | -1.4505e+29 | -4.7361e+02 | 5 |
| Random Problem | 1000 5000 | | -400.6831e+36 | -124.3891e+07 | 500 |

TABLE V
COMPARISON ERRORS OF VEA AND CLASSIC METHODS

| Name | Error of Linprog | Error of VEA |
|-------|------------------|--------------|
| agg | 3.9217e+16 | 67 |
| qap8 | 1.6987e+16 | 11 |
| SC50A | 6.5313e+20 | 0.3 |
| AFIRO | 1.4505e+29 | 8.9 |

C. Route Optimization Design in Internet of Vehicles environment

The objective of this problem is to maximize the connectivity probability and link quality of the available routes from source to destination as illustrated in Fig. 11 [19]. The maximization process is subject to Signal to Interference and Noise Ratio threshold ($SINR_{th}$) in order to find more reliable and connected route in urban SDN based vehicular scenarios. The city road networks in vehicular scenario is represented as graph model $G(i,e)$ where i is an intersection and e is the road segment between two intersections. Therefore, each optimal route ζ consists of a set of intersections $(i_1, i_2, i_3, i_4, i_5, i_6, \dots, i_m)$ and a set of streets $(e_1, e_2, e_3, e_4, e_5, e_6, \dots, e_n)$, where $n=m-1$. According to

the aforementioned assumptions, the objective function of the optimization problem can be written as:

$$\max_{\zeta} F(\zeta) = \lambda_1 \times PC(\zeta) + \lambda_2 \times SINR(\zeta) \quad (13)$$

$$\text{where } PC(\zeta) = \prod_{i=1}^n PC(e_i),$$

$$SINR(\zeta) = \frac{\sum_{i=1}^n SINR(e_i) - \sum_{i=1}^n SINR_{th}(e_i)}{\sum_{i=1}^n SINR(e_i)}, \quad (14)$$

subject to

$$SINR(\zeta) \geq SINR_{th}(\zeta). \quad (15)$$

TABLE VI
COMPARISON AMONG VEA AND OTHER ALGORITHMS FOR LARGE SIZE PROBLEMS

| Problems | Size | North-West | Vogel | VEA |
|------------------|-------------|------------|----------|---------|
| Transportation 1 | 80 20 | 132804 | 30123 | 22150 |
| Transportation 2 | 100 25 | 177666 | 26462 | 24367 |
| Transportation 3 | 160 40 | 185366 | 85456 | 62859 |
| Transportation 4 | 200 50 | 297629 | 26566 | 21578 |
| Transportation 5 | 210 70 | 322356 | 27619 | 23160 |
| Transportation 6 | 261 87 | 245311 | 152930 | 119526 |
| Transportation 7 | 10000 10000 | 12736903 | 10321697 | 5896123 |

TABLE VII
IMPROVEMENT AMOUNT OF VOGEL ALGORITHM BY VEA

| Problems | Size | Vogel | VEA | Improvement by VEA |
|------------------|--------|--------|--------|--------------------|
| Transportation 1 | 80 20 | 30123 | 22150 | 0.26 |
| Transportation 2 | 100 25 | 26462 | 24367 | 0.08 |
| Transportation 3 | 160 40 | 85456 | 62859 | 0.26 |
| Transportation 4 | 200 50 | 26566 | 21578 | 0.19 |
| Transportation 5 | 210 70 | 27619 | 23160 | 0.16 |
| Transportation 6 | 261 87 | 152930 | 119526 | 0.22 |

where $F(\zeta)$ is defined as the objective function with a set of routes ζ from source to destination. λ_1 and λ_2 are the weights that empirically set in the simulation and their summation is equal to 1. $PC(\zeta)$ and $SINR(\zeta)$ connectivity and reliability of routes respectively. $PC(\zeta)$ and $SINR(\zeta)$ connectivity and reliability of routes respectively. $PC(e_i)$ and $SINR(e_i)$ representing the street's connectivity and link reliability. Fig. 11 illustrates the routing process in SDIoV.

This problem is addressed by both mathematical and heuristic algorithms. Laying Chicken Algorithm (LCA) [12] has been used to find optimal route from source to destination [19]. The comparison of results of LCA and VEA are provided in Table VIII.

For each problem an initial solution has been generated randomly and these initial solution are different for both LCA and VEA algorithms. Table IX shows improvement of their initial solutions after five iterations.

V. CONCLUSION

A novel and new meta-heuristic optimization algorithm inspired from a natural phenomenon as opposed to commonly considered behavior of creatures was proposed in this article. VEA is an easy to simulate algorithm which can optimize all kinds of optimization problems.

Numerical results presented in this paper show that the proposed approach can solve unconstrained and constrained, linear and nonlinear, single level and multi level problems with small and large feasible regions very effectively and more efficiently.

The following briefly outline areas for future work in this exciting area:

- 1) Explore the possibility of solving some NP hard problems such as travelling salesman problem using the proposed VEA.

- 2) VEA should be attempted to solve big data because it has appropriate complexity

- 3) The algorithm should be extended for solving discrete problems such as shortest path problem, etc.

- 4) Combination of proposed algorithm as an inspired approach with exact methods. For example finding an approximate gradient vector by VEA for using methods such as simplex which use gradient directly.

- 5) Implementation of similar ideas like floods, hurricanes, earthquakes and so on.

VI. CONFLICT OF INTEREST

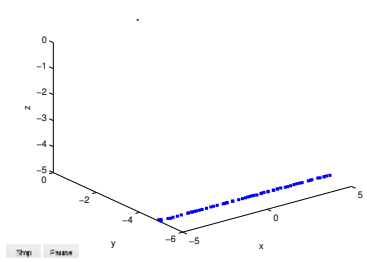
The author certifies that following details of affiliation or involvement in an organization or entity with a financial or non-financial interest in the subject matter or materials discussed in this manuscript.

TABLE VIII
COMPARISON OF LCA AND VEA FOR INTERNET OF VEHICLES

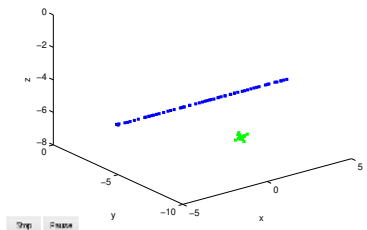
| Problems | Size | Best Solution by LCA | Best Solution by VEA |
|-----------------|-------------|-----------------------------|-----------------------------|
| IoV 1 | 100 100 | 775.8550 | 917.3405 |
| IoV 2 | 200 200 | 9.9319e+03 | 1.3014e+04 |
| IoV 3 | 500 500 | 5.8147e+04 | 6.9372e+04 |
| IoV 4 | 1000 1000 | 2.5991e+05 | 2.8461e+05 |
| IoV 5 | 2000 2000 | 9.8622e+05 | 1.2831e+06 |
| IoV 6 | 5000 5000 | 6.2266e+06 | 6.6281e+06 |
| IoV 7 | 10000 10000 | 2.4950e+07 | 2.7145e+07 |
| IoV 8 | 30000 30000 | 3.7632e+09 | 3.7916e+09 |

TABLE IX
IMPROVEMENT OF VEA AND LCA FROM THEIR RANDOM INITIAL SOLUTIONS (RIS) IN FIVE ITERATIONS

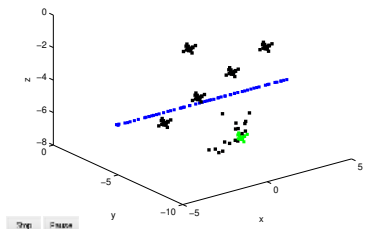
| Problems | Size | Improvement of RIS by LCA | Improvement of RIS by VEA |
|-----------------|-------------|----------------------------------|----------------------------------|
| IoV 1 | 100 100 | 0.031 | 0.221 |
| IoV 2 | 200 200 | 0.005 | 0.318 |
| IoV 3 | 500 500 | 0.008 | 0.202 |
| IoV 4 | 1000 1000 | 0.002 | 0.097 |
| IoV 5 | 2000 2000 | 0.002 | 0.303 |
| IoV 6 | 5000 5000 | 0.001 | 0.064 |
| IoV 7 | 10000 10000 | 0.0001 | 0.081 |



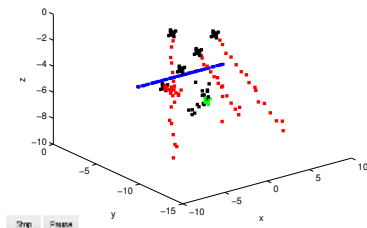
(a) Solutions near (1,-1) and OS



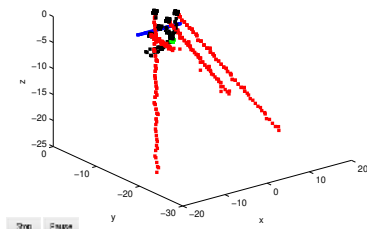
(b) Generation 2



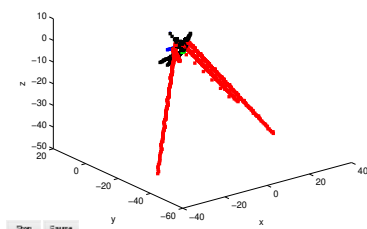
(c) Generation 2 and optimal solution



(d) Generation 2

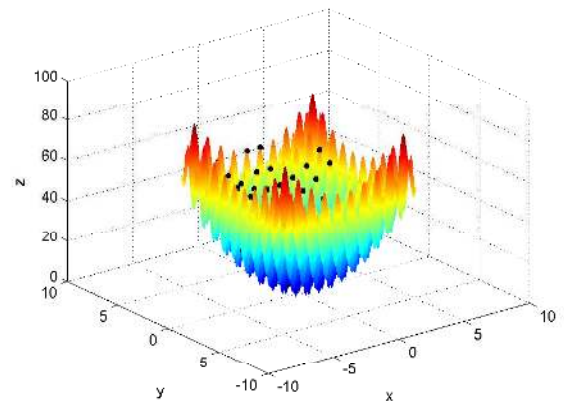


(e) Generation 2

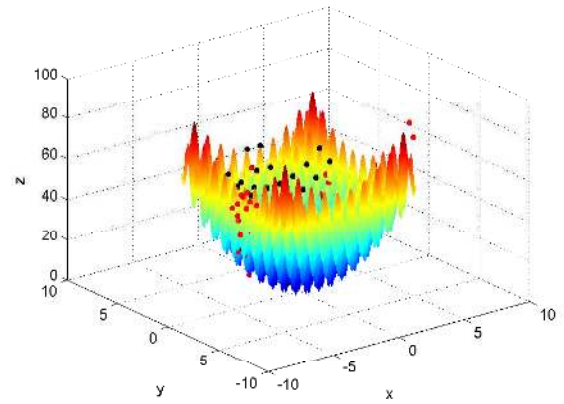


(f) Iteration 2 Optimal Solution

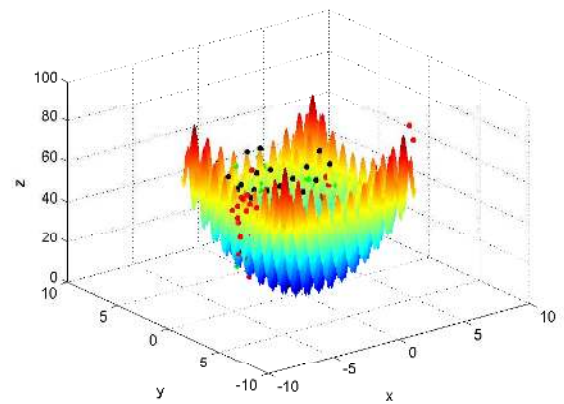
Fig. 6. Generations of VEA to find optimal solution - Example 3.



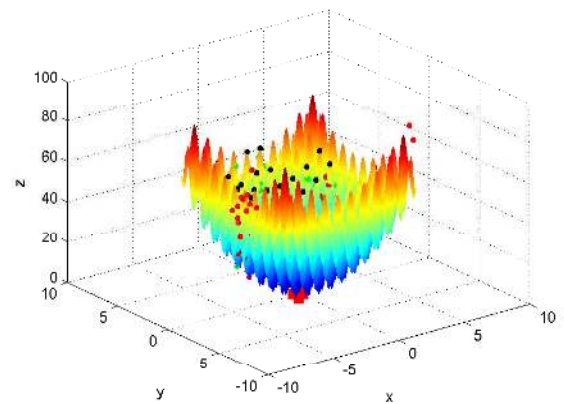
(a) Initial population



(b) Generation 2

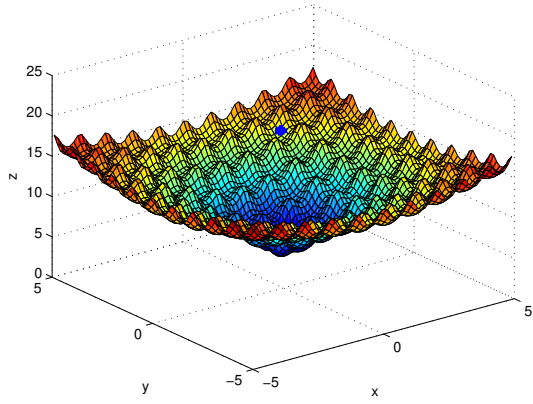


(c) Generation 4 and optimal solution for BLPP

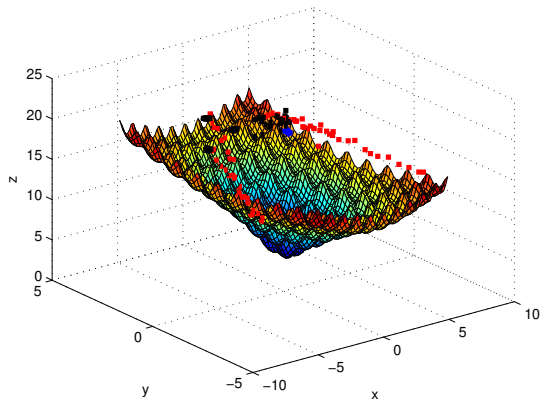


(d) Generation 4 and optimal solution for BLPP

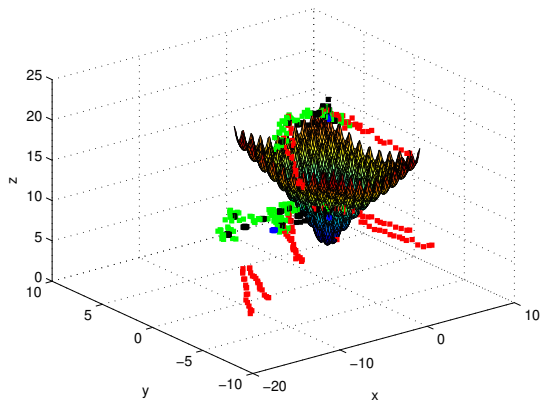
Fig. 7. Process of finding optimal solution by VEA- Example 4



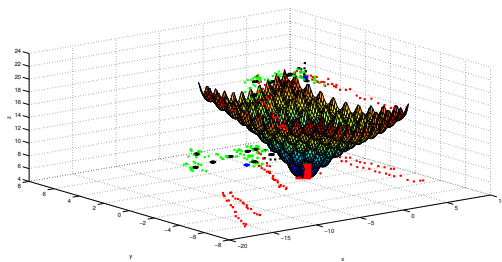
(a) Initial population



(b) Generation 2

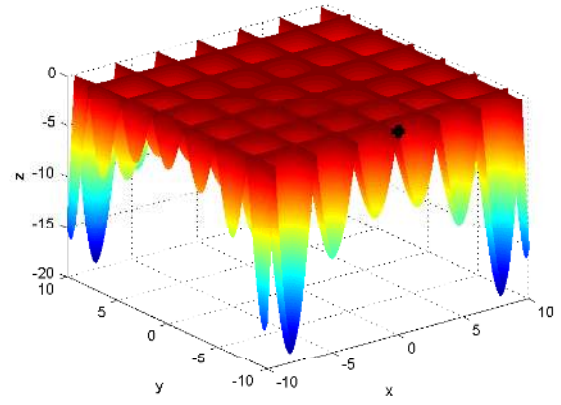


(c) Generation 4 and optimal solution for BLPP

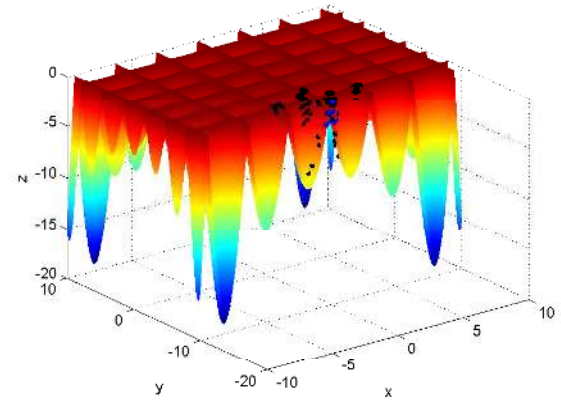


(d) Generation 4 and optimal solution for BLPP

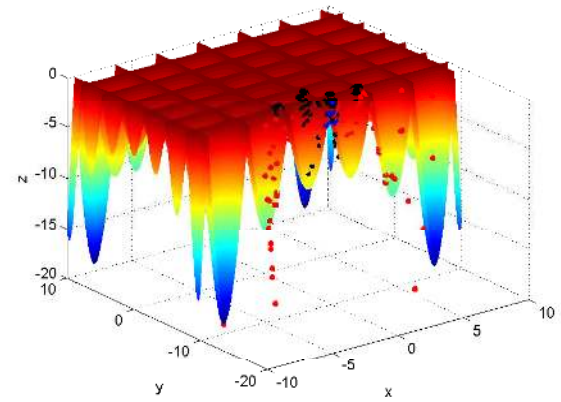
Fig. 8. Process of finding optimal solution by VEA- Example 5



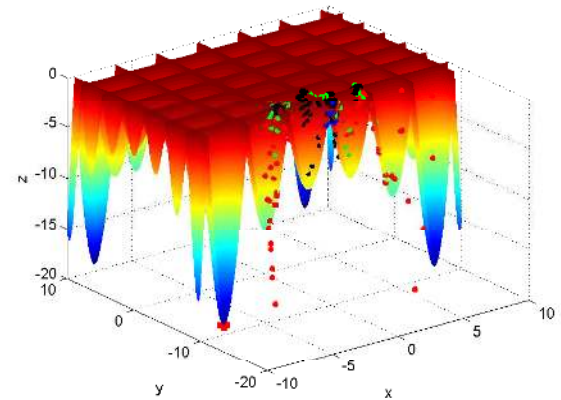
(a) Initial population



(b) Generation 2

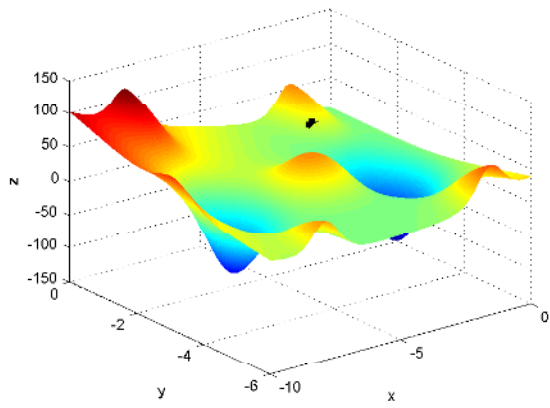


(c) Generation 4 and optimal solution for BLPP

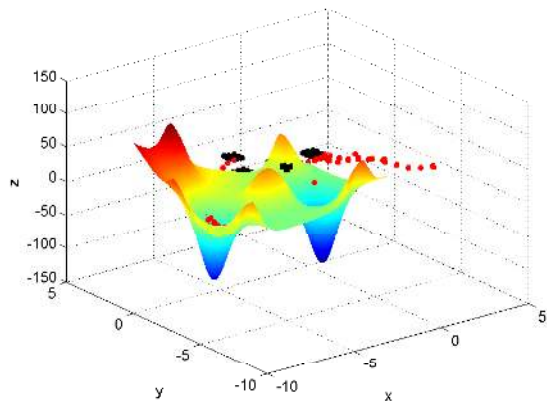


(d) Generation 4 and optimal solution for BLPP

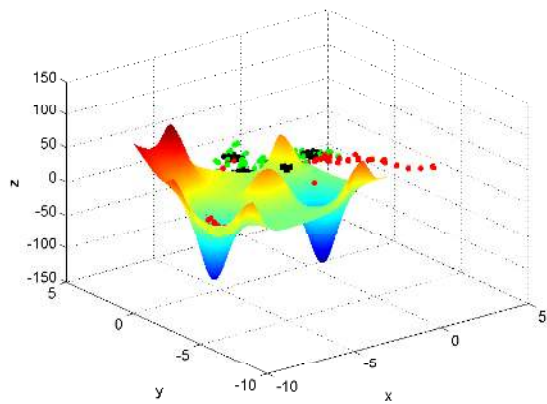
Fig. 9. Process of finding optimal solution by VEA- Example 6



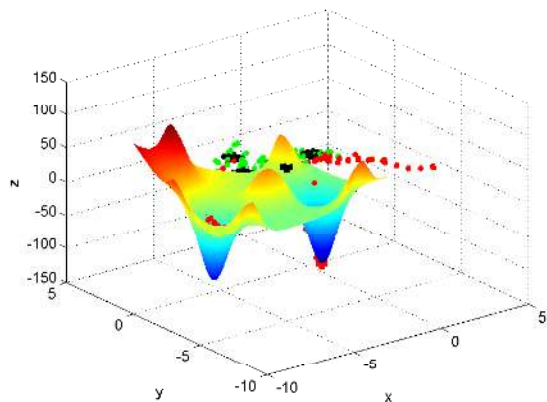
(a) Initial population



(b) Generation 2



(c) Generation 4 and optimal solution for BLPP



(d) Generation 4 and optimal solution for BLPP

Fig. 10. Process of finding optimal solution by MV- Example 7

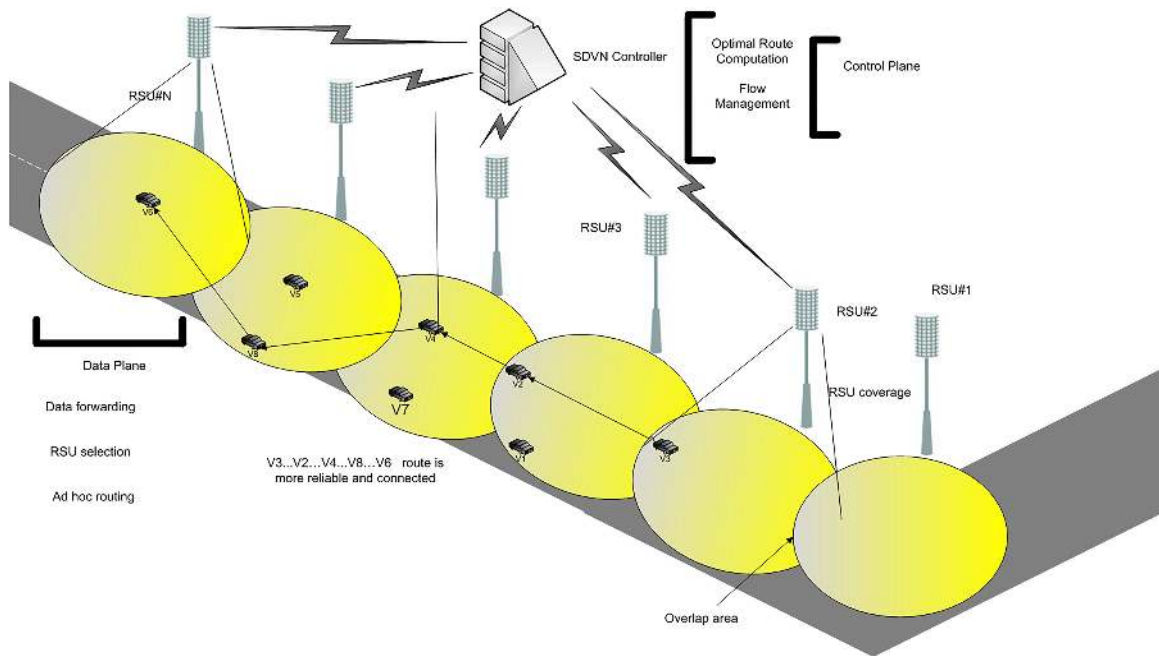


Fig. 11. Optimal routing process in IoV environment

REFERENCES

- [1] J. Kennedy and R. C. Eberhart. *Particle Swarm Optimization*. In *Proceedings of IEEE*. International Conference on Neural Networks, Perth, Australia, pp. 1942–1948, (1995).
- [2] Eghbal Hosseini. *Presentation and Solving Non-Linear Quad-Level Programming Problem Utilizing a Heuristic Approach Based on Taylor Theorem*.
- [3] Eghbal Hosseini. *Solving Linear Tri-level Programming Problem Using Heuristic Method Based on Bi-section Algorithm*. Asian Journal of Scientific Research 10 (4) (2017) .
- [4] Eghbal Hosseini. *Three new methods to find initial basic feasible solution of transportation problems*. Applied Mathematical Sciences 11 (no. 37.), 1803-1814. (2017).
- [5] D. Karaboga and B. Basturk. *A Powerful And Efficient Algorithm For Numerical Function Optimization: Artificial Bee Colony (ABC)*. Journal of Global Optimization, Springer Netherlands, Vol. 39, No. 3, pp: 459-471, (2007).
- [6] Gandomi, A.H., Alavi, A.H. *Krill herd: A new bio-inspired optimization algorithm*. Communications in Nonlinear Science and Numerical Simulation 17, 4831–4845 (2012).
- [7] Yang, X.S. *Bat algorithm: literature review and applications*. International Journal of Bio-inspired Computation 5(3), 141–149 (2013).
- [8] Cuevas, E., Cienfuegos, M., Zaldivar, D., Cisneros, M. *A swarm optimization algorithm inspired in the behavior of the social-spider*. Expert Systems with Applications 40, 6374–6384 (2013).
- [9] Xianbing Meng, Yu Liu, Xiaozhi Gao, and Hengzhen Zhang. *A New Bio-inspired Algorithm: Chicken Swarm Optimization*. ICSI 2014, Part I, LNCS 8794, pp. 86–94 (2014).
- [10] Xin She Yang. *Nature- Inspired Meta-Heuristic Algorithms*. University of Cambridge 2010.
- [11] Eghbal Hosseini. *Big Bang Algorithm: A New Meta-heuristic Approach for Solving Optimization Problems*. Asian Journal of Applied Sciences 10 (4), 334-344 (2017).
- [12] Eghbal Hosseini. *Laying Chicken Algorithm: A New Meta-Heuristic Approach to Solve Continuous Programming Problems*. Journal of Applied and Computational Mathematics 6 (1) (2017).
- [13] Eghbal Hosseini, IN Kamalabadi. *A Genetic Approach for Solving Bi-Level Programming Problems*. Advanced Modeling and Optimization 15 (3) (2013).
- [14] Eghbal Hosseini, IN Kamalabadi. *Line search and genetic approaches for solving linear tri-level programming problem*. International Journal of Management, Accounting and Economics 1 (4) (2015).
- [15] JT Salzer, WA Thelen, MR James, TR Walter, S Moran, R Denlinger. *Volcano dome dynamics at Mount St. Helens: Deformation and intermittent subsidence monitored by seismicity and camera imagery pixel offsets*. Journal of Geophysical Research: Solid Earth 121 (11), 7882-7902 (2016).
- [16] M. Bazzara. *non-linear programming Theory and Algorithms*. Wiley, Inc, New York (2007).
- [17] Eghbal Hosseini. , Isa Nakhai Kamalabadi *Solving Linear Bi-level Programming Problem Using Two New Approaches Based on Line Search and Taylor Methods*. Management sciences and Education, Vol.2 (6), 243-252 November (2014).
- [18] M. Bazzara. *linear programming and Network Flows*. Wiley, Inc, New York (2010).
- [19] Ghafoor Kayhan; Kong Linghe ; Danda Rawat; Hosseini Eghbal ; Sadiq Ali. *Quality of Service Aware Routing Protocol in Software-Defined Internet of Vehicles*. IEEE Internet of Things Journal, October (2018).
- [20] K. Deb;L. Thiele; M. Laumanns; E. Zitzleri. *Scalable multi-objective optimization test problems*. in Proc. IEEE Congr. Evol. Comput., Honolulu, HI, USA, pp. 825–830(2002).
- [21] Chugh, Tinkle; Jin, Yaochu; Miettinen, Kaisa; Hakanen, Jussi; Sindhya, Karthik. *A Surrogate-assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-objective Optimization*. IEEE Transactions on Evolutionary Computation, 22 (1), 129-142 (2018).
- [22] Mirjalili, S; Dong, J. S.; Lewis, A.; Sadiq, A. S. *Particle Swarm Optimization: Theory, Literature Review, and Application in Airfoil Design*. In Nature-Inspired Optimizers, (pp. 167-184), Springer, Cham (2020).
- [23] Mirjalili, S; Lewis, A.; Sadiq, A. S. *Autonomous particles groups for particle swarm optimization*. Arabian Journal for Science and Engineering, 39(6), 4683-4697 (2014).
- [24] Shahjehan, Waleed; Riaz, Asad; Khan, Imran; Sadiq, Ali Safaa; Khan, Suleman; Khan, Muhammad Khurram. *BAT Algorithm based Beam-forming for mmWave Massive MIMO Systems*. International Journal of Communication Systems, DOI: 10.1002/dac.4182 (2019).
- [25] Mafarja, M; Mirjalili, S. *Whale optimization approaches for wrapper feature selection*. Applied Soft Computing, 62, 441-453. (2018).
- [26] Javidy, B; Hatamlou, A; Mirjalili, S. *Ions motion algorithm for solving optimization problems*. Applied Soft Computing, 32, 72-79 (2015).
- [27] Mirjalili, S; Mirjalili, S; M.; Lewis, A. *Grey wolf optimizer*. Applied Soft Computing, 69, 46-61 (2014).
- [28] Mirjalili, S; Gandomi, A. H. *Chaotic gravitational constants for the gravitational search algorithm*. Applied Soft Computing, 53, 407-419 (2017).
- [29] Sadiq, A. S.; Faris, H.; Ala’M, A. Z.; Mirjalili, S.; Ghafoor, K. Z. *Fraud Detection Model Based on Multi-Verse Features Extraction Approach for Smart City Applications*. In Smart Cities Cybersecurity and Privacy (pp. 241-251). Elsevier (2019).
- [30] Mirjalili, S.; Dong, J. S.; Sadiq, A. S.; Faris, H. *Genetic Algorithm: Theory, Literature Review, and Application in Image Reconstruction*. In Nature-Inspired Optimizers (pp. 69-85). Springer, Cham (2020).
- [31] Claudio Gentile, Shuai Li. *Collaborative Filtering Bandits*. the 39th International ACM SIGIR conference (2015).
- [32] Nathan Korda, Balazs Szorenyi, Shuai Li. *Distributed Clustering of Linear Bandits in Peer to Peer Networks*. Proceedings of the 33 rd International Conference on Machine Learning, New York, NY, USA (2016).
- [33] Fei Hao, Doo-Soon Park, Shuai Li. *Mining Maximal Cliques from a Fuzzy Graph*. Sustainability, 8(6), 553, (2016).
- [34] Arikrishna Narasimhan, Shuai Li, Purushottam Kar, Sanjay Chawla, Fabrizio Sebastiani. *Stochastic Optimization Techniques for Quantification Performance Measures*. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016).