Turk J Elec Eng & Comp Sci
() : –

doi:10.3906/elk-

A Novel Hybrid Global Optimization Algorithm Having Training Strategy: Hybrid Taguchi - Vortex Search Algorithm

Mustafa SAKA[1]*, Melih ÇOBAN[1], İbrahim EKE[2], Süleyman Sungur TEZCAN[1]
Müslüm Cengiz TAPLAMACIOĞLU[1]
[1]Department of Electrical and Electronics Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey,
[2]Department of Electrical and Electronics Engineering, Faculty of Engineering, Kırıkkale University, Kırıkkale, Turkey,

Received: .201 • **Accepted/Published Online:** .201 • **Final Version:** ..201

Abstract: In this paper, a novel Hybrid Taguchi-Vortex Search Algorithm (HTVS) is proposed for solving global optimization problems. Taguchi orthogonal approximation and Vortex Search Algorithm (VS) are hybridized in presenting method. In HTVS, orthogonal arrays in the Taguchi method are trained and obtained better solutions are used to find global optima in VS. Thus, HTVS has better relation between exploration and exploitation so it exhibits more powerful approximation to find global optimum value. Proposed HTVS algorithm is applied to sixteen well-known benchmark optimization test functions with different dimensions. The results are compared with the Taguchi Orthogonal Array Approximation (TOAA), Vortex Search Algorithm, Grey Wolf Optimizer (GWO), Sine-Cosine Algorithm (SCA), Moth-Flame Optimization Algorithm (MFO), Whale Optimization Algorithm (WOA) and Salp Swarm Algorithm (SSA). In order to compare the effectiveness of HTVS statistically, Wilcoxon Signed Rank Test (WSRT) is used in this study. Furthermore, HTVS is applied to two different real engineering problems having some constraints (tension/compression spring design and pressure vessel design). All obtained results are suggested that HTVS can find optimal or very close to optimal results. Moreover, it has good computational ability and fast convergence behavior as well.

Key words: Hybrid taguchi-vortex search algorithm,taguchi orthogonal arrays, vortex search algorithm, global optimization, engineering design problems with constraints

1. Introduction

Global optimization techniques have been very important in engineering applications such as electrical, mechanical engineering and also robotic etc. since the past. In the globalizing and modernizing world, engineering systems and their problems have become more complex. Everyday to solve complicated problems, many researchers have been searched and developed a lot of metaheuristic optimization methods in the literature.

Metaheuristic algorithms mostly gained inspiration from the nature. If these are wanted to be categorized, they can be considered in three main groups. First group can be classified as population based. The interactions of individuals in the community with each other are modeled in population based algorithms. These algorithms have different search strategies, for example hunting, seeking food etc. of swarms [1]. Grey Wolf Optimizer (GWO) [2], Particle Swarm Optimization (PSO) [3], Salp Swarm Algorithm (SSA) [4], Krill Herd Algorithm

*Correspondence: msaka@gazi.edu.tr

(KH) [5], The Whale Optimization Algorithm (WOA) [6], Artificial Bee Colony (ABC) [7] are some of developed algorithms in this group. Second group is classified as physical action based. While developing these algorithms, nature and physical events are taken into account and modeled. Gravitational Search Algorithm (GSA) [8], Big Bang Big Crunch Algorithm (BBBC) [9], Water Wave Optimization (WWO)[10], Black Hole (BH) [11] are some of second group algorithms. Third group can be classified as evolution based. Genetic Algorithm (GA) [12] may be well known and most popular algorithm of this group. Apart from GA, Differential Evolution (DE) [13] and Bio-geography Based Optimizer (BBO) [14] are other algorithms in this group. Mathematical based analytical methods are classified as fourth group. Dynamic Programming [15] and others [16–19] can be categorized in this group. Artificial intelligence based techiques can be classified as five group. These methods such as Artificial Neural Network [20] and Artificial Immune System [21] etc. are applied to different problems.

Metaheuristic optimization algorithms have exhibit good exploration and good exploitation. However, the convergence performances and systematic search states of these algorithms may be insufficient. Additionally, balance of exploration-exploitation may disrupted for complicated problems. In such cases, these algorithms can be plugged into local optimum points instead of global optimum points. For this reason, different metaheuristic algorithms have been combined with each other or different reinforcement techniques are added to metaheuristic algorithms. Thus, various advantages of algorithms are combined and their various disadvantages are eliminated. Opposition Based Learning (OBL) and Adaptive Differential Evolution (ADE) were combined and Partial Opposition Based Learning-Adaptive Differential Evolution (POBL-ADE) was developed in [22], Genetic Algorithm and Big Bang Big Crunch were hybridized and Hybrid Genetic Algorithm Big Bang Big Crunch algorithm (HGAB3C) was developed in [23], Sine Cosine Algorithm (SCA) and Multi Orthogonal Search Strategy (MOSS) were hybridized and Multi Orthogonal Sine Cosine Algorithm (MOSCA) was developed in [24], Particle Swarm Optimization and Grey Wolf Optimizer were hybridized and Hybrid Particle Swarm Optimization - Grey Wolf Optimizer (HPSOGWO) was developed in [25], Mean Variance Mapping Optimization was combined with Swarm Intelligence (MVMOSH) in [26], Self-adaptive Search Equation-based Artificial Bee Colony (SSEABC) in [27] and others [28–30] are some hybridized and strengthened algorithms.

In this study, two different methods are combined. First is Taguchi orthogonal array approximation (TOAA) [31]. This method is an experimental method and based on orthogonal arrays (OAs). The biggest advantage of OAs is that they can obtain good solutions with less numerical operations. But this method is not guaranteed the best results. Second is Vortex Search Algorithm [32, 33] and this algorithm can be thought in second group algorithms. VS has strong capability for numerical optimization problems and it needs to few user defined parameters. However, if the parameters are not selected properly, this algorithm can exhibit a non-optimal convergences. Additionally, if the problem to be solved is too complex, this algorithm can trapped to local minimum values like other metaheuristic algorithms. For this reasons, HTVS has been developed to eliminate the disadvantages of both TOAA and VS methods and create a more powerful and superior algorithm. Proposed HTVS algorithm shows better performance with lower initial candidate solutions and lower iteration number for all global optimization problems.

This paper organized three section after Introduction part. In second section, Taguchi Orthogonal Approximation, Vortex Search Algorithm and proposed Hybrid Taguchi-Vortex Search Algorithm are explained. In section three, sixteen optimization test function are defined and comparative results are given. Furthermore, WSRT statistics that confirm the effectiveness of HTVS are proved. Moreover, the results are obtained with HTVS are given for two different real engineering problems. The Conclusion is given in section 4.

2. Hybrid Taguchi-Vortex Search Algorithm

In this section, hybridized Taguchi orthogonal array approach and vortex search algorithm is defined. After than proposed HTVS algorithm is explained.

2.1. Taguchi Orthogonal Arrays

The Taguchi orthogonal arrays approximation method was developed by Genichi Taguchi [31]. Orthogonal arrays offer many advantages. First, OAs has fractional factorial characteristics [31]. It means that, desirable solutions can be obtained with fewer probability situation. For example, a ten parameters (considering that each parameter has 3 levels) problem, there are 3^{10} probability situations. However, with the use of OA, the probability situations are reduced to 27 [31]. Second, all possible states up to variable k are distributed equally in OAs [31]. Thus, the levels of these variables are analyzed equally. Finally, if some columns are removed from the OA, then the property of the OA does not be disrupted [31]. In this way, instead of using too many columns, up to k columns can be used. An example of OA are shown in Table 1.

Table 1. An orthogonal array OA(27,10,3,2)

Probability Situation	Parameters									
	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3	3
4	1	2	2	2	1	1	1	2	3	2
5	1	2	2	2	2	2	2	3	1	3
6	1	2	2	2	3	3	3	1	2	1
7	1	3	3	3	1	1	1	3	2	3
8	1	3	3	3	2	2	2	1	3	1
9	1	3	3	3	3	3	3	2	1	2
10	2	1	2	3	1	2	3	1	1	2
11	2	1	2	3	2	3	1	2	2	3
12	2	1	2	3	3	1	2	3	3	1
13	2	2	3	1	1	2	3	2	3	3
14	2	2	3	1	2	3	1	3	1	1
15	2	2	3	1	3	1	2	1	2	2
16	2	3	1	2	1	2	3	3	2	1
17	2	3	1	2	2	3	1	1	3	2
18	2	3	1	2	3	1	2	2	1	3
19	3	1	3	2	1	3	2	1	1	3
20	3	1	3	2	2	1	3	2	2	1
21	3	1	3	2	3	2	1	3	3	2
22	3	2	1	3	1	3	2	2	3	1
23	3	2	1	3	2	1	3	3	1	2
24	3	2	1	3	3	2	1	1	2	3
25	3	3	2	1	1	3	2	3	2	2
26	3	3	2	1	2	1	3	1	3	3
27	3	3	2	1	3	2	1	2	1	1

OAs are can be symbolized as OA(N, k, s, t). In this form, N represent row, k represent column (also optimized parameters number), s represent level and t represent strengt of an OA.

In Table 1, s and t are selected 3 and 2 respectively. This means that, every parameters have three level (s=3) values (1, 2, 3) and selected any two column (t=2) have different double combinations as row for example (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3) [31].

While determining the level values, initial solution of problem $x_{i=1}$ may chosen as midpoint of upper limit and lower limit. This midpoint is selected center of level value. For example, if s=3, this point equal to level 2. The other level values are determined by adding or subtracting LD_i (level difference) to level 2 [31]. LD is found from following equation [31]:

$$LD_{i=1} = \frac{maximum\ limit - minimum\ limit}{level + 1} \tag{1}$$

Here; i is iteration number (i=1, 2, 3, ...), x is candidate solution, *maximum* and *minimum limits* are boundary of problem. After defining of parameter level values, all probability situations are tried and results are calculated as in [31]. Optimal level values are found for every parameters and chosen center level values for next iterations. Every iteration LD value is decreased by reduced rate coefficient *(RR)* and this equation is given as follows [31]:

$$LD_{i+1} = LD_i \times RR \tag{2}$$

This periods is maintained until the finish condition are met. This criteria is defined below [31]:

$$\frac{LD_i}{LD_{i=1}} < target\ error\ value \tag{3}$$

2.2. VS Algorithm

This algorithm was developed based on the sample shape of the mixed liquids by Berat Dogan and Tamer Olmez [32]. This algorithm resembles nested circles in a 2D space when viewed from above [32]. The working system of the algorithm is depicted in Figure 1. In this figure, green point represents circle center, blue point represents best candidate solution and red points represent candidate solutions. The best solution in outer circle is memorized and it is placed to the center of the next inner circle for the next iteration.

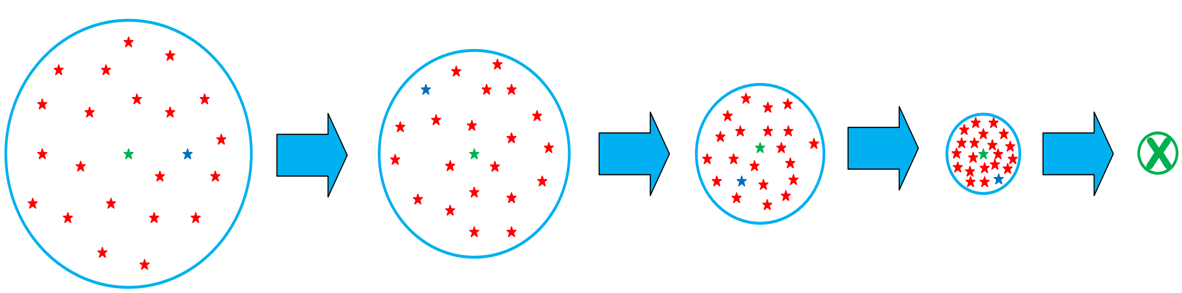

Figure 1. Systematic running of VS algorithm

The center of outermost circle (μ_0) is :

$$\mu_0 = \frac{maximum\ limit + minimum\ limit}{2} \tag{4}$$

The radius of this circle (σ_0) is:

$$\sigma_0 = \frac{max(maximum\ limit) - min(minimum\ limit)}{2} \tag{5}$$

Every candidate solutions boundaries are checked in every iteration. If they are not within boundaries, they relocated into the boundaries using following equation [32]:

$$cs_k = minimum\ limit + (maximum\ limit - minimum\ limit) \times rand \tag{6}$$

In here, k represents number of candidate solutions and rand is a random variable interval 0 and 1. Radius of circles (r_i) are decreased every iteration with inverse gamma function *(gammaincinv)* [32]:

$$r_i = \sigma_0 \times \frac{1}{x} \times gammaincinv(x, a_i) \tag{7}$$

In here a is shape parameter and x is constant value. a_i is reduced every iteration and given in as follows [32]:

$$a_i = a_0 - \frac{i}{MaxItr} \tag{8}$$

In here, i and $MaxItr$ represent iteration number and maximum iteration respectively. For contained all search area, a_0 is chosen 0 [32].

2.3. Proposed HTVS Algorithm

The proposed Hybrid Taguchi-Vortex Search (HTVS) Algorithm is formed by hybridizing with Taguchi Orthogonal Array Approximation and Vortex Search Algorithm. Orthogonal arrays may be preferred in population initialization stage [34]. OAs drastically reduce the number of probability situation during the process and so better results are achieved with fewer operations. Randomly generated initial candidate solutions are scattered using TOAA. Thus, TOAA is used in training of generating a candidate solution in proposed algorithm. Each candidate solution is distributed at certain equal intervals in the search space with TOAA. Thus, exploration phase of HTVS is enhanced. These candidate solutions are evaluated according to the probability situations of OA and so reinforced candidate solutions are found. These re-defined and improved candidate solutions are used in VS algorithm. Thus, exploitation phase and the convergence behaviour of HTVS are enhanced with good approximation features of VS. Thanks to these improvements, much better results can be obtained from HTVS using fewer solutions and less iterations. Moreover since the trained parameters are used in the VS during the entire iteration period, optimal or very close to optimal results are achieved with a fast convergence in HTVS.

The process of the HTVS algorithm is simply listed below:

(a) Randomly generated initial design parameters are scattered using TOAA,

(b) Probability situations are evaluated,

(c) Trained new design parameters are generated,

(d) These parameters are used in VS,

(e) Updating the parameters for next iterations.

Step1: Initializing of HTVS

Necessary definitions are made for use in the problem and algorithm such as problem boundaries, dimension, iteration number, reduce rate etc. The desired OA is constituted according to the problem dimensions. If problem dimension lower than OA columns, OA columns are selected as many as problem dimension. Thus, the number of OA columns are synchronized with the problem dimension. After than, candidate solutions are formed and controlled whether they are within limits.

Step2: Training of OA

Every level value is determined for each candidate solution. These level values are associated with OA. Objective values of probability situations in OA are calculated. Optimal level values are determined for every parameters. These values are chosen as best values for training of OA. After than, level difference is decreased by reduce rate coefficient and this process is continued until to reach the target error value. Thus, candidate solutions are improved.

Figure 2 shows an example of the training of OA. In this example, OA (9,4,3,2) is chosen to make it easier to understand. Red points show the candidate solution being trained. Other hollow points indicate the placement of this candidate solution in the OA. The optimum levels of the parameters are determined by controlling the entire probability situation in the OA. These determined levels are analyzed again with trained OA and so an improved candidate solution is found.

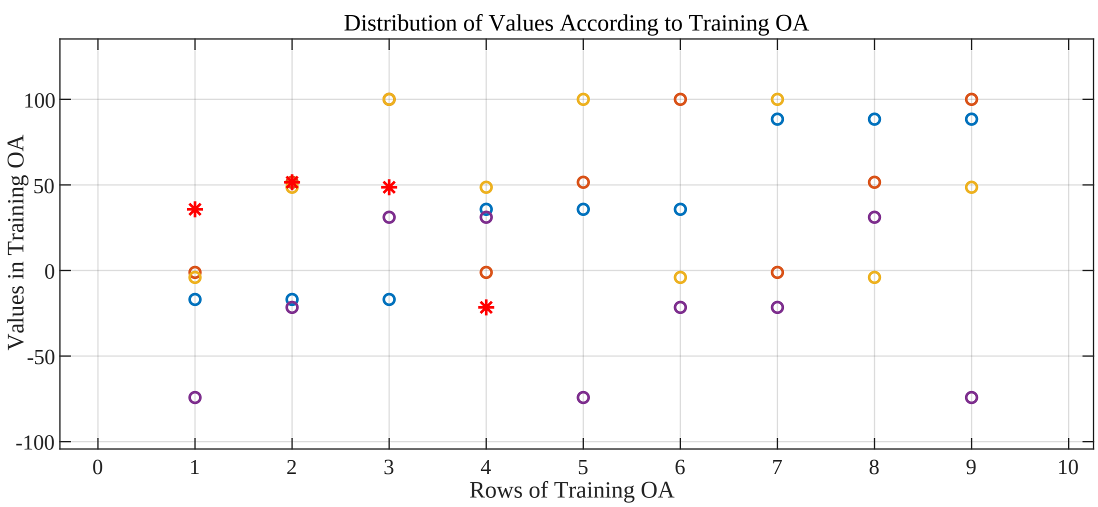

Figure 2. Illustration of training OA

Step3: Evaluation and Iteration

Improved candidate solutions are sending into the circle for evaluation. The best solution among of them is determined as the best solution of the iteration. If the best solution of the iteration better than the global best solution, the best solution of the iteration selected as the global best solution and memorized. After than, this solution is shifted to the center of the next circle. Then the radius of the circle is reduced. All these processes are continued until the number of iterations is equal to defined maximum iteration number. Detailed steps of HTVS algorithm are delineated in Figure 3.

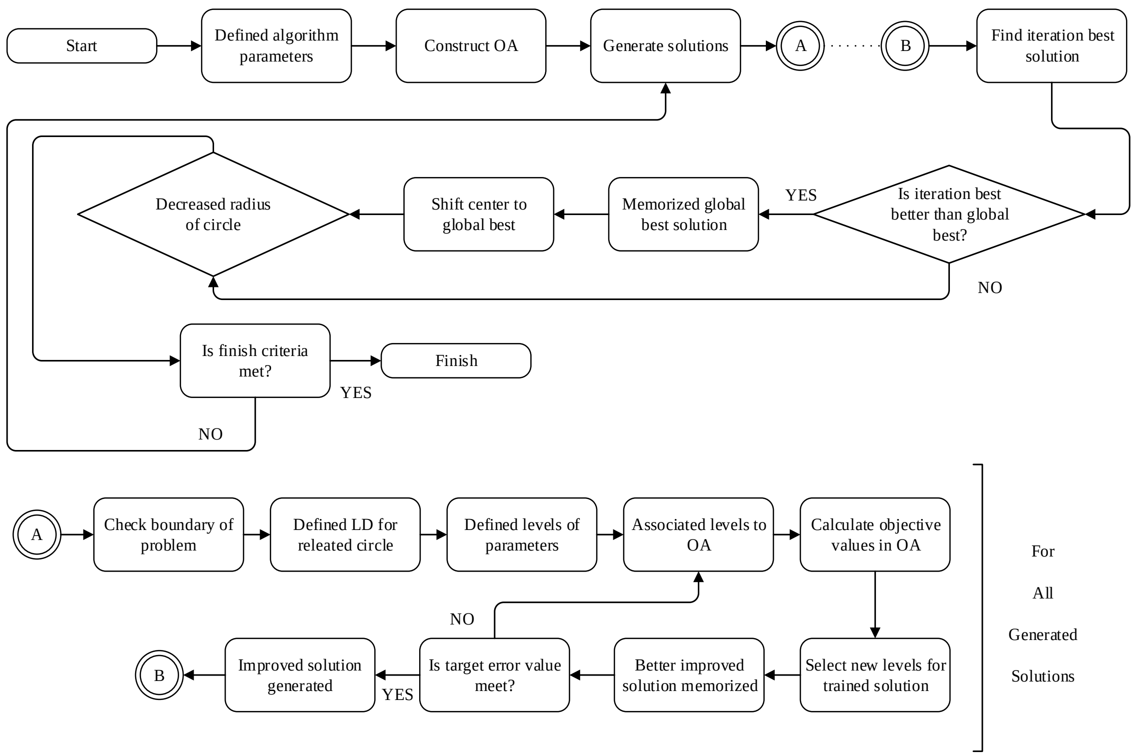

Figure 3. Flow diagram of HTVS

Algorithm 1: Pseudo-code for HTVS Algorithm

Begin Procedure HTVS Algorithm
Set parameters;
Generate Orthogonal Array related with the problem dimension;
for *Up to maximum iteration* **do**
    Check solutions boundary;
    for *Each solutions* **do**
        Defined Level Difference;
        while *Target error value* **do**
            Designate solutions level;
            Correlate levels to Orthogonal Array;
            Evaluate probability situations;
            Describe optimal level values;
            Find improved new solutions;
        end
    end
    Determine iteration best;
    if *iteration best value* $<$ *global best value* **then**
        global best value = iteration best value;
    end
    Reduce radius;
end
End Procedure

---

3. Experimental Study

In this part, two different experiments have been carried out to examine the performance of optimization algorithms. The first experiment has been realized on benchmark functions (BFs), the second experiment on real engineering problems in the literature.

3.1. Experimental test 1

In this part, 16 BFs have been utilized to examine the performance and efficiency of the improved HTVS algorithm. BFs have been selected from [34]. Six optimization algorithms (GWO [2], SSA [4], WOA [6],VS [32], SCA [35] and MFO [36]) used in the literature have been utilized to affirm the validity and performance of the proposed HTVS algorithm.

3.1.1. Benchmark Functions and Algorithm Settings

The BFs utilized in the first experiment are listed in Table 2. In this table, the limits of the variables used for each function, the equations used in the calculation, the type of the function and the size information are given. Additional information and parameters for Penalized, Penalized2 $u(x_i, 10, 100, 4)$ and Foxholes (a_{ij}) functions in Table 2 are as defined in [32]. If a function has a single optimum point in a certain range, it is called the uni-modal function. If also a function has many local optimum points, it is called a multi-modal function. Separability is associated with the concept of mutual relationship between the variables of the function. Nonseparable functions cannot be expressed in this way because there is a relationship between variables . Optimizing non-separable functions is harder than optimizing separable functions [37].

Table 2. Chosen BFs (n: Dimension, T: Type, U: Uni-modal, M: Multi-modal, S: Separable, N: Nonseparable).

Function					
No.	Range	n	T	Name	Formulation
Fnc1	[-100, 100]	30	US	Sphere	$f(y) = \sum_{j=1}^{n}(y_j)^2$
Fnc2	[-10, 10]	30	UN	Schwefel 2.22	$f(y) = \sum_{j=1}^{n}\lvert y_j\rvert + \prod_{j=1}^{n}\lvert y_j\rvert$
Fnc3	[-100, 100]	30	UN	Schwefel 1.2	$f(y) = \sum_{j=1}^{n}(\sum_{k=1}^{j} y_k)^2$
Fnc4	[-30, 30]	30	UN	Rosenbrock	$f(y) = \sum_{j=1}^{n}[100(y_{j+1} - y_j^2)^2 + (y_j - 1)^2]$
Fnc5	[-100, 100]	30	US	Step	$f(y) = \sum_{j=1}^{n}(\lvert y_j + 0.5\rvert)^2$
Fnc6	[-1.28, 1.28]	30	US	Quartic	$f(y) = \sum_{j=1}^{n}(jy_j)^4 + random[0, 1)$
Fnc7	[-500, 500]	30	MS	Schwefel	$f(y) = \sum_{j=1}^{n} -y_j \sin(\sqrt{y_j})$
Fnc8	[-5.12, 5.12]	30	MS	Rastrigin	$f(y) = \sum_{j=1}^{n}[(y_j)^2 - 10\cos(2\pi y_j) + 10]$
Fnc9	[-32, 32]	30	MN	Ackley	$f(y) = -20\exp\Big(-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n} y_j^2}\Big) - \exp\Big(\frac{1}{n}\sum_{j=1}^{n}\cos(2\pi y_j)\Big) + 20 + \exp(1)$
Fnc10	[-600, 600]	30	MN	Griewank	$f(y) = \frac{1}{4000}\sum_{j=1}^{n}(y_j)^2 - \prod_{j=1}^{n}\cos(\frac{y_j}{\sqrt{j}}) + 1$
Fnc11	[-50, 50]	30	MN	Penalized	$f(y) = \frac{\pi}{n}\{10\sin(\pi z_1)^2 + \sum_{j=1}^{n-1}(z_j - 1)^2[1 + 10\sin(\pi z_{j+1})^2]\} + \sum_{j=1}^{n} u(y_j, 10, 100, 4), z_j = 1 + \frac{1}{4}(y_j + 1)$
Fnc12	[-50, 50]	30	MN	Penalized2	$f(y) = 0.1\{\sin(\pi y_1)^2 + \sum_{j=1}^{n-1}(y_j - 1)^2[1 + \sin(3\pi y_{j+1})^2] + (y_n - 1)^2[1 + \sin(2\pi y_n)^2]\} + \sum_{j=1}^{n} u(y_j, 10, 100, 4)$
Fnc13	[-65.536, 65.536]	2	MS	Foxholes	$f(y) = \Big[\frac{1}{500} + \sum_{k=1}^{25}\frac{1}{k + \sum_{j=1}^{2}(y_j - a_{jk})^6}\Big]$
Fnc14	[-5, 5]	2	MN	Six Hump Camel Back	$f(y) = 4y_1^2 - 2.1y_1^4 + \frac{1}{3}y_1^6 + y_1y_2 - 4y_2^2 + y_2^4$
Fnc15	[-5, 10]&[0, 15]	2	MS	Branin	$f(y) = \big(y_2 - \frac{5.1}{4\pi^2}y_1^2 + \frac{5}{\pi}y_1 - 6\big)^2 + 10(1 - \frac{1}{8\pi})\cos(y_1) + 10$
Fnc16	[-2, 2]	2	MN	GoldStein-Price	$f(y) = [1 + (y_1 + y_2 + 1)^2(19 - 14y_1 + 3y_1^2 - 14y_2 + 6y_1y_2 + 3y_2^2)][30 + 3(2y_1 - 3y_2)^2(18 - 32y_1 + 12y_1^2 + 48y_2 - 36y_1y_2 + 27y_2^2)]$

Population size has been determined as 50 and iteration number is 1000 for each compared algorithm. 30 independent runs have been executed each test function. The best, worst, mean and standard deviation (S.D.) parameters have been obtained from these runs.

3.1.2. Statistical Analysis

In the first experiment, proposed HTVS algorithm is compared to TOAA, VS, GWO, SCA, MFO, WOA and SSA algorithms. Statistical values of TOAA, VS and GWO are listed in Table 3. Also, statistical values of SCA, MFO, WOA and SSA are given in Table 4. It can be clearly seen that from this table, HTVS algorithm obtained better mean value and lower standard deviation value in other comparison functions except Fnc1, Fnc2 and Fnc4. Mean and standard deviation(S.D.) values can be used as an indicator about the robustness of the algorithm. By examining the best and worst values, an idea about the quality of the optimization algorithm can be obtained [34]. Although these values provide a rough idea, pairwise statistical test is often used for a stronger comparison. Therefore, WSRT has been chosen to perform a pairwise statistical test. HTVS and other selected algorithms have been run different 30 times for each function. WSRT has been performed using the results obtained from this process. The obtained statistical pair-wise results are illustrated in Table 5.

In the WSRT, it can be understood which of the two algorithms compared using the hypothesis test is superior. Two hypotheses can be determined as a (H_0) and (H_1). The (H_0) hypothesis means that there is no difference between the compared pairs. Unlike (H_0), the (H_1) hypothesis means that there is a difference. In Table 5 When the h value is equal to 0, it is seen that there is no critical contrast between the compared two algorithms. When $h = 1$, a major difference is observed between the two compared algorithms. In addition, as $p - value$ which is the probability of observing a test statistic, decreases, the similarity of the two algorithms decreases. In WSRT, statistical significance value is determined as $a = 0.05$. When $p - value$ used to determine whether algorithms have superiority to each other is less than a, it can be said that two compared algorithms are statistically distinctive from each other at 95% confidence level. In this table, 'h=1+' indicates situations where the zero hypothesis is refused and the HTVS performs statistically predominant in the WSRT at 95% significance level; 'h=1-'denotes states where the (H_0) is refused and the HTVS algorithm performed lower performance; and '$h = 0$' denotes states that are not critical contrast between the two algorithms. The nineteenth line and last line of Table 5 demonstrates the total number of three statistically major states $(+/=/-)$ in the comparison between pairs.The $+$ sign demonstrates that the HTVS algorithm is superior to the compared algorithm, the $=$ sign demonstrates that the HTVS algorithm draws with the compared algorithm, the $-$ sign demonstrates that the HTVS algorithm is worse than the compared algorithm. In each comparison, the HVTS algorithm outperformed all the compared algorithms because the number of $+$ signs is greater than the $=$ and $-$ sign. The superiority of the HVTS algorithm is more dominant when it compared to GWO, SCA, MFO, SSA algorithms and less dominant when it compared to VS and WOA algorithms.

3.1.3. Convergence Analysis

Convergence analysis has been performed to illustrated how the proposed HTVS algorithm converges to the solution. A total of 4 convergence graphics have been obtained from each function type (MS, US, MN, UN). The graphics has been drawn using information about average solutions 30 different runs for 1000 iterations. As seen in the Figure 4, convergence graph is drawn for each function type. It has been viewed that the HTVS algorithm is more competitive than other algorithms. HTVS algorithm presents one convergence behavior while optimizing test functions. HTVS is generally very close to optimum value in the first few iterations.

Table 3. Statistical Results for 30 Runs

No.	Min.		HTVS	TOAA	VS	GWO
Fnc1	0	Mean	3.0368E-147	0	8.7754E-68	3.2601E-70
		S.D.	1.8419E-147	0	4.7513E-67	7.4302E-70
		Best	4.4734E-157	0	3.1944E-90	1.2881E-72
		Worst	7.0213E-147	0	2.6033E-66	3.9161E-69
Fnc2	0	Mean	5.8818E-74	0	2.7291E-36	3.9362E-41 7
		S.D.	1.9096E-74	0	1.3341E-35	5.8096E-41
		Best	1.7604E-74	0	3.0048E-47	3.7870E-42
		Worst	8.1700E-74	0	7.3306E-35	3.1475E-40
Fnc3	0	Mean	0	0	9.7719E+03	3.4143E+03
		S.D.	0	0	2.3343E+04	3.1775E+03
		Best	0	0	1.8897E-90	2.3814E+02
		Worst	0	0	8.2809E+04	3.1775E+03
Fnc4	0	Mean	12.409	3758.975	1.5679E-33	26.4631
		S.D.	10.4425	3758.975	2.0959E-33	0.8077 5
		Best	7.6498E-31	3758.975	0	25.1885
		Worst	21.9412	3758.975	4.2762E-33	28.51
Fnc5	0	Mean	0	0	0	0.3583
		S.D.	0	0	0	0.2907
		Best	0	0	0	9.3673E-06
		Worst	0	0	0	1.0043
Fnc6	0	Mean	8.8371E-07	0.08	1.6124E-04	4.1014E-04
		S.D.	8.5843E-07	0.064	1.5318E-04	2.1143E-04
		Best	7.8925E-08	6.5572E-04	1.7327E-05	6.6443E-05
		Worst	4.2453E-06	0.2567	5.3662E-04	0.0011
Fnc7	-12569.5	Mean	-1.2569E+04	-3686.29	-1.2569E+04	-6.3757E+03
		S.D.	2.5502E-12	-3686.29	1.8501E-12	8.6519E+02
		Best	-1.2569E+04	-3686.29	-1.2569E+04	-7.6185E+03
		Worst	-1.2569E+04	-3686.29	-1.2569E+04	-3.2684E+03
Fnc8	0	Mean	0	0	0	0.1504
		S.D.	0	0	0	0.8235
		Best	0	0	0	0
		Worst	0	0	0	4.5107
Fnc9	0	Mean	4.4409E-15	8.88E-16	8.8818E-16	1.3204E-14
		S.D.	0	8.88E-16	0	3.1959E-15
		Best	4.4409E-15	8.88E-16	8.8818E-16	7.9936E-15
		Worst	4.4409E-15	8.88E-16	8.8818E-16	2.2204E-14
Fnc10	0	Mean	0	2.82E-144	0	0.002
		S.D.	0	2.82E-144	0	0.0048
		Best	0	2.82E-144	0	0
		Worst	0	2.82E-144	0	0.0157
Fnc11	0	Mean	1.5705E-32	0.7519	1.5705E-32	0.0284
		S.D.	5.5674E-48	0.7519	5.5674E-48	0.0154
		Best	1.5705E-32	0.7519	1.5705E-32	0.0065
		Worst	1.5705E-32	0.7519	1.5705E-32	0.072
Fnc12	0	Mean	1.3498E-31	0.0443	1.3498E-31	0.3097
		S.D.	6.6809E-47	0.0443	6.6809E-47	0.1715
		Best	1.3498E-31	0.0443	1.3498E-31	2.1270E-05
		Worst	1.3498E-31	0.0443	1.3498E-31	0.7138
Fnc13	1	Mean	0.998	0.998604	0.9991	2.8953
		S.D.	1.1292E-16	0.998604	0.0054	3.2636
		Best	0.998	0.998604	0.998	0.998
		Worst	0.998	0.998604	1.0273	10.7632
Fnc14	-1.0316	Mean	-1.0316	-1.03163	-1.0035	-1.0316
		S.D.	6.7752E-16	-1.03163	0.0335	2.3754E-09
		Best	-1.0316	-1.03163	-1.0316	-1.0316
		Worst	-1.0316	-1.03163	-0.9108	-1.0316
Fnc15	0.398	Mean	0.3979	0.397887	0.4028	0.3979
		S.D.	0	0.397887	0.0093	4.4472E-05
		Best	0.3979	0.397887	0.3979	0.3979
		Worst	0.3979	0.397887	0.4472	0.3981
Fnc16	3	Mean	3	99	3.4535	3
		S.D.	3.1939E-16	99	0.5875	2.1318E-06
		Best	3	99	3.0003	3
		Worst	3	99	5.6076	3

Table 4. Statistical Results for 30 Runs

No.	Min.		SCA	MFO	WOA	SSA
Fnc1	0	Mean	3.0000E-03	2.0000E+03	1.2955E-173	8.8119E-09
		S.D.	5.5000E-03	4.0684E+03	0	1.8119E-09
		Best	1.0119E-07	2.2826E-06	1.1802E-187	6.1125E-09
		Worst	2.0600E-02	1.0000E+04	1.7212E-172	1.3440E-08
Fnc2	0	Mean	6.1720E-06	27.3334	2.1033E-108	0.5467
		S.D.	1.5267E-05	17.7983	1.0678E-107	0.7322
		Best	7.5984E-10	2.1225E-04	5.4601E-120	5.0879E-04
		Worst	7.6365E-05	70.0000	5.8555E-107	3.4807
Fnc3	0	Mean	3.4143E+03	1.6118E+04	1.0251E+04	35.2190
		S.D.	3.1775E+03	1.0657E+04	6.5943E+03	22.5824
		Best	2.3814E+02	273.8736	725.1963	9.2185
		Worst	1.2813E+04	4.5013E+04	2.7663E+04	106.1447
Fnc4	0	Mean	66.7187	1.2820E+04	26.5654	49.1718
		S.D.	80.2239	3.0839E+04	0.2899	45.3955
		Best	28.0344	7.3142	26.0486	19.9603
		Worst	327.5366	9.0081E+04	27.0279	200.2024
Fnc5	0	Mean	4.2785	1.3267E+03	0.0044	8.8659E-09
		S.D.	0.4562	4.3123E+03	0.0022	1.8151E-09
		Best	3.6326	2.6010E-06	8.9371E-04	5.6152E-09
		Worst	5.8445	1.9801E+04	0.0117	1.1868E-08
Fnc6	0	Mean	0.0264	3.8282	8.5780E-04	0.0585
		S.D.	0.0197	8.0671	9.1209E-04	0.0289
		Best	0.0046	0.0301	1.5986E-05	0.0183
		Worst	0.0731	40.3245	0.0040	0.1494
Fnc7	-12569.5	Mean 3	-3.9844E+03	-8.6512E+03	-1.1384E+04	-7.4102E+03
		S.D.	2.7826E+02	861.1861	1.4685E+03	835.0389
		Best	-4.6838E+03	-1.0571E+04	-1.2569E+04	-9.0163E+03
		Worst	-3.6256E+03	-6.8511E+03	-8.2506E+03	-5.9019E+03
Fnc8	0	Mean	16.8594	137.2066	1.8948E-15	43.9440
		S.D.	20.7797	36.3597	1.0378E-14	13.4026
		Best	8.1685E-06	73.6266	0	19.8992
		Worst	72.3418	205.2448	5.6843E-14	76.6117
Fnc9	0	Mean	1.2487E+01	11.6837	3.0198E-15	1.6068
		S.D.	9.4183	8.4275	2.5721E-15	1.1970
		Best	3.5559E-04	6.7085E-04	8.8818E-16	1.9931E-05
		Worst	2.0311E+01	19.9630	7.9936E-15	3.6819
Fnc10	0	Mean	0.2481	18.0233	0.0026	0.0090
		S.D.	0.2188	36.6376	0.0143	0.0093
		Best	3.8536E-06	7.7973E-06	0	1.9672E-08
		Worst	0.5971	90.1836	0.0783	0.0344
Fnc11	0	Mean	1.6176	0.4448	0.0014	2.9985
		S.D.	2.877	1.2554	0.0029	2.1077
		Best	0.362	1.6537E-05	1.7603E-04	0.1086
		Worst	10.9041	6.7120	0.0149	9.8207
Fnc12	0	Mean	3.8187	1.3669E+07	0.0589	0.0069
		S.D.	4.1862	7.4867E+07	0.0706	0.0076
		Best	2.1981	2.4950E-05	0.0030	3.4119E-10
		Worst	19.1298	4.1006E+08	0.2875	0.0308
Fnc13	1	Mean	1.1965	1.6238	2.1729	1.0311
		S.D.	0.6054	1.4774	2.9739	0.1815
		Best	0.998	0.9980	0.9980	0.9980
		Worst	2.9821	5.9288	10.7632	1.9920
Fnc14	-1.0316	Mean	-1.0316	-1.0316	-1.0316	-1.0316
		S.D.	9.6489E-06	0	7.1907E-12	6.7921E-15
		Best	-1.0316	-1.0316	-1.0316	-1.0316
		Worst	-1.0317	-1.0316	-1.0316	-1.0316
Fnc15	0.398	Mean	0.399	0.3979	0.3979	0.3979
		S.D.	0.0015	1.1292E-16	2.0882E-07	1.2085E-14
		Best	0.3979	0.3979	0.3979	0.3979
		Worst	0.4036	0.3979	0.3979	0.3979
Fnc16	3	Mean	3	3.0000	3.0000	3.0000
		S.D.	1.9622E-05	2.5657E-15	2.9100E-06	6.9564E-14
		Best	3	3.0000	3.0000	3.0000
		Worst	3.0001	3.0000	3.0000	3.0000

3.2. Experimental Test 2 (Real Engineering Problems)

In this part, tension / compression spring design(T/CSD) and pressure vessel design(PVD) problems have been solved with proposed HTVS algorithm. HTVS has been run with population sizes 20 and 500 iterations in two problems. The performance and applicability of the proposed HTVS algorithm has been compared with the solutions of other algorithms in the literature. The results of the compared algorithms have been taken directly from literature. As a result of the comparison, the feasibility of the HTVS algorithm has been confirmed.The

Table 5. Wilcoxon Signed - Rank Test Results

No		F1	F2	F3	F4	F5	F6	F7	F8
HTVS vs. VS	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.00×10^{0}	1.73×10^{-6}	1.00×10^{0}	1.00×10^{0}
	h	1+	1+	1+	1-	0	1+	0	0
	Tp	0	0	0	465	0	0	0	0
	Tn	465	465	465	0	0	465	0	0
HTVS vs. TOAA	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.00×10^{-0}	1.50×10^{-6}	1.00×10^{0}	1.73×10^{-6}	4.32×10^{-8}	1.00×10^{0}
	h	1-	1-	0	1+	0	1+	1+	0
	Tp	465	465	0	0	0	0	0	0
	Tn	0	0	0	465	0	465	465	0
HTVS vs. GWO	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	2.50×10^{-1}
	h	1+	1+	1+	1+	1+	1+	1+	0
	Tp	0	0	0	0	0	0	0	0
	Tn	465	465	465	465	465	465	465	6
HTVS vs. SCA	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}
	h	1+	1+	1+	1+	1+	1+	1+	1+
	Tp	0	0	0	0	0	0	0	0
	Tn	465	465	465	465	465	465	465	465
HTVS vs. MFO	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	11.73×10^{-6}
	h	1+	1+	1+	1+	1+	1+	1+	1+
	Tp	0	0	0	0	0	0	0	0
	Tn	465	465	465	465	465	465	465	465
HTVS vs. WOA	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.00×10^{0}
	h	1-	1-	1+	1+	1+	1+	1+	0
	Tp	465	465	0	0	0	0	0	0
	Tn	0	0	465	465	465	465	465	1
HTVS vs. SSA	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}
	h	1+	1+	1+	1+	1+	1+	1+	1+
	Tp	0	0	0	0	0	0	0	0
	Tn	465	465	465	465	465	465	465	465
No		F9	F10	F11	F12	F13	F14	F15	F16
HTVS vs. VS	p-val.	1.73×10^{-6}	1.00×10^{-0}	4.32×10^{-8}	1.00×10^{-0}	6.63×10^{7}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}
	h	1-	0	1+	0	1-	1+	1+	1+
	Tp	465	0	0	0	455	0	165	0
	Tn	0	0	465	0	1	465	300	465
HTVS vs. TOAA	p-val.	4.32×10^{-8}	4.32×10^{-8}	4.32×10^{-8}	4.32×10^{-8}	2.03×10^{-7}	1.00×10^{-0}	1.00×10^{0}	4.32×10^{-8}
	h	1-	1+	1+	1+	1-	0	0	1+
	Tp	465	0	0	0	459	0	0	0
	Tn	0	465	465	465	0	0	0	465
HTVS vs. GWO	p-val.	1.04×10^{-6}	6.25×10^{-2}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}
	h	1+	0	1+	1+	1-	1+	1-	1+
	Tp	0	0	0	0	374	0	464	0
	Tn	465	15	465	465	91	465	1	465
HTVS vs. SCA	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}
	h	1+	1+	1+	1+	1-	1+	1-	1+
	Tp	0	0	0	0	459	0	294	0
	Tn	465	465	465	465	6	465	171	465
HTVS vs. MFO	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.46×10^{-6}	1.00×10^{-0}	1.00×10^{0}	5.00×10^{-1}
	h	1+	1+	1+	1+	1-	0	0	0
	Tp	0	0	0	0	420	0	0	59
	Tn	465	465	465	465	21	0	0	0
HTVS vs. WOA	p-val.	1.56×10^{-4}	1.00×10^{-0}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	2.55×10^{-6}	1.73×10^{-6}	1.73×10^{-6}
	h	1-	0	1+	1+	1-	1-	1-	1+
	Tp	360	0	0	0	444	464	465	0
	Tn	10	1	465	465	21	0	0	465
HTVS vs. SSA	p-val.	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	1.73×10^{-6}	3.18×10^{-7}	2.00×10^{-3}	2.50×10^{-5}	1.69×10^{-6}
	h	1+	1+	1+	1+	1-	1-	1-	1-
	Tp	0	0	0	0	455	255	437	437
	Tn	465	465	465	465	1	0	0	28
	+/=/-	HTVS vs. VS 8/5/3	HTVS vs. TOAA 7/5/4	HTVS vs. GWO 12/2/2	HTVS vs. SCA 14/0/2	HTVS vs. MFO 12/3/1	HTVS vs. WOA 8/2/6	HTVS vs. SSA 12/0/4	

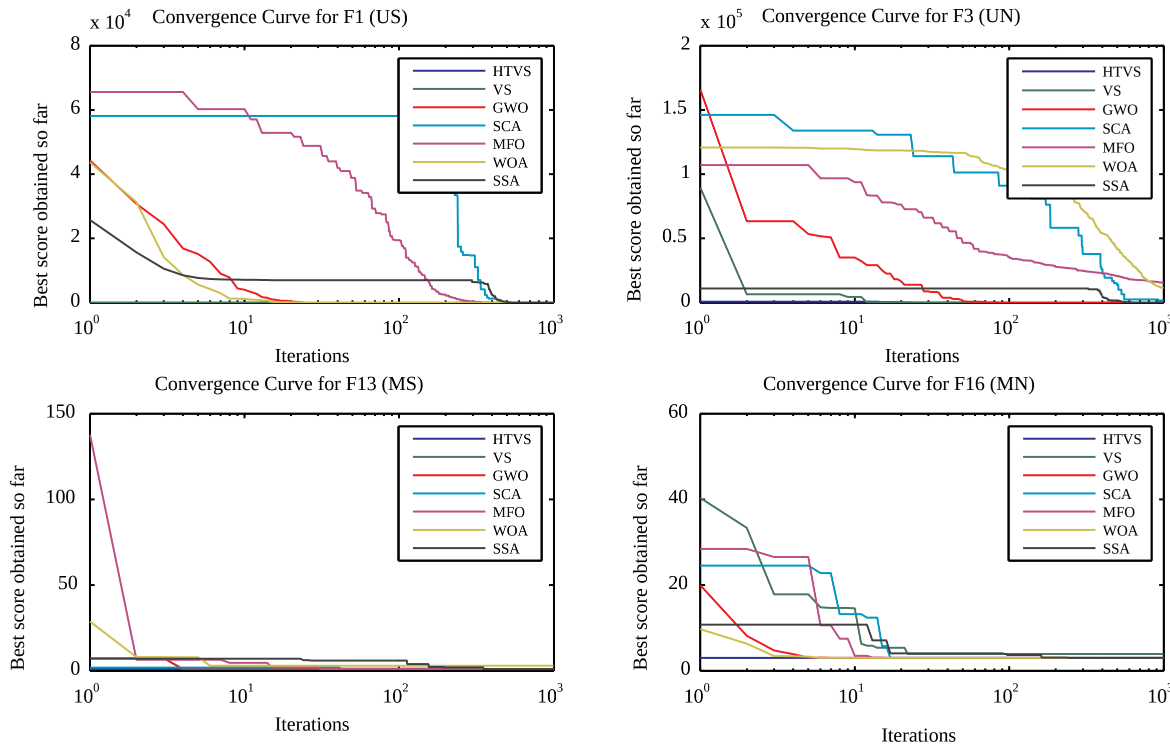

Figure 4. Convergence Curve

setting parameters of the algorithms have been found from the studies in the literature and have been expressed in the Appendix section.

3.2.1. Tension/Compression Spring Design

The constrained design problem shown in Figure 5 has been solved and the minimum weight of the tension / compression spring has been tried to be found [38],[42]. The optimum design should provide restrictions on shear stress, ripple frequency and deviation. Three design parameters are wire diameter (d), average coil diameter (D) and active coil number (N).

The equations of constrained design problem is defined as follows [38],[42]:

Consider $Y = [y_1 \; y_2 \; y_3] = [d \; D \; N]$

Minimize $f(Y) = (y_3 + 2)y_2 y_1^2$

Subject to $h_1(Y) = 1 - \frac{y_2^3 y_3}{71785 y_1^4} \leq 0$

$$h_2(Y) = \frac{4y_2^2 - y_1 y_2}{12566(y_2 y_1^3 - y_1^4)} + \frac{1}{5108 y_1^2} - 1 \leq 0$$

$$h_3(Y) = 1 - \frac{140.45 y_1}{y_2^2 y_3} \leq 0$$

$$h_4(Y) = \frac{y_1 + y_2}{1.5} - 1 \leq 0$$

where $0.05 \leq y_1 \leq 2.00, \; 0.25 \leq y_2 \leq 1.30, \; 2.00 \leq y_3 \leq 15.00$

The results obtained with HTVS algorithm are compared with various techniques applied to this design problem in the literature. Founded values and comparative cost are shown in the Table 6 and it shows that

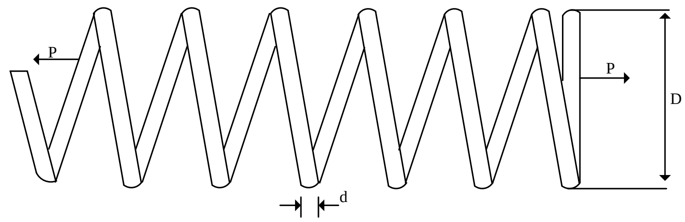

Figure 5. Schematic of T/CSD Problem [38],[42]

the optimum cost parameter obtained with the proposed HTVS algorithm is the same as the optimum cost value obtained with the HEAA and WCA algorithm in the literature. The optimum cost value found with the proposed HTVS, WCA and the HEAA algorithm is better than the optimum cost values found by other algorithms in Table 6. It is also worth noticing here that although optimum costs value found by proposed HTVS, HEAA and WCA are equal, the obtained optimal design parameters are different. So, HTVS finds a new optimal design for this problem. Also, the proposed HTVS algorithm shows that it can compete with other algorithms in the literature with the optimal cost result for the T/CSD design problem.

Table 6. The experimental results for T/CSD problem.

Algorithms	Optimum Parameters			Optimum Cost
	d	D	N	
HTVS	0.05176	0.35845	11.18786	0.012665
WOA [6]	0.05127	0.34521	12.00402	0.01267
HEAA [39]	0.05168	0.35672	11.28829	0.012665
CPSO [40]	0.05172	0.35764	11.24454	0.012674
WCA [41]	0.05168	0.35637	11.30922	0.012665
GA [42]	0.05148	0.35166	11.63220	0.012704
AIS-GA [43]	0.051660	0.35603	11.32955	0.012666
CDE [44]	0.05160	0.35471	11.41083	0.012670

3.2.2. Pressure Vessel Design

The main purpose of this section is to optimize the overall cost function of PVD problem under the different constraints. PVD schematic is illustrated in Figure 6 [38],[42]. Whereas the head is semi-spherical in shape, both ends of the container are covered. It has four design parameter: the thickness (T_s), the thickness of the head (T_h), the inner radius (R), the length, regardless of the head (L).

The equations and constraints of this problem can be written as follows [38],[42]:

Consider $Y = [y_1, y_2, y_3, y_4] = [T_s, T_h, R, L]$

Minimize $f(Y) = 0.6224y_1y_3y_4 + 1.7781y_2y_3^2 + 3.1661y_1^2y_4 + 19.84y_1^2y_3$

Subject to $h_1(Y) = -y_1 + 0.0193y_3 \leq 0$

$h_2(Y) = -y_2 + 0.00954y_3 \leq 0$

$h_3(Y) = -\pi y_3^2 y_4 - \frac{4}{3}\pi y_3^3 + 1296000 \leq 0$

$h_4(Y) = y_4 - 240.0 \leq 0$

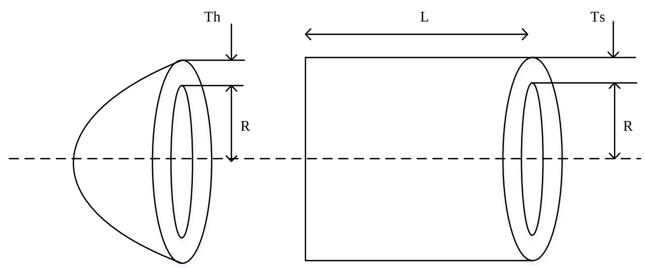

Figure 6. Schematic of PVD Problem [38],[42]

Where $0 \leq y_1 \leq 99,\ 0 \leq y_2 \leq 99, 10 \leq y_3 \leq 200, 10 \leq y_4 \leq 200 - 240^*$.

There are some studies in the literature with a maximum y_4 value of 200 [6], [40],[42] and 240 [47] . The results obtained in both cases are given in Table 7. Also, results where y_4 is a maximum of 240 are marked with $*$. This problem is frequently used by researchers in optimization applications. According to this table, HTVS algorithm has found better optimum cost than other algorithms. Setting parameters of compared algorithms are given in Table 8.

Table 7. The experimental results for PVD problem.

Algorithms	Optimum Parameters				Optimum Cost
	T_s	T_h	R	L	
HTVS	0.7828	0.3869	40.5575	196.7148	5893.2314
WOA [6]	0.81250	0.43750	42.09826	176.63899	6059.7410
CPSO [40]	0.8125	0.4375	42.091266	176.7465	6061.0777
GA [42]	0.8125	0.4345	40.3239	200.00	6288.7445
CDE [44]	0.81250	0.43750	42.09841	176.63769	6059.7340
DELC [45]	0.8125	0.4375	42.09844	176.63659	6059.7143
G-QPSO [46]	0.8125	0.4375	42.0984	176.6372	6059.7208
HTVS*	0.7455	0.3685	38.62635	224.9935	5831.7849
BGRA* [47]	0.75	0.375	38.8601	221.36547	5850.383061
IHSA*[48]	0.75	0.375	38.86010	221.36553	5850.38363
DSO*[49]	0.75	0.375	38.86010	221.36547	5850.38309

4. Conclusion

In this article, Vortex Search, a single-solution based metaheuristic algorithm, is explored and adjusted by means of the orthogonal array concept. Taguchi orthogonal approximation and VS algorithm are hybridized in proposed Hybrid Taguchi-Vortex Search algorithm. Thus, more powerful and more reliable HTVS is developped.

This paper presented two experiment to examine the success of the HTVS algorithm in solving optimization problems. In the first experiment, the proposed method has been applied on 16 benchmark functions and performance comparison with TOAA, VS, GWO, SCA, MFO, WOA and SSA algorithms. The success of HTVS in solving numerical optimization problems has been expressed using the Wilcoxon Signed-Rank Test. In the second experiment, two real engineering problems with constraints (i.e. design of a tension/compression spring and design of a pressure vessel) have been solved to learn more about the proposed algorithm. When analyzed all obtained results, HTVS is extremely competitive with the other optimization algorithms used in this study.

Appendix

Table 8. Setting parameters of compared algorithms.

Algorithms	Parameters	Values
WOA for T/CSD[6]	Search agents	10
	Iteration number	500
WOA for PVD[6]	Search agents	20
	Iteration number	500
HEAA[39]	N	60
	Q_1, Q_2	200, 60
	Simplex crossover parameters	10, 5, 10
	Fitness function evaluations	200,000
CPSO[40]	Size of swarms M_1, M_2	50, 20
	Number of generations G_1, G_2	25, 8
	Acceleration coefficients c_1, c_2	2, 2
	Maximum particles position $w_{1,max}$, $w_{2,max}$	1000, 1000
	Minimum particles position$w_{1,min}$, $w_{2,min}$	0, 0
WCA[41]	N_{total}	50
	N_{sr}	8
	d_{max}	1-03
GA[42]	$populationsize_1$	60
	$populationsize_2$	30
	$G_{max}1$	25
	$G_{max}2$	20
AIS-GA[43]	Population size	20
	Binary gray code	50 bits
	Crossover probability	1
	Mutation ratio	0.02
	Elitism	2
	Maximum iteration	20
	Cumber of clones	3
	Critical distance	10%
CDE[44]	M_1, M_2	32, 8
	G_1, G_2	10, 10
	F_1, F_2	0.6, 0.8
	CR_1 , CR_2	0.2, 0.1
DELC[45]	Decision variable n	4
	Population size N	80
	Level parameter	0.1
	Total number of function evaluation TNFE	30,000
G-QPSO[46]	Population size	20
	Iteration number	400
BGRA[47]	Population size	200
	Iteration number	2000
	level with *sr*	0.6
	count with *r*	0.1
IHSA[48]	Harmony memory considering rate	0.95
	Pitch adjusting rate PAR_{max}, PAR_{min}	0.99, 0.45
	Harmony memory size	6
	Arbitrary distance bandwidth bw_{min}, bw_{max}	5e-4, 0.05
	NI (stopping criterion)	200,000
DSO[49]	Population size	40
	Forward probability	0.8
	Forward coefficient	1
	Backward coefficient	10
	Genetic mutation probability	0.01
	Iteration number	1000

References

[1] Shabani A, Asgarian B, Gharebaghi SA, Salido MA, Giret A. A new optimization algorithm based on search and rescue operations. Mathematical Problems in Engineering 2019; 2019: 1-23. doi: 10.1155/2019/2482543

[2] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in Engineering Software 2014; 69: 46-61. doi: 10.1016/j.advengsoft.2013.12.007

[3] Kennedy J, Eberhart R. Particle swarm optimization. In:Proceedings of IEEE International Conference on Neural Network; Perth, Western Australia; 1995. pp. 1942–1948.

[4] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H et al. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software 2017; 114: 163-191. doi: 10.1016/j.advengsoft.2017.07.002

[5] Gandomi AH, Alavi AH. Krill herd: A new bio-inspired optimization algorithm. Communications in Nonlinear Science and Numerical Simulation 2012; 17 (12): 4831-4845. doi: 10.1016/j.cnsns.2012.05.010

[6] Mirjalili S, Lewis A. The whale optimization algorithm. Advances in Engineering Software 2016; 95; 51-67. doi: 10.1016/j.advengsoft.2016.01.008

[7] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 2007; 39: 459-471. doi: 10.1007/s10898-007-9149-x

[8] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A gravitational search algorithm. Information Sciences 2009; 179 (13): 2232–2248. doi: 10.1016/j.ins.2009.03.004

[9] Erol OK, Eksin I. A new optimization method: Big bang-big crunch. Advances in Engineering Software 2006; 37 (2): 106–111. doi: 10.1016/j.advengsoft.2005.04.005

[10] Zheng YJ. Water wave optimization: A new nature-inspired metaheuristic. Computers & Operations Research 2015; 55; 1–11. doi: 10.1016/j.cor.2014.10.008

[11] Hatamlou A. Black hole: A new heuristic optimization approach for data clustering. Information Sciences 2013; 222: 175-184. doi: 10.1016/j.ins.2012.08.023

[12] Holland JH. Genetic algorithms. Scientific American 1992; 267(1): 66–72.

[13] Storn R, Price K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 1997; 11: 341–359.

[14] Simon D. Biogeography-based optimization. IEEE Transactions on Evolutionary Computation 2008; 12 (6): 702–713. doi: 10.1109/TEVC.2008.919004

[15] Bellman R. Dynamic programming. Princeton University Press, 1957, Princeton, NJ.

[16] Gupta S, Dalal U, Mishra VN.Performance on ICI self-sancellation in FFT-OFDM and DCT-OFDM system. Journal of Function Spaces 2015; 2015: 1–7. doi:10.1155/2015/854753.

[17] Dubey R, Deepmala, Mishra VN.Higher-order symmetric duality in nondifferentiable multiobjective fractional programming problem over cone contraints. Statistics, Optimization & Information Computing 2020; 8: 187–205.

[18] Gupta S, Dalal U, Mishra VN. Novel analytical approach of non conventional mapping scheme with discrete hartley transform in OFDM System. American Journal of Operations Research 2014; 4: 281-292.

[19] Vandana, Dubey R, Deepmala, Mishra LN, Mishra VN. Duality relations for a class of a multiobjective fractional programming problem involving support functions. American Journal of Operations Research, 2018; 8: 294-311.

[20] Villarrubia G, Paz JFD, Chamoso P, Prieta FDL. Artificial neural networks used in optimization problems. Neurocomputing, 2018; 272: 10-16. doi:10.1016/j.neucom.2017.04.075

[21] Outa R, Chavarette FR, Mishra VN, Gonçalves AC, Roefero LGP, Moro TC. Prognosis and fail detection in a dynamic rotor using artificial immunological system. Engineering Computations, 2020; 1-19. doi:10.1108/EC-08-2019-0351.

[22] Hu Z, Bao Y, Xiong T. Partial opposition-based adaptive differential evolution algorithms: Evaluation on the CEC 2014 benchmark set for real-parameter optimization. In: 2014 IEEE Congress on Evolutionary Computation; Beijing, China; 2014. pp. 2259-2265.

[23] Kaur K, Kumar S, Saxena J. HGAB3C: A new hybrid global optimization algorithm. Turkish Journal of Electrical Engineering & Computer Sciences 2019; 27: 3557-3566. doi: 10.3906/elk-1810-74

[24] Rizk-Allah RM. Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems. Journal of Computational Design and Engineering 2018; 5 (2): 249-273. doi: 10.1016/j.jcde.2017.08.002

[25] Singh SB, Singh N. Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. Journal of Applied Mathematics 2017; 2017: 1-15. doi: 10.1155/2017/2030489

[26] Erlich I, Rueda JL, Wildenhues S, Shewarega F. Evaluating the mean-variance mapping optimization on the IEEE CEC 2014 test suite. In: 2014 IEEE Congress on Evolutionary Computation; Beijing, China; 2014. pp. 1625-1632.

[27] Yavuz G, Aydın D, Stützle T. Self-adaptive search equation-based artificial bee colony algorithm on the CEC 2014 benchmark functions. In: 2016 IEEE Congress on Evolutionary Computation; Vancouver, BC, Canada; 2016. pp. 1173-1180.

[28] Yıldız AR. Hybrid taguchi-harmony search algorithm for solving engineering optimization problems. International Journal of Industrial Engineering 2008; 15(3): 286-293.

[29] Wang Z, Wu G, Wan Z. A novel hybrid vortex search and artificial bee colony algorithm for numerical optimization problems. Wuhan University Journal of Natural Sciences 2017; 22(4): 295-306. doi:10.1007/s11859-017-1250-5.

[30] Doğan B. A Modified Vortex Search Algorithm for Numerical Function Optimization. International Journal of Artificial Intelligence & Applications 2016; 7(3): 37-54. doi:10.5121/ijaia.2016.7304

[31] Weng WC, Yang F, Elsherbeni AZ. Linear antenna array synthesis using taguchi's method: A novel optimization technique in electromagnetics. IEEE Transactions on Antennas and Propagation 2007; 55 (3): 723-730. doi: 10.1109/TAP.2007.891548

[32] Doğan B, Ölmez T. A new metaheuristic for numerical function optimization: Vortex search algorithm. Information Sciences 2015; 293: 125-145. doi: 10.1016/j.ins.2014.08.053

[33] VSA (2020). Vortex Search (VS) Algorithm for Numerical Optimization Problems : Matlab Code [online]. Website https://web.itu.edu.tr/ bdogan/VortexSearch/VS.htm [accessed 14 10 2020].

[34] Altınöz Ö, Yılmaz AE, Weber GW. Orthogonal array based performance improvement in the gravitational search algorithm. Turkish Journal of Electrical Engineering & Computer Sciences 2013; 21: 174-185. doi: 10.3906/elk-1105-27

[35] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. Knowledge-Based Systems 2016; 96: 120–133. doi: 10.1016/j.knosys.2015.12.022

[36] Mirjalili S. Moth-flame optimization algorithm: A novel nature inspired heuristic paradigm. Knowledge-Based Systems 2015; 89: 228–249. doi: 10.1016/j.knosys.2015.07.006

[37] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 2009; 214 (1): 108–132. doi: 10.1016/j.amc.2009.03.090

[38] Dong M, Wang N, Cheng X, Jiang C. Composite differential evolution with modified oracle penalty method for constrained optimization problems. Mathematical Problems in Engineering 2014; 2014: 1-15. doi: 10.1155/2014/617905

[39] Wang Y, Cai Z, Zhou Y, Fan Z.Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint handling technique. Structural and Multidisciplinary Optimization 2009;37: 395–413. doi: 10.1007/s00158-008-0238-3

[40] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence 2007;20: 89–99. doi: 10.1016/j.engappai.2006.03.003

[41] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm–A novel meta heuristic optimization method for solving constrained engineering optimization problems. Computers & Structures 2012;110-111: 151–66. doi: 10.1016/j.compstruc.2012.07.010

[42] Coello CCA . Use of a self-adaptive penalty approach for engineering optimization problems. Computers in Industry 2000;41: 113–27. doi: 10.1016/S0166-3615(99)00046-9

[43] Bernardino H, Barbosa I, Lemonge A. A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In:Proceedings of IEEE Congress on Evolutionary Computation (CEC-2007); Singapore, Singapore; 2007. pp.646–53. doi: 10.1109/CEC.2007.4424532

[44] Huang F, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. Applied Mathematics and Computation 2007;186(1): 340–56. doi: 10.1016/j.amc.2006.07.105

[45] Wang L, Li LP. An effective differential evolution with level comparison for constrained engineering design. Structural and Multidisciplinary Optimization 2010; 41: 947–63. doi: 10.1007/s00158-009-0454-5

[46] CoelhoL DS. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Systems with Applications 2010; 37: 1676–83. doi: 10.1016/j.eswa.2009.06.044

[47] Hsieh TJ. A bacterial gene recombination algorithm for solving constrained optimization problems. Applied Mathematics and Computation 2014; 231: 187–204. doi:10.1016/j.amc.2013.12.178

[48] Mahdavi M, Fesanghary M, Damangir E.An improved harmony search algorithmfor solving optimization problems. Applied Mathematics and Computation 2007; 188: 1567–1579. doi:10.1016/j.amc.2013.12.178

[49] Zou D, Liu H, Gao L, Li S. Directed searching optimization algorithm for constrained optimization problems. Expert Systems with Applications 2011; 38: 8716–8723. doi:10.1016/j.amc.2013.12.178