Velella Swarm Optimization Algorithm (VSO)

Asgar Jangah
Shahid Bahonar University of Kerman
Hamid Mirvaziri (h.vaziri@gmail.com)
Shahid Bahonar University Faculty of Engineering https://orcid.org/0000-0003-2863-3750

Research Article

Keywords: Velella, Sailboat, Optimization, Heuristic search algorithms, Venom

Posted Date: February 11th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1320069/v1

Velella Swarm Optimization Algorithm (VSO)

1.Asgar Jangah, 2. Hamid Mirvaziri*

Computer Engineering Department, Shahid Bahonar University of Kerman. Email:hmirvaziri@uk.ac.ir

A B S T R A C T

Some algorithms have been developed in heuristic optimization area inspired by swarm behaviour in nature which are the most popular optimization problems. In this paper, a theory for subgroups of Velella are introduced and a new biologically-inspired optimization algorithm is proposed based on subgroups behaviour of the individual Velella in colonies. Our proposed method is compared with some of the state-of-the-art heuristic optimization algorithms. Finally, performance and results of the proposed method is prepared for solving various benchmark functions. Obtained results prove that our method can solve same problems significantly better than the others.

Keywords: Velella, Sailboat, Optimization, Heuristic search algorithms, Venom

1. Introduction

Many collective behaviors are exhibited by social insects and animals as well as swarm's phenomena in nature which demonstrates their intelligent characteristics. Swarm Systems benefit from swarm intelligence (SI) to solve existed challenges [1]. An SI system is typically consist a population of individuals inspired by natural phenomena. Interaction rules between individuals or agents show especial patterns and behavior as SI systems. Birds flying in flocks, ants' path-planning behavior and honey bee waggle dances which are all performed to find food sources and interactions of human society to learn from each other are a few biological examples that exhibit multiple necessary behavior for success of colony in SI systems. [2], [3].

Many phenomena can be observed in nature and velella animal is one of these remarkable one getting advantage of SI system that emerge subgroup behavior.

The velella are floating on the surface of Open Ocean. They have a purple color with a flat elliptical transparent float and an erect sail projecting a vertical at an angle to the axis of its body. They feed on small prey and fish by their venom that can be caught below the surface of the water. The wind-based projection of the sail, water flow and swimming by its tentacles can take the best advantage of movement at any given moment.

The velella feeding is limited to the surface of water because it is not big enough to cover a large surface of water by venom and its tentacles do not sense far distances. Venom establish trap in a radius for bait and velella sense the food caught in trap by its tentacles. Their mouth is located in the middle of the underside of body and feed the bait by shaking their tentacles [4].

Older zoological opinions assumed velella was a colony of specialized individuals. Some taxonomists classified velella as Siphonophore because their view did not consider velella to be a single animal but they are organisms which were linked together with their interactions. (A colony of specialized polyps). However, more recent studies, have classified the velella as a highly modified individual hydroid polyps, but not a colonial hydrozoan. Because it is believed that velella is a single highly modified polyp. For example, Research playwright, Nick Darke describe the movement of this species as an action of wind upon each subgroup which can push them in different directions. The interplay of the wind and the animal's physical structure results in a simple behavior that (presumably) is beneficial for the species [5] [6]. However, behavior of the velella is not so important if one refers to the old definition or the new one in the evaluation of velella. The effect of the sail is so that the velella can take the useful advantages of the wind and water force, at any given moment [4].

In this article, we proposed a hypothesis that logically prove velella subgroups are a colony and have interactions together. As we know that velella feed from small prey and fish below the surface of water and also, they use traps with venom to hunt its bait [6]. Thus, according to the evidence of velella's research behavior, when velellas are together on the surface of water, the probability of catching food is increased. In fact, the more rate of venom increases the probability prey in the water surface and for this reason, they prefer to make subgroup formation. Subgroup state and staying together makes much of water area surface covered by venom traps (see Fig. 1). Velellas do not organize subgroups formation without reason although air and water are effective factor on their movements. Subgroup formation on water surface need to be intelligent to make velella resist against the wind force and water flow or to use them efficiently. In subgroups behavior, tentacles of velella sense surroundings to hunt the bait. It's feasible that tentacles of velella sense the nearest neighbors in sense radius and move toward it by useful water flow, wind force on sailing and even with swimming by tentacles. After staying on next neighbors, velella try to keep balance due to changes of subgroups. If weather conditions are being suitable, (without storm) subgroups become in forms of swarms or colonies that try to feed on surface of water [7][8].

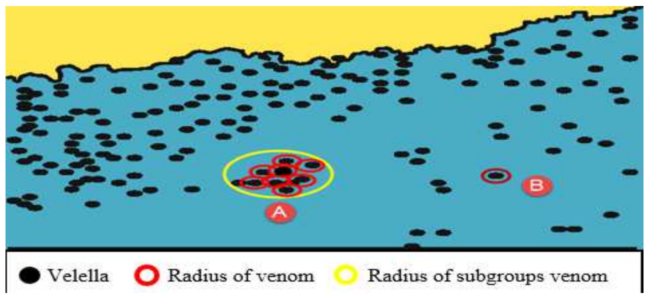


Fig. 1 A. subgroups of velella on surface of water B. alone velella on surface of water. Shown in figure area A, the radius of venom is bigger than area B. it's clear that, in the surface of water getting together of velella increases the cover radius of venom in area and increases covered area of venom that effects catching food.

Research on Swarm intelligence (SI) behavior for solving optimization problem have been widely developed in artificial intelligence sciences. An SI system ability refers to artificial intelligence techniques in swarm system to behave in a complex pattern and self-organized way without any specific individual control. In solving optimization problem by a heuristic algorithm, each single individual of swarms has an intelligence interaction in problem space which represents a potential solution. In problem space, local interaction between simple agent's exhibit swarm intelligence for local search and then exchanged processing information emerge complex patterns as global search by lapse of time. Thus, interaction between individuals is very important to develop a new heuristic optimization algorithm based on a biologically-inspired behaviour. This paper is organized in seven section. In the next section originality of this work is introduced and after that motivation and theoretical approach is discussed. Methodology is in the four section and section five is about comparing methods and experimental results are shown in section six. Finally conclusion section is at the end of this article.

2. Motivation and originality

Providing a simple mathematical model to describe the phenomena of the real world has a significant effect on perception of heuristic algorithms. Hence, in last decades, a wide range of approaches for heuristics algorithms were presented such as Simulated Annealing(SA) [9], Genetic Algorithm(GA) [10], Particle Swarm Optimization (PSO) [11] and Ant Colony Optimization (ACO) [12], which are the most representative samples in heuristic approaches that use different phenomena to solve complex computational and optimization problems [10]. These algorithms well suited to solve different problems such as finding objective functions [6], pattern recognition [13], image processing [4], swarm robotics [14], etc. However, each heuristic algorithm has a various performance in a specific problem to find the best solution [15].

This paper presents a new heuristic optimization algorithm based on collective behavior of velella in the nature. Proposed algorithm simulates the subgroups behavior of velella according to the interaction rules. In this optimization algorithm the objective function in problem space is defined as the amount of toxin (venom) concentration which sense by tentacles of each individual of velella in a radius. Water flow, wind force and swimming by tentacles are three effective moving factors in position of velella. In more, content of this paper is as follows: characteristics of velella swarm optimization (VSO) algorithm and theory of velella movement introduced in third section. In section four, results of another optimization algorithm are compared with our proposed algorithm. Eventually, Conclusion of the contents and suggestion for future research provided as the fifth Section.

3. Sailing physics

Due to details study of velella simulation, movement behavior on the water surface with water flow and wind power is a problematic task. Normally, in the real world, whether conditions and environment vary and they have impact on velella sail. Water and wind force variation with diverse direction are effective samples of velella movement. However, using motion Physics can propose a simple and comprehensible equations for velella movement. A simple sailing theory is practically attractive and can describe performance of those animals because physical structure of velella is similar to sailboats. Hence, in this research, sailboat physics is used to describe performance of the velella sail.

In sailboats physic study, a mathematical equation is presented for movements of sailboats which is relevant and similar to the aerodynamic performance of the velella sail explained in [17].

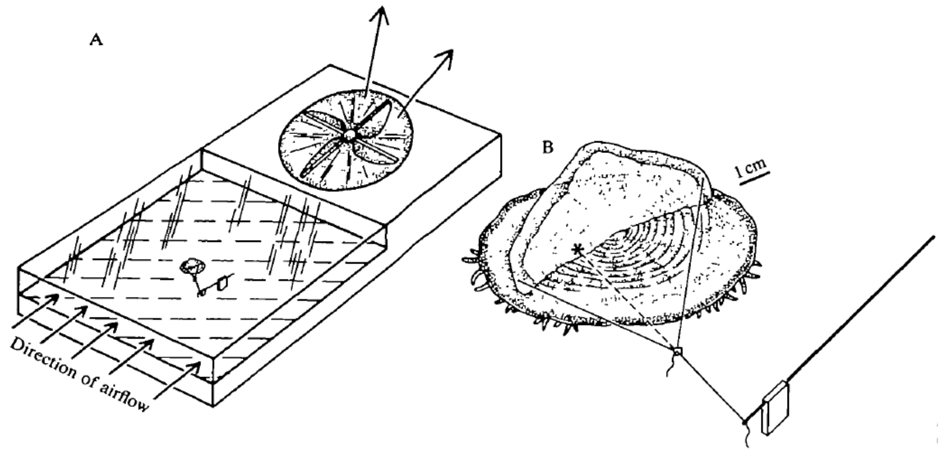


Fig.2 Artificial model of velella.

Therefore, in this research an artificial model is designed according to the physical structure of velella and then, based on experimental results obtained from simulation of movement, a theory is presented for velellas which expressed by an equation as follows:

$$C_f = \frac{2F}{\rho s U} \tag{1}$$

In equation (1), F is wind power which is entered on the sail of velella. Cross section sail of velella is calculated with equation (2):

$$AR = \frac{H^2}{S} \tag{2}$$

In equation above, H is maximum sail height and S determines width of the sail. In fact, AR determines cross-section width of sail in the artificial model.

In equation (1), ρ is coefficient fluid density and s are the coefficient of the water force push to the body of the velella and U is the velocity of water flow.

Thus, in equation (1); C_f; the resultant vectors of wind and water force pushing to the sail of velella, can be achieved which makes their movement. Regarding to this study, physical structure of the velellas is similar to the physical structure of the sailboats. As sailboat physics is similar to velella's movement so water and wind force are effective on sailboat's movement physics. Thus, in this article sailboat movement physics on water surface is proposed based on theory of equation (1). Generally, research on sailboat movement physics expressed in the form of equation (4) [18].

$$m^{*}\frac{d}{dt}u(t)=\overrightarrow{F}_{d}(wind)+\overrightarrow{F}_{d}(water) \tag{4}$$

Equation (4) introduce a simple expression of wind forces $\vec{F}_{d}(wind)$ and simple expression of water force $\vec{F}_{d}(water)$ on sailboat which are effected simultaneously. These equations are defined in the following respectively:

$$\vec{F}_{d}(wind)\equiv\frac{1}{2}C_{D}(sail)\times A(sail)\times\rho(air)\times V^{2} \tag{5}$$

In equation (5), $\overrightarrow{F}(wind)$ is the wind force with a direction which enters to the sail of boat, C_D is a constant (usually equals to 4/3), A is coefficient of sail cross-section, ρ is density of the air and V is the velocity of wind which hits the sail.

$$\vec{F}_{d}(water)\equiv\frac{1}{2}C_{D}(bull)\times A(bull)\times\rho(water)\times U^{2} \tag{6}$$

Where $\overrightarrow{F}(water)$ is the water force with a direction is entered to airfoil of sailboat, C_D is constant coefficient of resistance (usually equals to 4/3), A is a part of sailboat airfoil touch within water, ρ is fluid density and U is mainstream fluid velocity.

Due to the physics movement of velella and sailboat, in equation (1) and equation (4) it can be concluded that theory of velella movement fits with theory of sailboat movement on the water. Proposed equation can be expressed in equation (7) which is derived from the above equations:

$$m^{*}\frac{d}{dt}u(t)=\frac{\overrightarrow{F}_{d}(wind)}{\overrightarrow{F}_{d}(water)}\Rightarrow C_{f}=\frac{2F}{psU} \tag{7}$$

In artificial system of velella that is proposed here, a movement is introduced based on sailing of sailboat physics. Therefore, according to the physics movements of sailboat we are describe a new simple equation for artificial system model which's expressed in equation (8) as follows:

$$\begin{aligned}\vec{F}_{d}(wind)&\equiv\ \tfrac{1}{2}C_{D}(sail)\times A(sail)\times\rho(air)\times V^{2}\ \rightarrow\ \vec{F}_{d}(wind)=\alpha\times\vec{V}\\ \vec{F}_{d}(water)&\equiv\ \tfrac{1}{2}C_{D}(bull)\times A(bull)\times\rho(water)\times U^{2}\ \rightarrow\ \vec{F}_{d}(water)=\ \beta\times\vec{U}\end{aligned} \tag{8}$$

In the equation above, the number of coefficients for the artificial model is reduced. In equation (8) α, is parameter that controls the coefficient of wind power (coefficient cross-section of sail) and β, is a parameter that controls the coefficient of water force (coefficient of influence of water force to body or tentacle of velella). In next section, our proposed method is expressed based on the above methodology.

4. Velella algorithm

Now, according to the physics of boat sailing movement and similarity of velella physics sailing, it is possible to design and implement a new optimization algorithm inspired by the behavior of this animal in nature. In this section, we proposed an artificial model based on biological behavior and theory performance of the velella sail. According to subgroups behavior, each agent on the water surface has interaction with the other agents by tentacles. In the proposed method in accordance with a case study, competency is required to create subgroups formation behavior. For this propose, the venom amount of agent is set as objective function. In other words, the position of each agent has a rate of venom and radius neighborhoods which exchange processing information to each other as local search. Deaths is considered as global search due to storms and directing to the shore by the water of flow. In fact, the power of wind always guides agents toward the ocean shore which is a common property of agents.

In the proposed algorithm, water surface is set as problem space in which velellas are agents and their competency are measured by rate of their venom. We have assumed as a strategy to determine the global search which an agent to have more competency comparing with other agents is nearest agent to the shore. In our proposed method navigate agents toward the shore considered as the strategy of global search. In fact, an agent with the most rate of venom is defined as ocean shore. The direct communication of agents for subgroups behavior formation guarantees the exploitation property step in search space and convergence of all agents in any movement to the shore guarantees the exploration in the algorithm exclusivity. In velella swarm optimization, each agent has four characteristics: position, initial value of venom, wind and water force. Each position of the velellas in search space guarantees a solution of problem, and its venom rate is determined as an objective function.

Now, with the methodology of optimization algorithms, it is expected that velella act as an artificial world and surface of water considered as the search space of the problem. The position of agent (velella) on search space is defined in equation (9):

$$\overrightarrow{X_i} = (x_i^1 \ldots . x_i^d \ldots . x_i^n) \quad for\ each\ i\ from\ 1\ to\ N \tag{9}$$

Where, N presents the number of agents and vector x describes the position of agents for each i in the dth dimension space. In addition, each agent with vector x represents the quality of fitness (rate of venom). The value of venom is determined by given agent's position on the objective function. In other words, the position of each agent represents quality of venom or concentration rate of venom which deposited by the velella and determined as $fit\ (x(t))_i^d$.

In maximization and minimization problems the fitness value is calculated by the following equations respectively:

$$best(t) = \max_{i \in \{1...m\}} fit_i(t)$$

$$best(t) = \min_{i \in \{1...m\}} fit_i(t) \tag{10}$$

An agent requires a vector of velocity and radius neighbourhood sense to update current position ($x(t)$) to a new position ($x(t+1)$) so the size of radius neighborhood is configurable. The speed vector of each agent affected by the water and wind force impact on current state ($F_i^d(t)$) going to the next state ($F_i^d(t+1)$).

The new position of agent i; ($X_i^d(t+1)$) is calculated with the total current position ($X_i^d(t)$) and updated by the new velocity in equation (11) to $F_i^d(t+1)$; velocity dimensional of agent i in time t.

$$F_i^d(t+1) = F_i^d(t)$$

$$X_i^d(t+1) = X_i^d(t) + F_i^d(t+1) \tag{11}$$

It should be noted that in our proposed method, the radius of the neighborhood does not mean the step movement length of the agent and only determines the range of vision for that agent. Therefore, in the problem space wind and water force determine movement step length of the agent. In this algorithm, the status of each agent is updated by velocity according to the global and local optimization strategy. Therefore, for global search strategy, agent with best fitness is defined as the edge of shore and global optimization in problem space. In the other word global optimization strategy is a common feature between agents which wind force navigated them toward the edge of shore. Thus, the way of velella's death in nature is considered to be as global optimization strategies as illustrated in Fig 3.

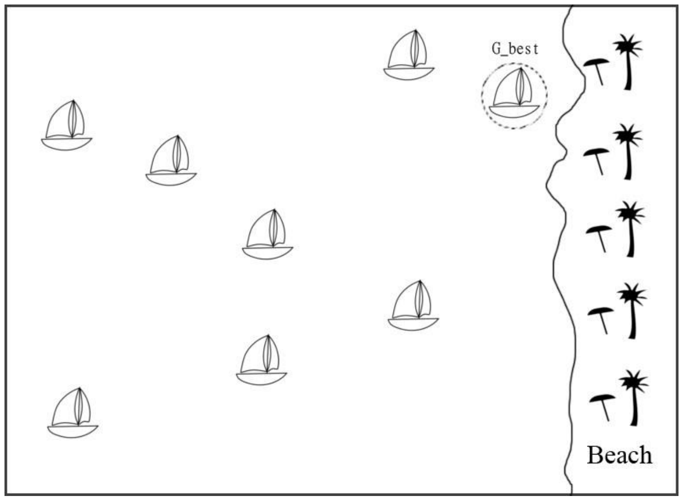


Fig.3. G_best agent has the best amount of fitness among search agents. According to the contract the G_best is closest agent to the edge of shore and considered as global search strategy.

Thus, in the artificial models, a simple equation is defined to update global search strategy of agent movement from x(t) state to x(t+1)as follows:

$$\mathrm{F}(t+1)_i^d = \alpha_i(t) \times V(t) \times rand(G_{best} - X_i(t)) \tag{12}$$

Above equation determine wind impact on agent movement on the water surface toward the global optimization. α is sail cross-section coefficient, V is velocity of wind which push sail of artificial agent and it is considered to be 2. rand is a random number in range of [0,1] with a uniform distribution. The value of Wind force impact coefficient (α) considered to be 0.9. G_best is the best position of an agent with high competency which is considered as the global search (ocean shore).

Local search strategies are also defined based on subgroup behavior formation of velella. Subgroups behavior movement of agents to target is in Sin form. In local search strategies, there are three possible occurrences that describes in the following respectively.

First state: If there is a best neighborhood ($L_{nigberhood\_best}$) for the sense radius of neighborhood agent x(t) in search space, agent x(t) moves one step towards its neighbor which its force is defined in equation 13:

$$\vec{F}(water)\,_i^d(t) = \beta_i(t) \times U(t) \times \text{ Sin}\,(rand)\,(L_{nigberhood_{best_i}} - X_i(t)) \tag{13}$$

In equation (13) β is impact coefficient of water force on agent movement and U is velocity of water pushed to the agent body (water force) and consider constant (U=2). Sin, is sinus shape of water, and rand is a random number in the range of [0,1] with a uniform distribution. Impact coefficient of water force (β) considered to be 0.2 for all three states as illustrated in fig 4.

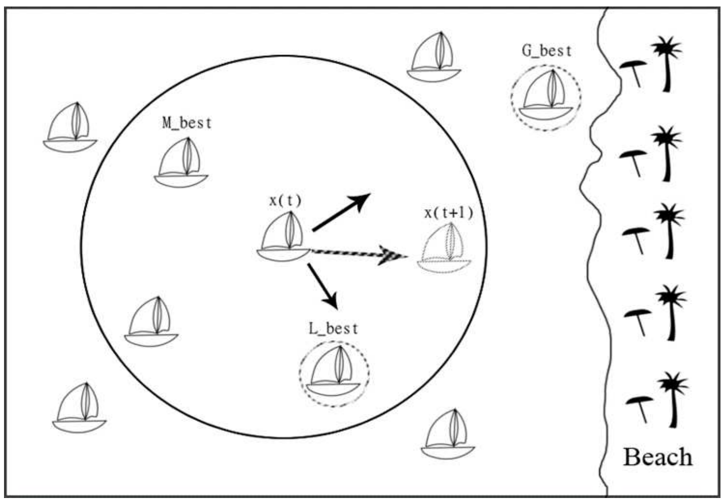


Fig.4. an agent x (t) with the sense radius evaluates the fitness of its surrounding agents and determines the best value of neighboring target's for moving toward it with the water force step. The next step for agent x (t) is moving toward to G_best.

Second state: If there was no agent for the best neighbor in the sense radius, then it would act according to the best memory of its own as a step, which is calculated as follows:

$$\vec{F}(water)\,_i^d(t) = \beta_i(t) \times U(t) \times \text{ Sin}\,(rand)\,(L_{nigberhood_{best_{mem_i}}} - X_i(t)) \tag{14}$$

$L_{nigberhood_{\_best\_}mem}$, is the best personal memories of agent x(t) as illustrated in Fig 5.

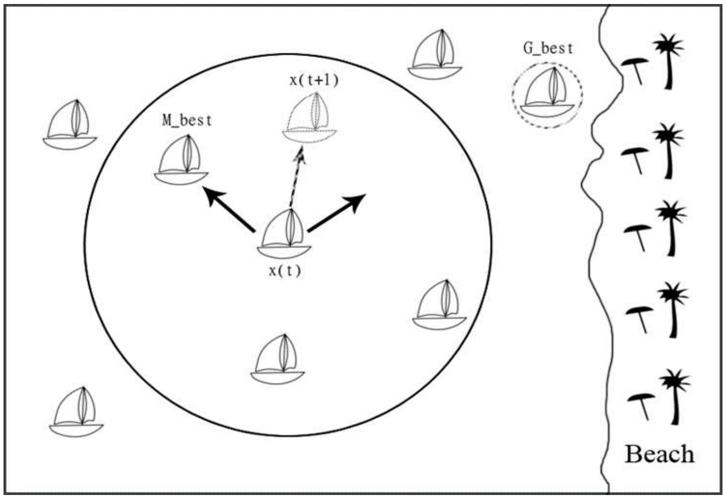


Fig.5. If there isn't better neighboring around agent x(t) then moves one step towards the best of its own personal memory.

Third state: If the agent x(t) doesn't have a personal memory for the second state, then it picks a random step which is shown in fig 6 and calculates as follows:

$$\vec{F}(water)\,_i^d(t) = \beta_i(t) \times U(t) \times \text{ Sin }(rand) \quad (15)$$

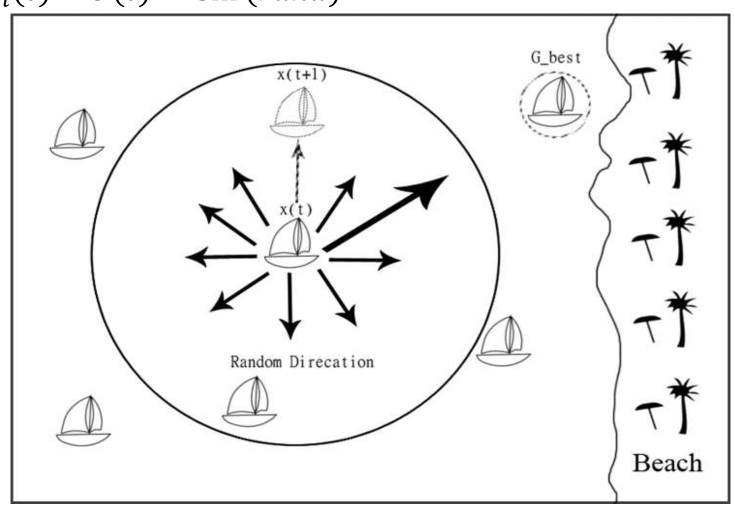


Fig.6. If there is no agent with better and personal memory around agent x (t) then moves one random step. The next step of x (t) is in the direction of G_best.

In general, total forces entered into agents is calculated as follows:

$$F_i^d(t) = \alpha_i \times V(t) \times rand(\ G\_best) + \beta_j(t) \times U \times Sin(rand) \times (L_{nigh}, p_{best}, rand)$$

(16)

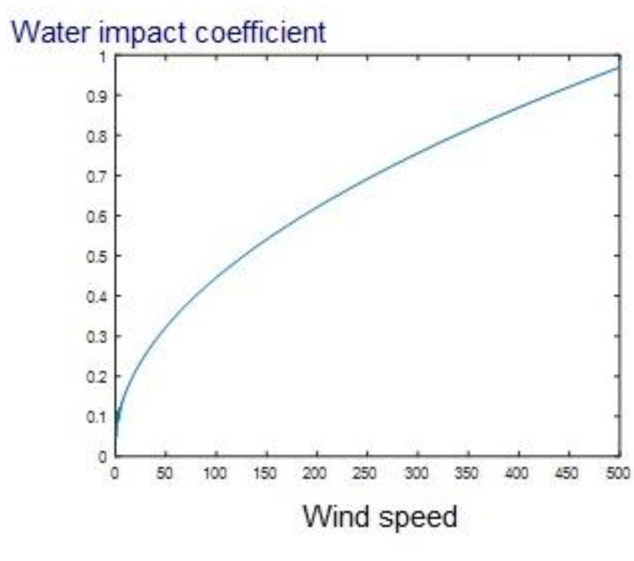
With regard to the equation (16), the wind force is common feature of agent's move. In the other side, each of the three state provides resistance to create subgroups behavior preventing them from navigation to the shore. Logically, coefficients of impact forces determine the value of forces push to the agents. Because acceleration is a proportional to speed, acceleration is replaced with velocity movement in the proposed method.

Regard to methodology of this optimization algorithm, the wind force coefficient considered as exploration approach and water force coefficient describes exploitation capability of the algorithm. According to the coefficient of wind and water force they can be increases and decreases in exploration and exploitation approaches at the end of each step of the algorithm.

These coefficients can tune the exploration and exploitation performance of algorithm at the end of stages depending on the type of optimization problem. For example, coefficient of the global search optimization algorithm (wind force) starts from an initial value and decreases over time in the process of operation. In other words, the wind velocity is selected higher at the initial stages to search more freely in problem space. But at the end stages of the algorithm, agents resist to search so freely so water and wind velocity coefficient should be decreased which define local search strategies. The coefficients of water and wind velocity can be increased and decreased at end stage of the algorithm i.e. relation of those velocities make an increase in water velocity. For example, in equation (17) the factor of water velocity (β) is increased which is shown in fig 7:

$$\beta = \frac{\alpha}{\beta} + swim \tag{17}$$

In the above equation, swimming (swim) parameter considered to be 0.04 and relation of impact coefficient of water and wind leads to increase rate of water velocity.



wind

Fig.7. Water impact coefficient resistance to wind velocity diagram.

Thus, the best neighbour in each of those three states (local optimal) is selected with respect to the fitness of neighbourhoods. The fitness level is adjustable for the best neighbourhoods regarding the type of optimization problem (maximization and minimization).

Pseudo code of the proposed algorithm is described below:

Pseudo code of the VSO algorithm ---

a) Start.
b) Define problem search space.
c) Initialize each agent randomly.
d) Evaluate fitness of agent based on venom rate.
e) Calculate wind and water force impact on agents.
f) Update α and β parameters.
g) Update the position of agents.
h) Repeat levels d to g until the stop criteria is reached.
i) End.

Fig.8 Pseudo code of VSO

Principles of VSO algorithm is shown in Fig.8 and its flowchart is illustrated in Fig.9.

To see how the proposed algorithm is efficient some remarks are noted:

- In VSO algorithm an agent with more venom rate considered as global search strategy.
- Global search strategy is equivalent to the oceans which is determined as a common feature between searcher agents.
- Agents with wind force moves toward the global search in the search space. The value of α coefficient determines impact of wind force to all agents.
- Exploration capability is strengthened with wind force in VSO. If the value of α is high then wind force increases effect of agents.
- Subgroups behavior formation on search space is described with three state as local search strategies.

- The present work provides communication and coordination of agents in local search strategies in the radius sense. Size of radius sense determine the range of vision agents.
- Size of radius sense neighbourhood is adjustable in a large area due to the steps of the agents move defined with water and wind forces. But limited radius sense reduces the performance of search algorithm.
- In local search strategies agents move by water force which is similar to water flow described as Sin form. The value of β coefficient determines impact of water force to all the agents.
- Best neighbourhood evaluation, personal memory and non- neighbourhood agents in radius sense is the component of local search strategies.
- Local search strategies guarantee the exploitation capability with water force. If the value of β is high enough then water force increases effectiveness of the agents.
- It is possible to increase or decrease coefficients of wind and water force in final stages according to the type of optimization problem.

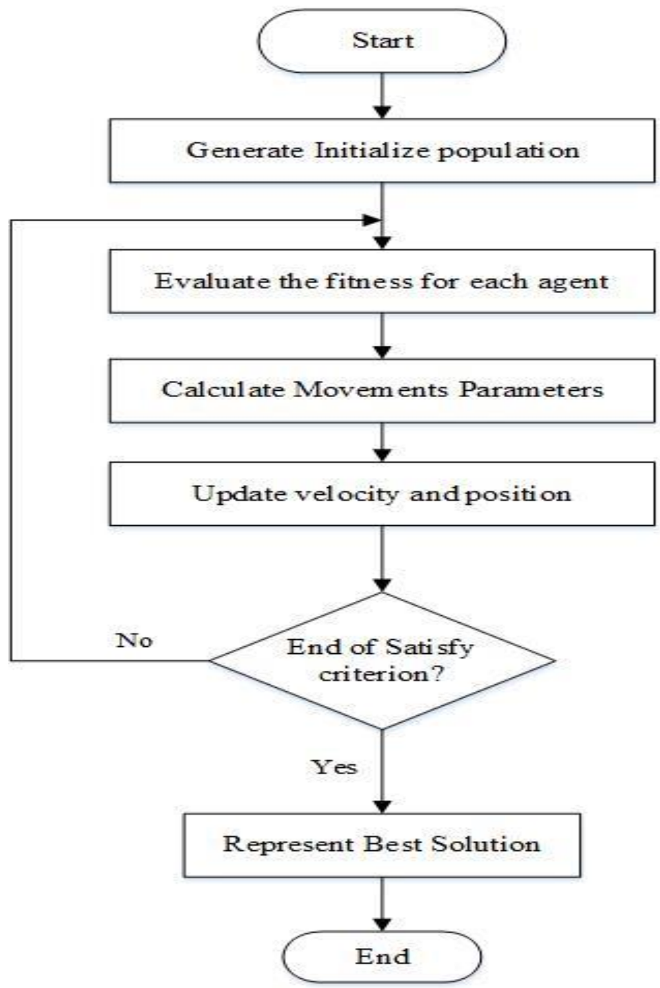


Fig.9 flowchart of VSO

5. Comparative study

In this section, our method is compared with other heuristic search algorithms. For this purpose, stochastic heuristic search, PSO (particle swarm optimization), and GA (Genetic algorithm) approaches compared with VSO theoretically. At the end of this section, an experimental result of those algorithms is reported in optimization problems.

- **Particle Swarm Optimization**

The PSO algorithm inspired from social behavior of birds and flocks. In simulation of this approach the particles at problem space update situations according to the fitness information. Best fitness in search space is considered to be objective function which population members move towards them. Position and velocity of particles in PSO are updated in below equations:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (18)$$

$$V_i(t+1) = wv_i(t) + C_1 r_{i1}\big(pbest_i - x_i(t)\big) + C_{2i2}(Gbest - x_i(t)) \quad (19)$$

By comparing theoretical differences, it can be concluded that, both PSO and VSO inspired from biological behavior in nature but updating equation movement of agent in VSO is different with PSO. Although VSO determines global best similar to PSO strategies, PSO uses a pbest and gbest to update the velocity while, VSO uses best local neighborhood and personal memories to obtain global best. VSO agents limited to radius neighborhood sense in search space, but, PSO uses direct interaction between agents to exchange information in problem space.

- **Genetic Algorithms**

GAs inspired from biological organisms which is used to solve optimization problems. In GA each individual represents a possible solution which is assigned by a fitness according to problem types. High fitness of chromosomes produces new generation with mutation and crossover operator. Mating of the best fitness individuals in the population leads to converge to an optimal solution. In the next section experimental results of those algorithms are compared with proposed method.

- **Gravity Search Algorithms**

Gravitational search algorithm is inspired by the law of gravity and the mass concept. Searching agents in this algorithm are a set of objects. According to the laws of physics all materials attract each other by gravity and heavier objects have a greater impact than light objects. In this algorithm, the gravitational force is considered as an interface for data transfer and communicate between objects. Therefore heaviest object is the optimal point of interest in ideal conditions. Objects follow by gravity and moving law in GSA. In the gravity law, any object, absorb other objects by a force directly proportional to their masses and inversely proportional to the square of the distance between them [16].

6. Experimental results

In this section, proposed method is applied to 26 minimization CEC 2014 standard benchmark functions and its results is compared with the results of GAs and PSO optimization algorithms.

The characteristics of GA, PSO algorithms are described in table 3.

Table 3. Parameter settings of evaluation algorithms

GA parameters		PSO parameters	
Crossover fraction	**0.8**	**C1 & C2**	**2**
Selection	**Tournament**	**W**	**0.9**
Crossover	**Arithmetic**	**damping**	**0.04**
Mutation	**Adaptive**	**-**	**-**

Also, evaluation characteristics of VSO algorithm is stated in table 4

Table 4. Parameter settings of VSO algorithm

Velella swarm Optimization algorithm	
α	**0.9**
β	**0.2**
Wind velocity	**2**
Water velocity	**2**
Swimming	**0.04**
Radius (Default)	**2.5**

In all cases of testing phases, population size on search space is 50 agents (N=50) and maximum number of iterations for each run is set to 500. Obtained results is averaged over 30 independent runs. Benchmark functions conducted on 26 well-known classic functions which their related images are illustrated in the following forms in Fig 10.a and Fig 10.b.

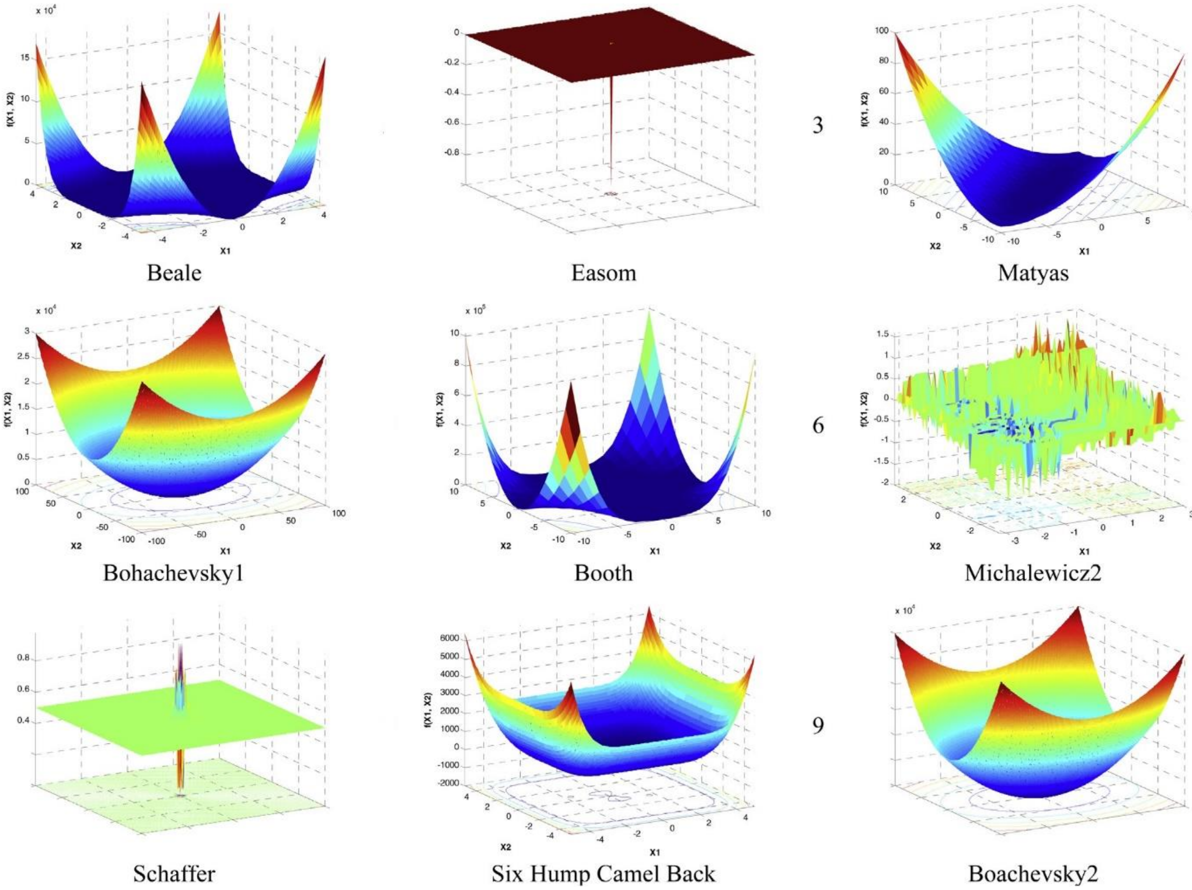


Fig. 10a. A perspective view of CEC2014 benchmark functions.

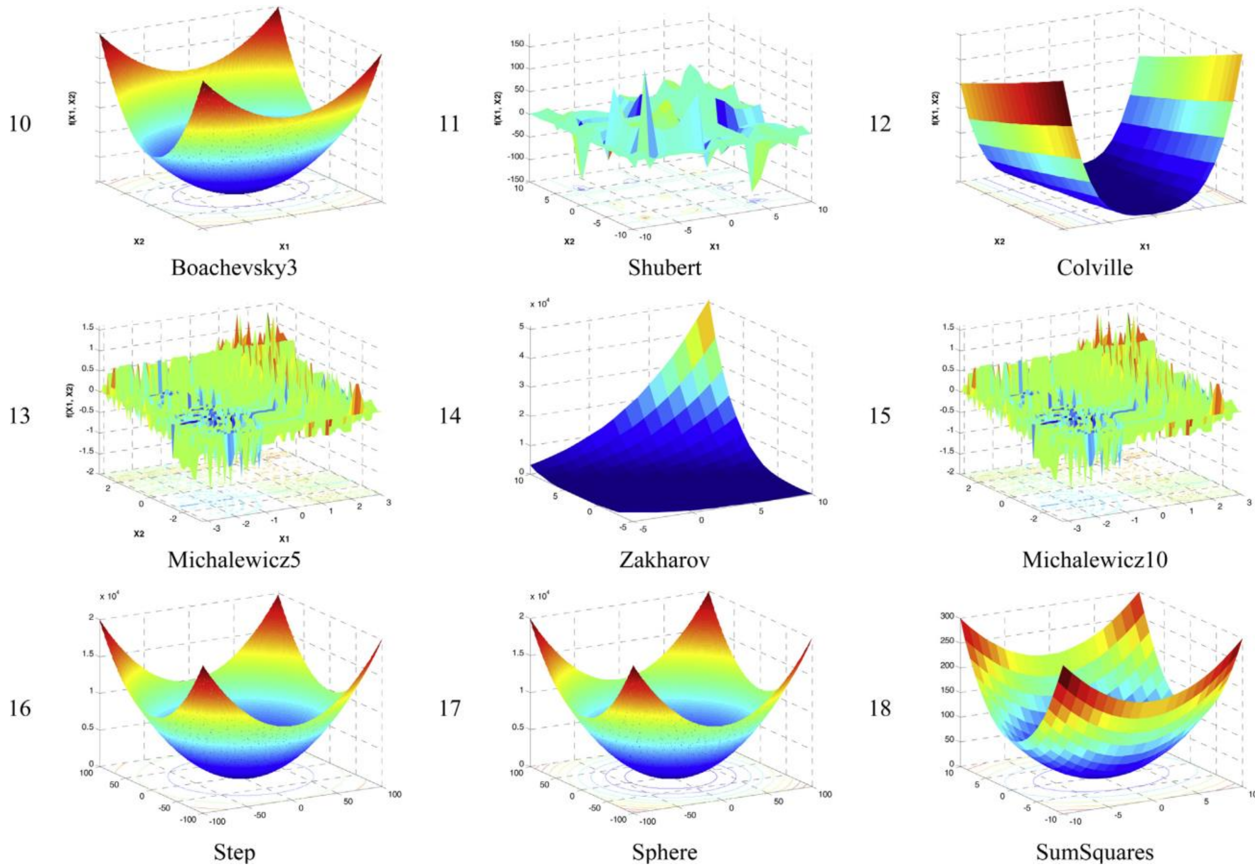


Fig. 10 b. A perspective view of CEC2014 benchmark functions.

Test 1. Unimodal and Separable benchmarks Functions

Converging to optimal solution is very important in unimodal and separable benchmark functions with high-dimension that can be seen in table 1.

Table 1. Features of classical unimodal and separable benchmark functions.

Function	Name	Expression	Range	d	F_{min}
F1	**Step**	$x=\sum_{j=1}^{d}(x_j+0.5)^2$	**[-5.12,5.12]**	**30**	**0**
F2	**Sphere**	$x=\sum_{j=1}^{d}x_j^2$	**[-100,100]**	**30**	**0**
F3	**Sum Squares**	$x=\sum_{j=1}^{d}jx_j^2$	**[-10,10]**	**30**	**0**
F4	**Quartic**	$x=\sum_{j=1}^{d}jx_j^4+Rand$	**[-1.28,1.28]**	**30**	**0**

Obtained result of evaluating those three algorithms are shown in table 2.

Table 2 Obtained results by GA, PSO and VSO through 30 independent runs on benchmark functions in table 1

Function		GA	PSO	VSO
F1	Best	2.6337e-03	6.0105e-13	**1.3682e-30**
	Worst	1.3059e-02	2.13444e+01	**1.2849e-27**
	Mean	6.4535e-03	4.2689e+00	**1.3586e-28**
	SD	2.3013e-03	8.6837e+00	**3.0575e-28**
F2	Best	1.0075e+00	2.5973e-07	**5.2452e-68**
	Worst	4.4123e+00	1e+04	**2.5742e-61**
	Mean	2.5543e+00	6.666667e+02	**8.9897e-63**
	SD	9.1794e-01	2.5370813e+03	**4.6941e-62**
F3	Best	1.116e-01	5.7613e-09	**7.9867e-71**
	Worst	6.4861e-01	1.4e+03	**4.7579e-66**
	Mean	2.9535e-01	4.133334e+02	**5.5316e-67**
	SD	1.183e-01	3.692801e+02	**1.3508e-66**
F4	Best	3.4413e-02	1.0301e-01	**8.0178e-04**
	Worst	1.8393e-01	8.2266e+00	**1.0902e-02**
	Mean	7.7226e-02	1.1005e+00	**3.5032e-03**
	SD	3.0953e-02	2.2111e+00	**2.5504e-03**

As seen in table 2, VSO provides better results than GA and PSO algorithm for unimodal and separable functions in table 1. The performance of VSO, PSO and GA algorithm is shown in the following graphs in Fig 11-14 for function F1 to F4 respectively. Radius sense of neighbourhood for VSO algorithm is demonstrated in table 4.

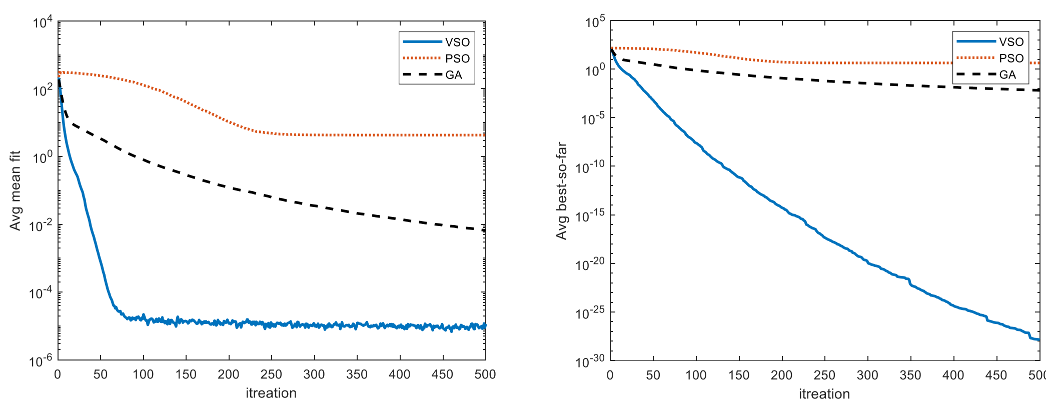


Fig. 11. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO, and GA performance for minimization benchmark functions F1.

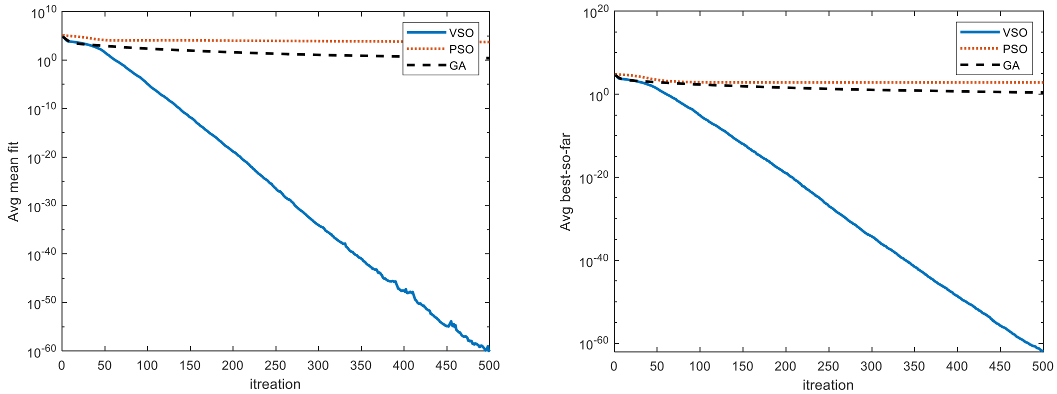


Fig. 12. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F2.

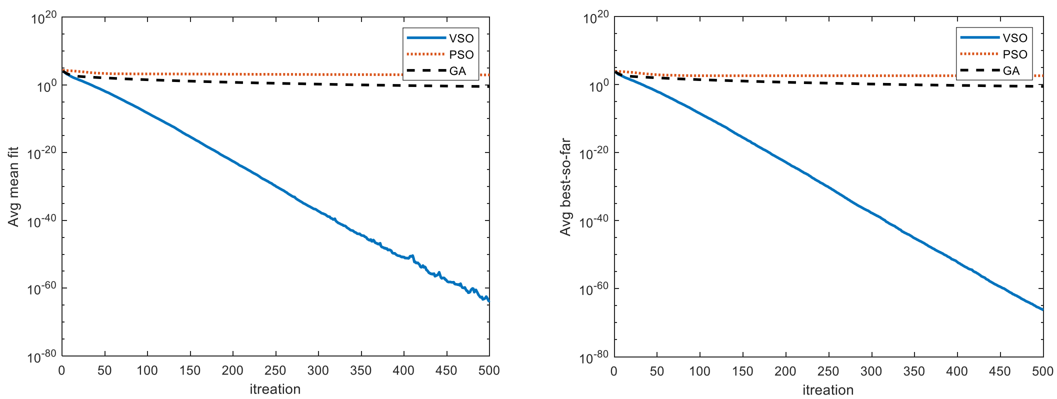


Fig. 13. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F3.

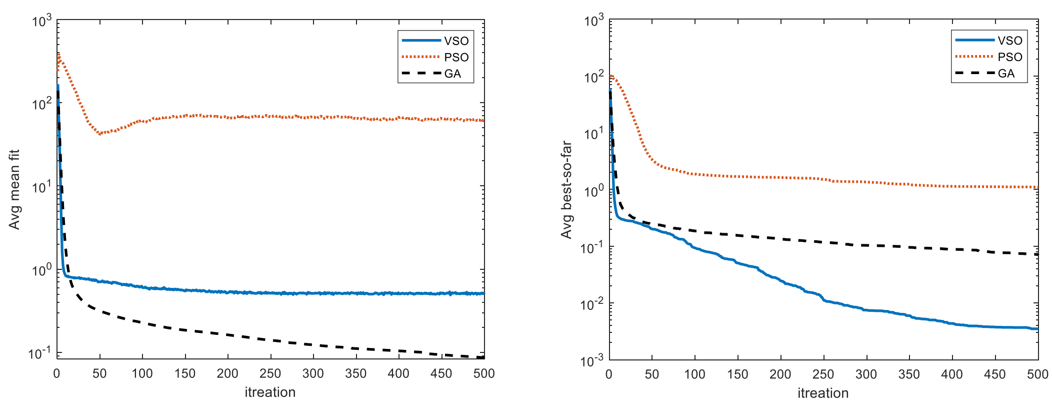


Fig. 14. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F4.

Table 3. Neighborhood radius tuning for classical unimodal and separable benchmark functions used in experimental test 1.

Function	VSO neighborhood radius for CEC 2014 benchmark functions
F1	**Default (2.5)**
F2	**6.5**
F3	**4**
F4	**Default**

Test 2. Unimodal and non-separable benchmark functions

Probability of being caught in local optimums is high in optimization algorithms limited to radius neighbourhood using unimodal and non-separable with convex function.

Table 4. Features of classical unimodal and non-separable benchmark functions.

Function	Name	Expression	Range	d	F_{min}
F5	**Beale**	$x=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1x_2)^2+(2.625-x_1+x_1x_2^3)^2$	**[-4.5,4.5]**	**2**	**0**
F6	**Easom**	$x=-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	**[-100,100]**	**2**	**-1**
F7	**Matyas**	$x=0.26(x_1^2-x_2^2)^2-0.48_{x_1 x_2}$	**[-10,10]**	**2**	**0**
F8	**Zakhrov**	$x=\sum_{j=1}^{d}x_j^2+(\sum_{j=1}^{d}0.5jx_j)^2+(\sum_{j=1}^{d}0.5jx_j)^4$	**[-5,10]**	**10**	**0**
F9	**Schwefel2.22**	$x=\sum_{j=1}^{d}\lvert x\rvert+\Pi_{j=1}^{d}\lvert x_j\rvert$	**[-10,10]**	**30**	**0**
F10	**Schwefel 1.2**	$x=\sum_{j=1}^{d}(\sum_{k=1}^{j}x_k)^2$	**[-100,100]**	**30**	**0**
F11	**Dixon-price**	$x=(x_1-1)^2+\sum_{j=2}^{d}j(2x_j^2-x_j-1)^2$	**[-10,10]**	**30**	**0**
F12	**Colville**	$x=100(x_1^2-x_2)^2+(x_1-1)^2+(x_3-1)^2+90(x_3^2-x_4)^2+10.1(x_2-1)^2+(x_4-1)^2+19.8(x_2-1)(x_4-1)$	**[-10,10]**	**4**	**0**

Capability of algorithms exploration must be high facing with such problems with Beale convex function according to table 4. In spite of proposed algorithm is limited to radius neighbourhood and sometimes fall in local optimum, it finds minimum solution in almost of independent executions as shown in table 5.Obtained result of evaluating those three algorithms are shown in table 5.

Table 5 Statistical results obtained by GA, PSO and VSO, through 30 independent runs on classical unimodal and non-separable benchmark functions.

Function		GA	PSO	VSO
F5	**Best**	0	6.4216e-08	**0**
	Worst	3.5751e-03	3.6006e-04	**1.3641e-25**
	Mean	5.864e-04	7.3731e-05	**5.1009e-27**
	SD	1.0265e-03	8.647e-05	**2.4981e-26**
F6	**Best**	-1	-1	**-1**
	Worst	-1	-1	**-1**
	Mean	-1	-1	**-1**
	SD	2.6281e-06	0	**0**

F7	**Best**	9.3119e-48	1.4704e-130	**0**
	Worst	1.0293e-04	3.5137e-107	**4.9525e-317**
	Mean	7.4122e-06	1.1759e-108	**1.6509e-318**
	SD	2.1195e-05	6.4142e-108	**0**
F8	**Best**	2.0022e-+01	1.4952e-08	**0**
	Worst	2.6627e+00	3.36607e+01	**8.2762e-318**
	Mean	1.0348e+00	4.1257e+00	**2.7588e-319**
	SD	6.9027e-01	1.07214e+01	**0**
F9	**Best**	2.318e-01	2.8147e-03	**0**
	Worst	6.1069e-01	6e+01	**1.2994e-157**
	Mean	3.9079e-01	3.05246e+01	**6.2785e-159**
	SD	9.2909e-02	1.38239e+01	**2.559e-158**
F10	**Best**	1.527e+01	1.3973e-06	**0**
	Worst	5.73605e+01	1.3e+05	**1.168e-313**
	Mean	3.18008e+01	2.46666907e+04	**3.8932e-315**
	SD	1.07887e+01	3.5982135e+04	**0**
F11	**Best**	2.3295e+00	1.248e+00	**6.6667e-01**
	Worst	1.2794e+01	1.166768822e+05	**6.6667e-01**
	Mean	6.0073e+00	1.94912836e+04	**6.6667e-01**
	SD	1.9815e+00	4.05025179e+04	**1.5641e-07**
F12	**Best**	5.2938e-04	3.204e-04	**1.1816e-16**
	Worst	6.4695e+00	1.8167e-01	**5.8951e-07**
	Mean	1.7274e+00	5.5179e-02	**1.9995e-08**
	SD	2.28e+00	5.4955e-02	**1.0758e-07**

As seen in table 5, VSO provides better results than GA and PSO algorithm for unimodal and separable functions in table 4. The performance of VSO, PSO and GA algorithm is shown in the following graphs in Fig 15-22 for function F5 to F12 respectively. Radius sense of neighbourhood for VSO algorithm is demonstrated in table 6.

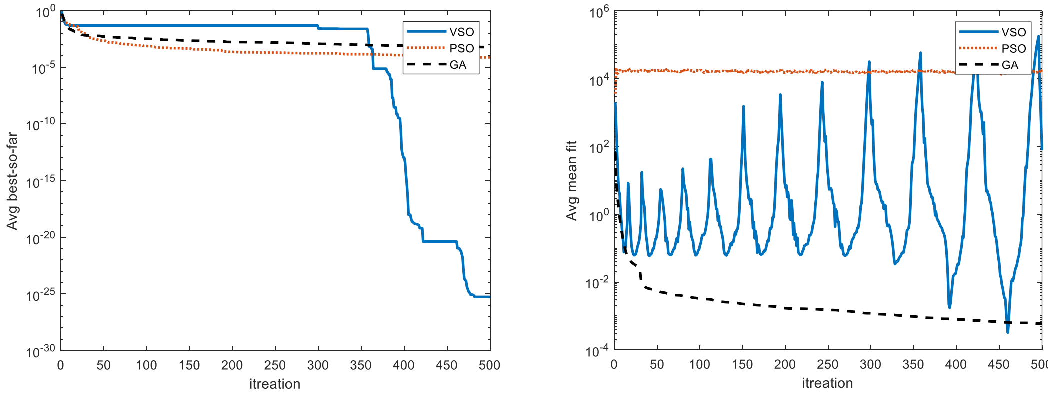


Fig. 15. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F5.

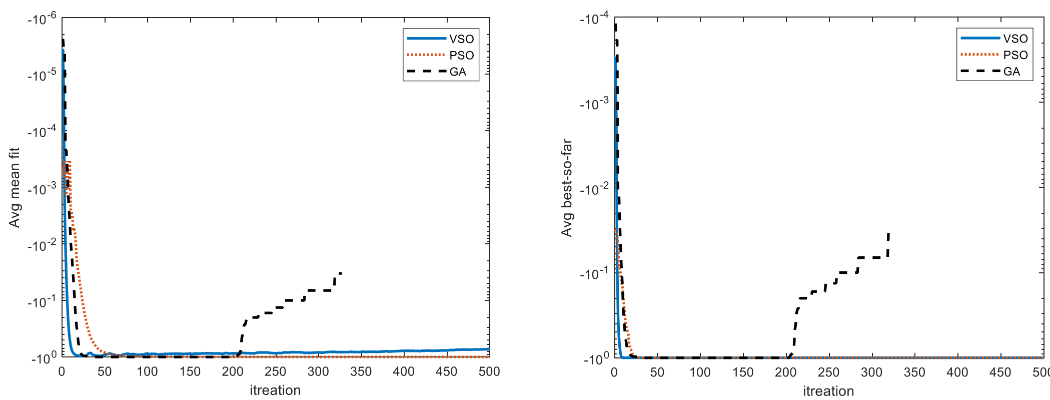


Fig. 16. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F6.

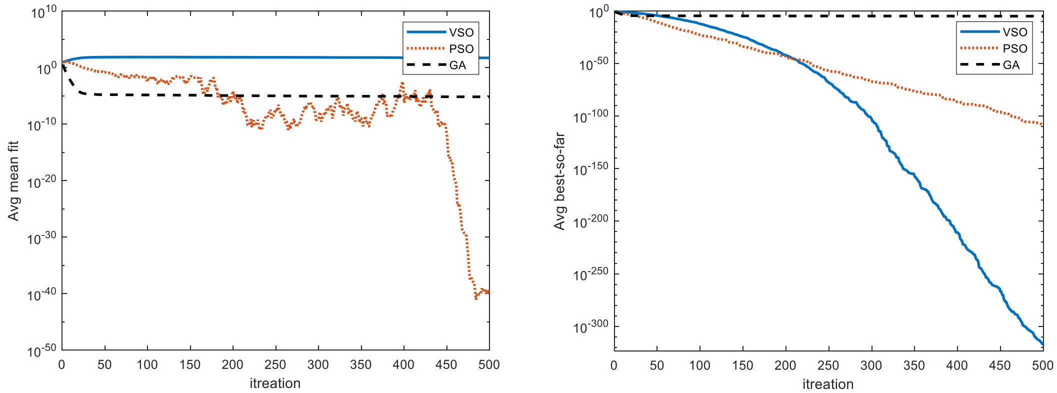


Fig. 17. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F7.

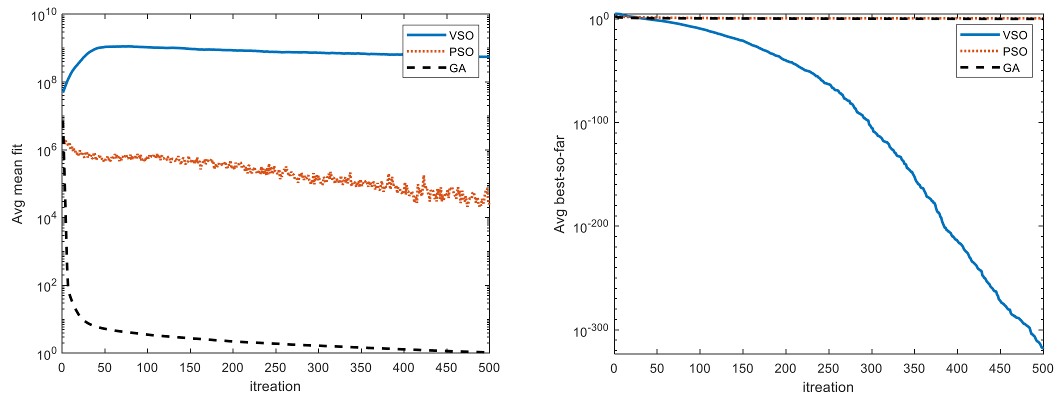


Fig. 18. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F8.

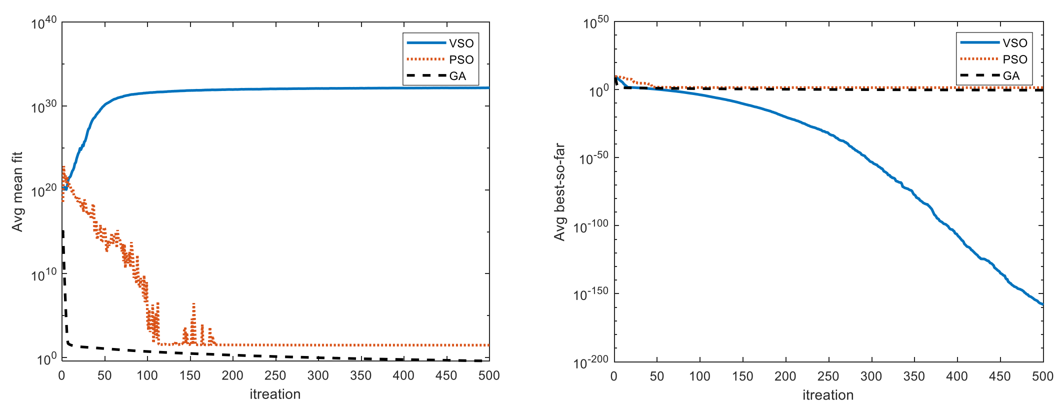


Fig. 19. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F9.

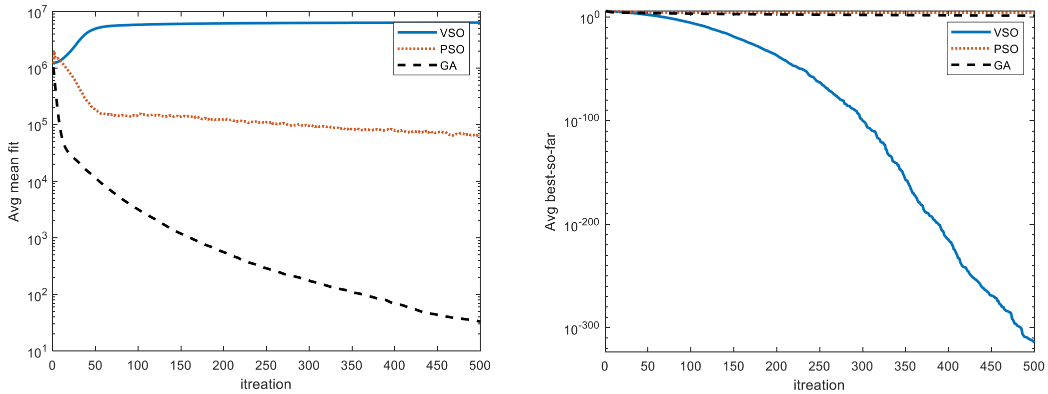


Fig. 20. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F10.

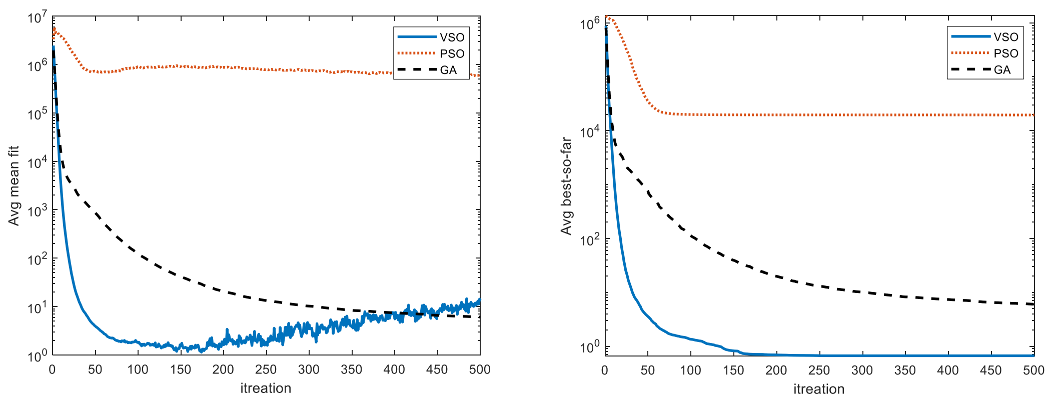


Fig. 21. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F11.

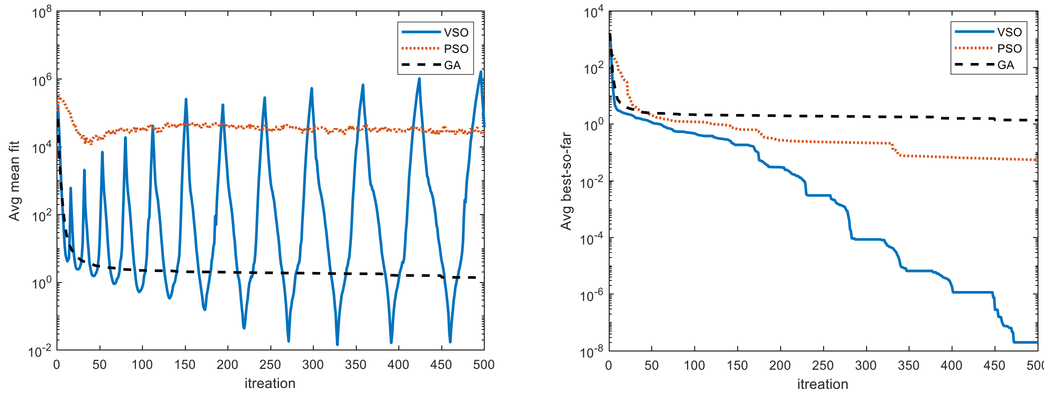


Fig. 22. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F12.

Table 6. Neighborhood radius tuning for classical unimodal and non- separable benchmark functions used in experimental test 2.

Functions	VSO neighborhood radius for the benchmark functions CEC 2014
F5	**default (2.5)**
F6	**default**
F7	**default**
F8	**6.5**
F9	**6.5**
F10	**default**
F11	**12.5**
F12	**6.5**

Test 3. Multimodal and separable benchmark functions

Multimodal and separable benchmark functions are listed in table 7 and obtained result of evaluating those three algorithms are shown in table 8. As seen in table 8, VSO provides better results than GA and PSO algorithm for multimodal and separable functions in table 7. The performance of VSO, PSO and GA algorithm is shown in following graphs in Fig. 23-28 for function F13 to F18 respectively. Radius sense of neighbourhood for VSO algorithm is demonstrated in table 9.

Table 7. Description of classical multimodal and separable benchmark functions.

Function	Name	Expression	Range	d	F_{min}
F13	**Bohachevsky1**	$x = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	**[-100,100]**	**2**	**0**
F14	**Booth**	$x = (x_1 - 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	**[-10,10]**	**2**	**0**
F15	**Michalewicz2**	$x = -\sum_{j=1}^{d} \sin(x_j)(\sin(jx_j^2/\pi))^{20}$	**[0,π]**	**2**	**-1.8013**
F16	**Michalewicz5**	$x = -\sum_{j=1}^{d} \sin(x_j)(\sin(jx_j^2/\pi))^{20}$	**[0,π]**	**5**	**-4.6877**
F17	**Michalewicz10**	$x = -\sum_{j=1}^{d} \sin(x_j)(\sin(jx_j^2/\pi))^{20}$	**[0,π]**	**10**	**-9.6602**
F18	**Rastrigin**	$x = \sum_{j=1}^{d} (x_j^2 - 10\cos(2\pi x_j) + 10)$	**[-5.12,5.12]**	**30**	**0**

Table 8. Statistical results obtained by GA, PSO and VSO, through 30 independent runs on classical unimodal and non-separable benchmark functions.

Function		GA	PSO	VSO
F13	**Best**	0	**0**	**0**
	Worst	5.7482e-05	**0**	**0**
	Mean	1.9861e-06	**0**	**0**
	SD	1.0488e-05	**0**	**0**
F14	**Best**	0	0	**4.3934e-09**
	Worst	5.2997e-01	5.0724e-01	**7.2617e-05**
	Mean	3.7398e-01	2.3678e-01	**5.4569e-06**
	SD	2.2938e-01	2.5732e-01	**1.5901e-05**
F15	**Best**	-1.8013	-1.8013	**-1.8013**
	Worst	-1.8013	-3.5992	**-1.8013**
	Mean	-1.8013	-1.8013	**-1.8013**
	SD	2.1788e-14	3.2479e-1	**9.0336e-16**
F16	**Best**	-4.6887	-4.6887	**-4.6887**
	Worst	-4.453	-4.6887	**-3.8658**
	Mean	-4.5698	-4.507	**-4.5592**
	SD	7.8603e-02	3.2479e-01	**1.6929e-01**
F17	**Best**	-9.6176	-9.4438	**-9.6135**
	Worst	-8.437	-6.52	**-8.7814**
	Mean	-9.2283	-8.1286	**-9.2915**
	SD	3.0752e-01	7.4312e-01	**2.0661e-01**
F18	**Best**	4.0181e+00	5.16942e+01	**0**
	Worst	2.20118e+01	2.045579e+02	**0**
	Mean	9.4497e+00	1.354352e+02	**0**
	SD	3.8604e+00	3.22772e+01	**0**

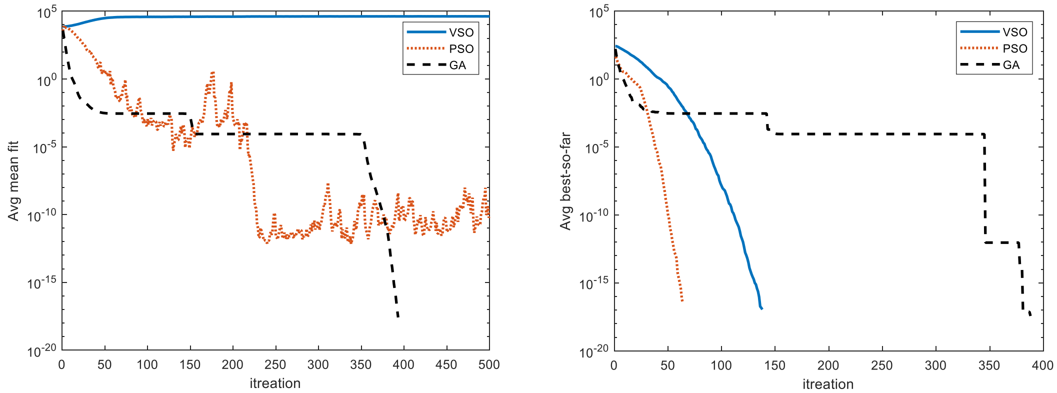


Fig. 23. Convergence rate comparison in best-so-far and average best-so-far of performance of VSO, PSO, GA for minimization benchmark functions F13.

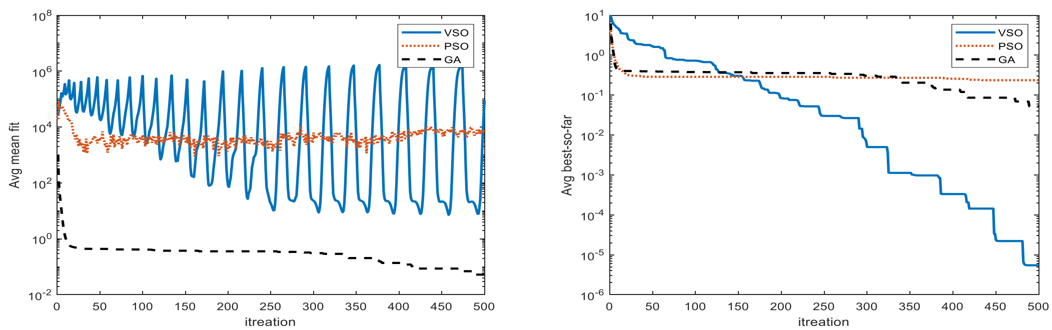


Fig. 24. Convergence rate comparison in best-so-far and average best-so-far of performance of VSO, PSO, GA for minimization benchmark functions F14.

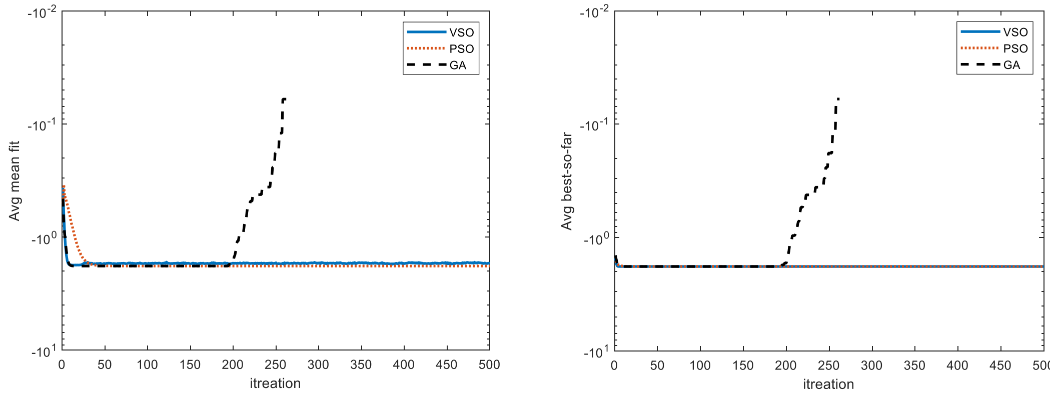


Fig. 25. Convergence rate comparison in best-so-far and average best-so-far of performance of VSO, PSO, GA for minimization benchmark functions F15.

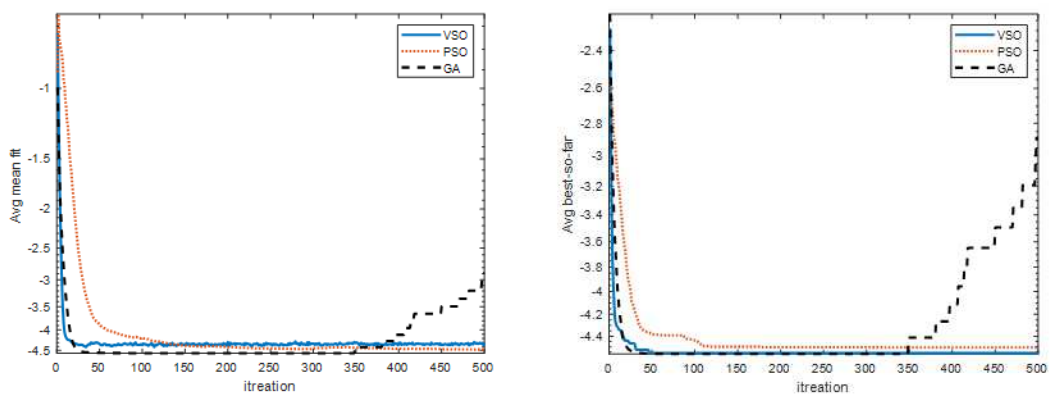


Fig. 26. Convergence rate comparison in best-so-far and average best-so-far of performance of VSO, PSO, GA for minimization benchmark functions F16.

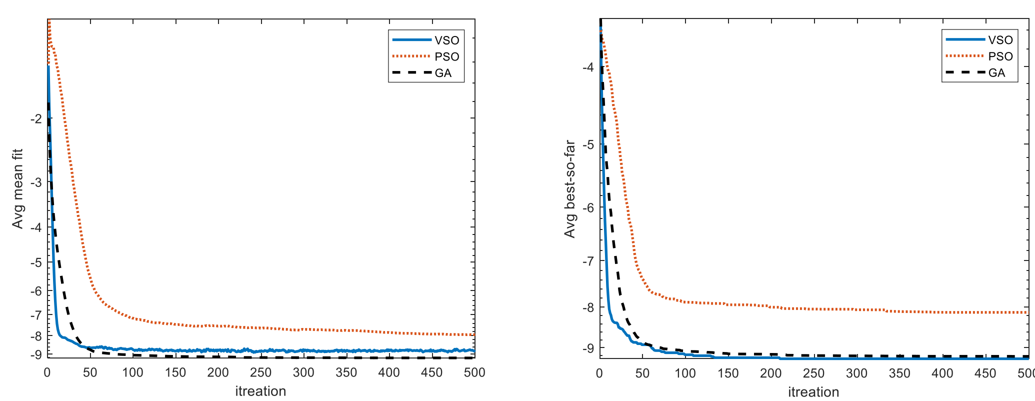


Fig. 27. Convergence rate comparison in best-so-far and average best-so-far of performance of VSO, PSO, GA for minimization benchmark functions F17.

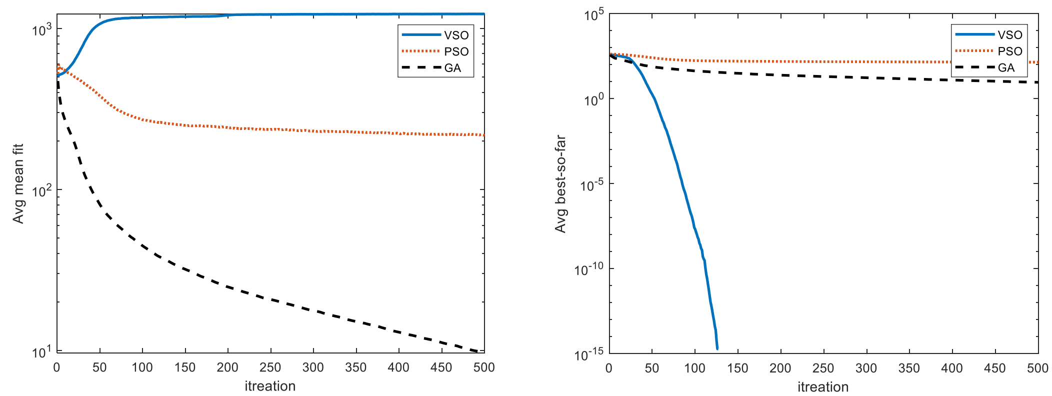


Fig. 28. Convergence rate comparison in best-so-far and average best-so-far of performance of VSO, PSO, GA for minimization benchmark functions F18.

Table 9. Neighborhood radius tuning for classical multimodal and separable benchmark functions.

The value of VSO neighborhood radius for the benchmark functions CEC 2014	
F13	**default (2.5)**
F14	**default**
F15	**default**
F16	**default**
F17	**default**
F18	**default**

Test 4. Multimodal and non-separable benchmark functions

Multimodal and non-separable benchmark functions are listed in table 10 and obtained result of evaluating those three algorithms are shown in table 11. As seen in table 11, VSO provides better results than GA and PSO algorithm for multimodal and non-separable functions in table 10. The performance of VSO, PSO and GA algorithm is shown in the following graphs in Fig 29-36 for function F19 to F26 respectively. Radius sense of neighbourhood for VSO algorithm is demonstrated in table 12.

Table 10. The description of classical multimodal and non-separable benchmark functions used in experimental test 4.

Function	Name	Expression	Range	d	F_{min}
F19	**Schaffer**	$x = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{((1 + 0.001(x_1^2 + x_2^2))}$	**[-100,100]**	**2**	**0**
F20	**Six hump**	$x = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^4 + x_1x_2 - 4x_2^2 + 4x_{24}$	**[-5,5]**	**2**	**-1.03163**
F21	**Boachevsky2**	$x = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$	**[-100,100]**	**2**	**0**
F22	**Boachevsy3**	$x = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$	**[-100,100]**	**2**	**0**
F23	**Shubert**	$x = (\sum_{j=1}^{5} j\cos(j+1)x_1 + j)(\sum_{j=1}^{5} j\cos((j+1)x_2 + j))$	**[-10,10]**	**2**	**-186.73**
F24	**Rosenbrock**	$x = \sum_{j=1}^{d-1} 100(x)\ (x_{j+1} - x_j^2)^2 + (x_j - 1)^2$	**[-30,30]**	**30**	**0**
F25	**Griewank**	$x = \frac{1}{4000}(\sum_{j=1}^{d}(x_j - 100)^2)) - (\pi_{j=1}^{d}\cos(\frac{x_j - 100}{\sqrt{j}})) + 1$	**[-600,600]**	**30**	**0**
F26	**Ackley**	$x = -20\exp(-0.2\sqrt{\frac{1}{d}\sum_{j=1}^{d} x_j^2}) - \exp(\frac{1}{3}\sum_{j=1}^{d}\cos(2\pi x_j))$	**[-32,32]**	**30**	**0**

Table 11. Statistical results obtained by GA, PSO and VSO, through 30 independent runs on classical multimodal and non-separable benchmark functions.

Function		GA	PSO	VSO
F19	**Best**	0	0	**0**
	Worst	4.3671e-02	1.2516e-05	**0**
	Mean	2.0688e-02	4.1721e-07	**0**
	SD	2.0688e-02	2.2851e-06	**0**
F20	**Best**	-1.0316	-1.0316	**-1.0316**
	Worst	-1.0316	-1.0316	**-1.0316**
	Mean	-1.0316	-1.0316	**-1.0316**
	SD	5.9741e-12	6.6486e-16	**5.7578e-16**
F21	**Best**	0	0	**0**
	Worst	1.2305e-06	0	**0**
	Mean	6.2032e-08	0	**0**
	SD	2.481e-07	0	**0**

F22	**Best**	0	0	**0**
	Worst	1.7217e-03	0	**0**
	Mean	5.989e-05	0	**0**
	SD	3.1405e-04	0	**0**
F23	**Best**	-186.7309	-186.7309	**-186.7309**
	Worst	-186.7297	-186.7309	**-186.7309**
	Mean	-186.7309	-186.7309	**-186.7309**
	SD	2.2879e-04	3.5796e-14	**2.3005e-14**
F24	**Best**	9.54433e+01	9.52854e+01	**1.6899e+00**
	Worst	1.9695155e+03	9.03661402e+04	**1.156452e+02**
	Mean	3.732739e+02	1.55688614e+04	**3.02094e+01**
	SD	3.868545e+02	3.39299888e+04	**2.50501e+01**
F25	**Best**	8.6524e-01	1.545e-06	**0**
	Worst	1.0272e+00	9.02544e+01	**0**
	Mean	9.9478e-01	6.0868e+00	**0**
	SD	3.8084e-02	2.28602e+01	**0**
F26	**Best**	3.2261e-01	9.7682e-01	**8.8818e-16**
	Worst	1.3954e+00	1.9963e+01	**4.4409e-15**
	Mean	5.9273e-01	6.4894e+00	**3.3751e-15**
	SD	2.392e-01	7.3965e+00	**1.6559e-15**

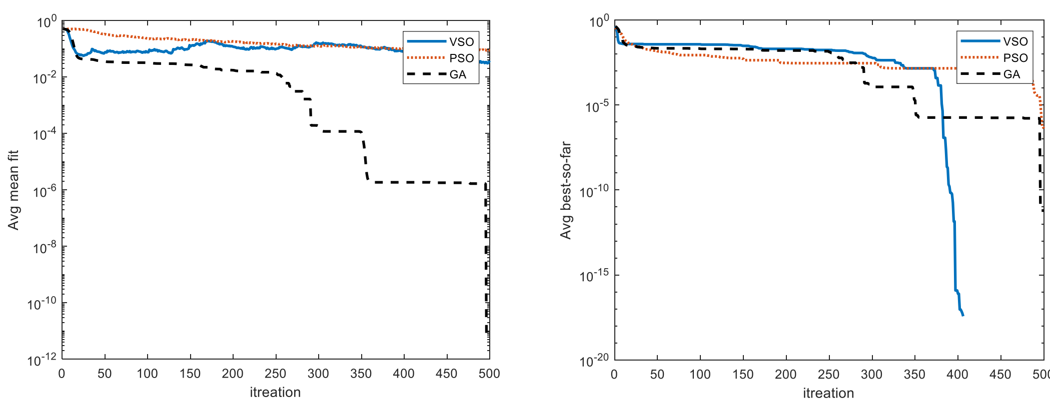


Fig. 29. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F19.

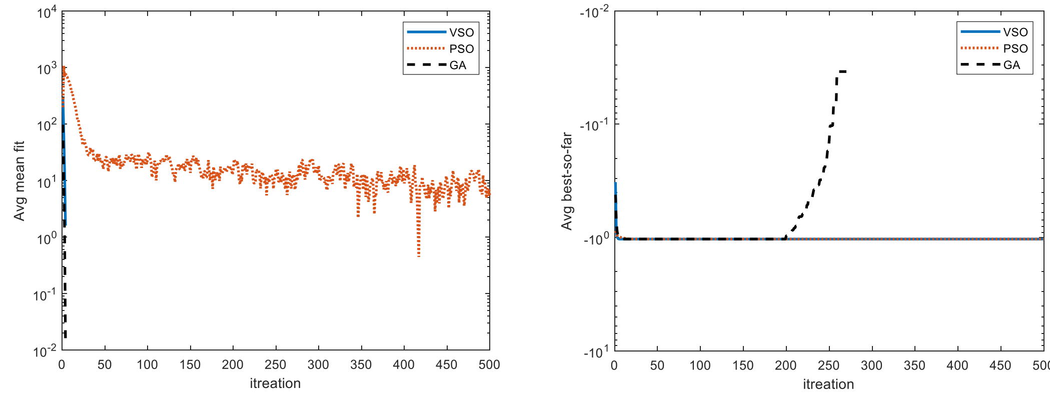


Fig. 30. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F20.

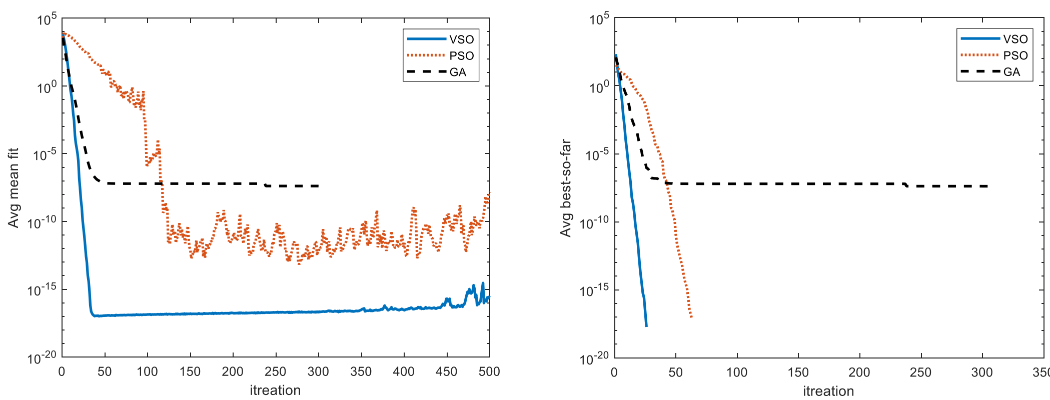


Fig. 31. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F21.

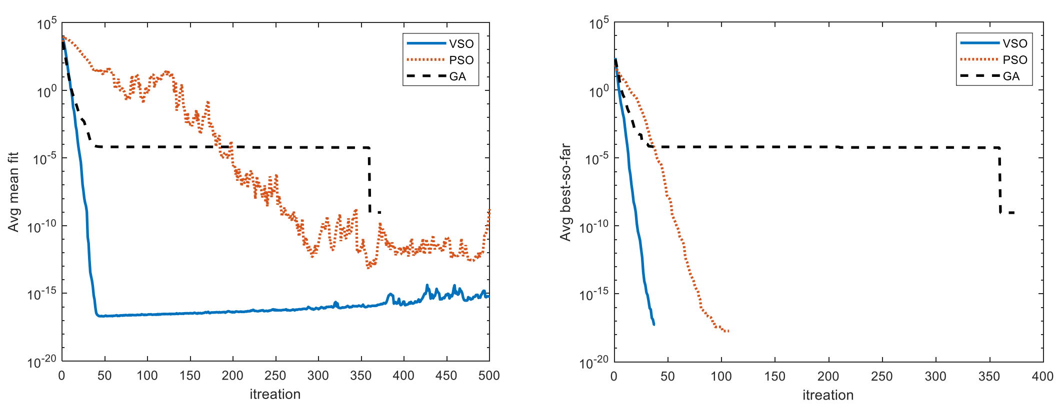


Fig. 32. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F22.

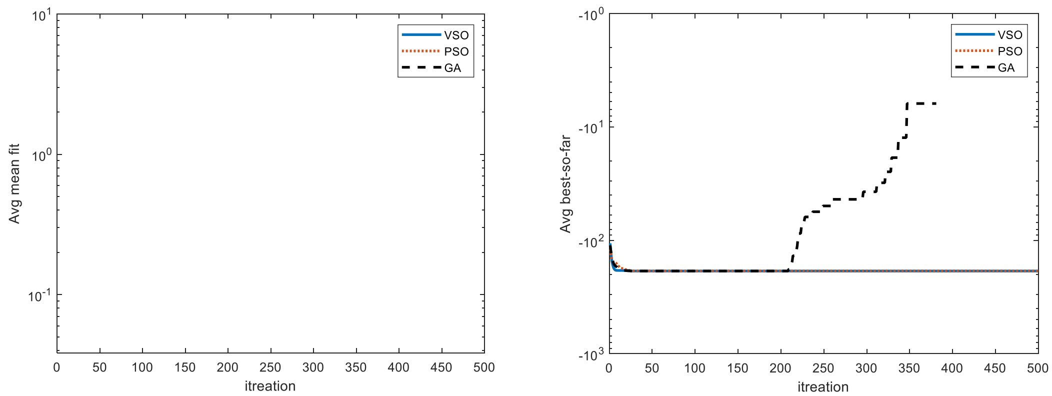


Fig. 33. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F23.

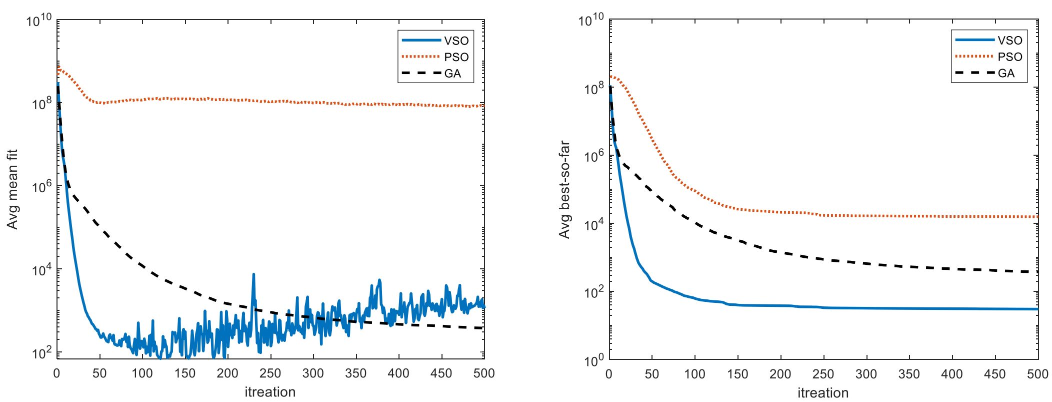


Fig. 34. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F24.

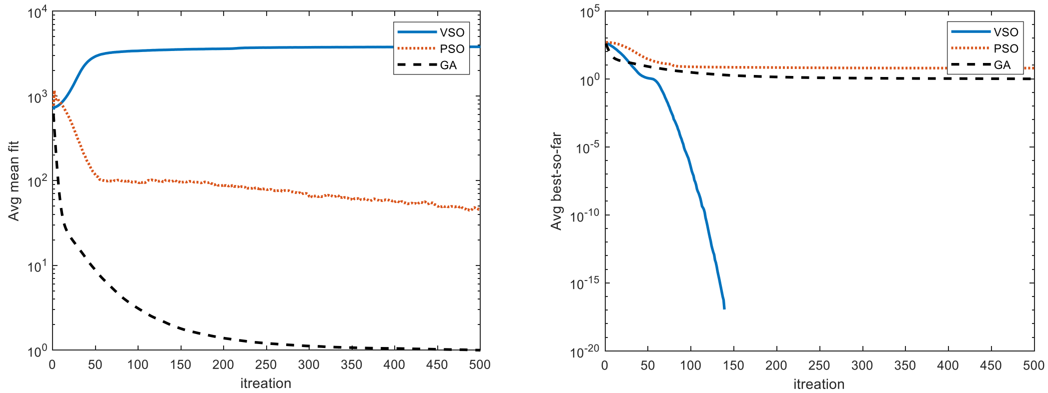


Fig. 35. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F25.

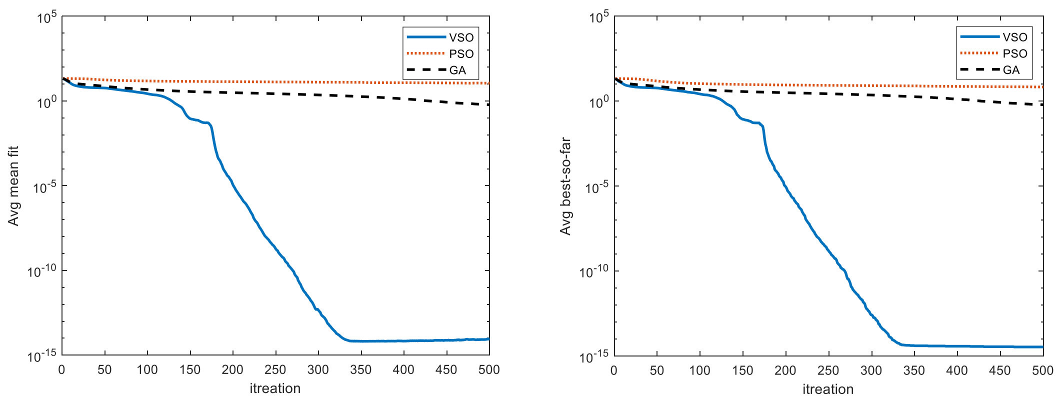


Fig. 36. Convergence rate comparison in best-so-far and average best-so-far of VSO, PSO and GA performance for minimization benchmark functions F26.

Table 12. Neighborhood radius tuning for classical multimodal and non-separable benchmark functions.

The value of VSO neighborhood radius for the benchmark functions CEC 2014	
F19	default (2.5)
F20	**default**
F21	**default**
F22	**default**
F23	**default**
F24	**default**
F25	**default**
F26	**default**

Table 13. Average mean results for all CEC 2014 standard benchmark functions

Function	VSO	PSO	GA
F1	3.2567e-17	2.8459e+00	7.0083e-03
F2	8.9897e-63	6.666667e+02	2.5543e+00
F3	5.5316-67	4.133334+02	2.9535e-01
F4	3.5032e-03	1.1005e+00	7.7226e-02
F5	5.1009e27	7.3731e-05	5.864e-04
F6	-1	-1	-1
F7	1.6113e-254	1.3405e-112	9.1267e-07
F8	1.3596e-02	2.9540607e+03	1.88101e+01
F9	7.713e-63	4.9738e+00	1.2108e+00
F10	5.3079e-39	2.80256e+01	4.2622e-01
F11	5.217e-65	4.1333336+e04	2.76083+01
F12	6.6667e-01	1.69777991e+04	6.0083e+00
F13	0	0	6.8172e-06
F14	5.4569e-06	2.3678e-01	3.5421e-01
F15	-1.8013	-1.8013	-1.8013
F16	-4.5592	-4.507	-4.5698
F17	-9.2915	-8.1286	-9.2283
F18	0	1.335959e+02	9.2268e+00
F19	0	4.1721e-07	2.0688e-02
F20	-1.0316	-1.0316	-1.0316
F21	0	0	6.2032e-08
F22	0	0	5.989e-05
F23	-186.7309	-186.7309	-186.7309
F24	3.02094e+01	1.55688614e+04	3.732739e+02
F25	7.9577e-02	6.0961e+00	9.9683e-01
F26	3.3751e-15	6.4894e+00	5.9273e-01
Total Best	26	0	0

Total best of Average mean results can be seen in Table 13 and this results show the proposed algorithm knockout the other two algorithm both in Multimodal and Unimodal (separable or non-separable) benchmark functions. Since VSO is limited to neighbourhood sense radius it can fall into the local optimum in separable multimodal functions because of their multi-picks and convex structures. This problem is solved with exploration and exploitation parameters so being cough in a local optimum is prevented.

Exploitation capability of VSO is configurable by water impact coefficients. In contrast exploration capability is configurable by wind impact coefficients. According to the principal of searching objective function, if objective function is more methodical then it is recommended to increase exploitation and for less methodical functions (more disturbed changes), it is better to increase exploitation. Concerning Table 13 it can be concluded that VSO has a better performance than GA and PSO regarding to the results.

7. Conclusion

In recent decades, various biological heuristic optimization algorithm has been developed. In this paper, a novel nature-inspired named Velella Swarm Optimization (VSO) algorithm is designed for optimization problems. The subgroups behavior and death of velellas in biological methodology is studied and their physics movement is demonstrated in accordance with similarity of sailboat movement physics. Our optimization algorithm is tested on low dimension of unconstraint benchmark functions (CEC 2014). Statistical analysis of global optimum solution shows that VSO algorithm have a high performance in comparison with other approaches. Population size is considered 50 and for higher values our results will be better compare to classical algorithms.

Funding: There is no funding for this paper.

Conflict of interests: There is no conflict of interests between authors of this article.

Availability of data and material: There is no data for this paper.

Availability of code: Code is available by request.

8.References

[1] E. O. Wilson. The relation between caste ratios and division of labor in the ant genus pheidole (hymenoptera: Formicidae). Behavioral Ecology and Sociobiology, 16(1):89–98, (1984).

[2] L. Chittka and J. Niven. Are bigger brains better? Current Biology, 19(21): R995–R1008, (2009).

[3] D. Nichols, The Oxford Book of Invertebrates. Oxford: Oxford University Press (1979).

[4] Ricketts, E., J. Calvin, J. Hedgpeth.. Between Pacific Tides (5th edition). Stanford, California: Stanford University Press (1997).

[5] N. Darke. The wrecking season, Boatshed films, (2004).

[6] T. O. Osborne W. A. Sheppe. Food Habits of peromyscus maniculatuson a California Beach. Journal of Mammalogy, Volume 52, Issue 4, 16 December, Pages 844–845, (1971).

[7] JenniferE.Purcell, Emmett.Clarkin, Thomas K. Doyle, Foods of Velella velella (Cnidaria: Hydrozoa) in algal rafts and its distribution in Irish seas, Springer Science+Business Media B.V. (2012).

[8] A. a.Jangah, a new heuristic optimization algorithm Based on swarm intelligence, M.Sc. Thesis, Shahid bahonar university of Kerman, Kerman, Iran, (2019).

[9] C. Blum, D. Merkle, Swarm Intelligence: Introduction and Applications. Natural Computing Series, Springer, Berlin, (2008)

[10] R.G. Reynolds. Version Space Controlled Genetic Algorithms. In Proceedings of the Second Annual Conference on Artificial Intelligence Simulation and Planning in High Autonomy Systems, pp. 6–14, April (1991).

[11] J. Kennedy, Swarm intelligence, in Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies, ed. by A.Y. Zomaya (Springer, Boston), pp. 187–219, (2006).

[12] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents. Trans. Syst. Man Cybern. B 26(1), 29–41 (1996)

[13] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel searchalgorithm, Journal Elsevier, Swarm and evolutionary Computation, Volume 44, Pages 148-175, February (2019).

[14] M. Yuan Cheng, D. Prayogo, A new metaheuristic optimization algorithm, Journal Elsevier, Computers & Structures Volume 139, Pages 98-112 15 July (2014).

[15] C. Blum, X. Li, Swarm intelligence in optimization, in Swarm Intelligence: Introduction and Applications, ed. by C. Blum, D. Merkle, pp. 43–85, Springer, Berlin (2008).

[16] E. Rashedi, H. Nezamabadipour and S. Saryazdi, "GSA: A Gravitational Search Algorithm", Information Sciences, Vol. 179, No. 13, pp. 2232-2248 (2009),. doi:10.1016/j.ins.2009.03.004

[17] L. Francis Sailing Downwind: Aerodynamic Performance Ofthe Velella Sail Printed in Great Britain © The Company of Biologists Limited (1991).

[18] J. Kimball, Physics of Sailing, thesis, CRC press Tylor and Francis Group Boca Raton, (2010).