A Hybrid Algorithm based on Invasive Weed Optimization and Particle Swarm Optimization for Global Optimization

Zeynab Hosseini
Department of Computer Engineering
Urmia Branch, Islamic Azad University
Urmia, Iran

Ahmad Jafarian*
Department of Mathematics
Urmia Branch, Islamic Azad University
Urmia, Iran

***Abstract*—In this paper, an effective combination of two Metaheuristic algorithms, namely Invasive Weed Optimization and the Particle Swarm Optimization, has been proposed. This hybridization called as HIWOPSO, consists of two main phases of Invasive Weed Optimization (IWO) and Particle Swarm Optimization (PSO). Invasive weed optimization is the nature-inspired algorithm which is inspired by colonial behavior of weeds. Particle Swarm Optimization is a swarm base Algorithm that uses the swarm intelligence to guide the solution to the goal. IWO algorithm is the algorithm which is not benefit from swarm intelligence and PSO converges to the local optimums quickly. In order to benefit from swarm intelligence and avoidance from trapping in local solutions, new hybrid algorithm IWO and PSO has been proposed. To obtain the required results, the experiment on a set of benchmark functions was performed and compared with other algorithms. The findings based on the non-parametric tests and statistical analysis showed that HIWOPSO is a more preferable and effective method in solving the high-dimensional functions.**

Keywords—Invasive weed optimization; Particle Swarm Optimization; Global optimization; Hybrid algorithm

I. Introduction

One of the affective methods in finding the best solution in numerical problems is the Optimization technique. In optimization, only a few solutions are considered the best which are called as the goal. Classical optimization techniques have some deficiencies on solving the complex optimization problems. These deficiencies are primarily interdependent on their inherent search systems. These classical optimization methods are strongly under effects of choosing proper objectives, constraints functions and type of variables. They also do not grant a universal result approach that can be used to solve problems where various type of variables, objective and constraint functions, are used [1]. For covering these deficiencies ,a new method with the name of Metaheuristic was designed, which is mainly originated from artificial intelligence research that developed by researchers [2]. A Metaheuristic is an algorithm designed for solving the various types of hard optimization problems without having to fully accommodate to each problem. The Greek word meta indicates that these methods are higher-level heuristics. The primary features of Metaheurisitc methods are as follows: they are nature-inspired (meaning that they have originated from nature physics, behavior and etc); stochastic components are one of the inseparable parts of these methods (involving random variables); they aren't gradient base method and don't use them; at the beginning of program, they have several parameters which needs to adjusts properly. Metaheuristic algorithms combine various intelligent procedures and guide basic heuristic methods [3]. These algorithms are inspired from different things such as natural phenomena, natural selections and social behaviors and applied in solving the optimization problems. Examples of the recently metaheurtistc algorithms are Vortex search [4], WOA (whale optimization algorithm) [5], MBA (mine blast algorithm) [6], WCA(water cycle algorithm) [7], and SFS (stochastic fractal search) [8].

The PSO [9] is the other Metaheuristic algorithm which has been utilized in the optimization of many problems. This algorithm uses the strategy of birds and folks in migration for finding better solutions. Individuals in the PSO are called as particles and each particle has velocity in the searching space. Particles are distributed randomly in the searching space and positions of the particles are changed based on the velocity which has been calculated. These particles tend to move toward the best positions which causes to seek a better position and find the best. One of the deficiencies that can be specified for PSO is that it often falls to the local minimum quickly, missing better opportunities when facing multimodal functions [10].

IWO is a nature inspired algorithm which is getting much more attention because it shows efficient exploration and dissimilarity properties [11] and took an exceptional place for solving continuous optimization problems. In formal IWO, the seeds are uniformly spread on the search space [11]. After all, if the searching space is too large, its not efficient to use this type of distribution for the searching. Also, initialization of the parameters is so important task in IWO and trapping in local solution is a probable event. Hence ,it was found that the efficiency of an IWO algorithm to achieve success goal in problems relies too much on its initial parameters and these parameters should be wisely selected based on the problem to be solved. To overcome these problems, a novel hybrid IWO and PSO algorithm is proposed and implemented for solving continuous optimization problems. Hybridization of IWO with other algorithms has been investigated in many studies. MICA-IWO [12]is a new type of hybrid method which combines two well-known Metaheuristic approaches: IWO and imperialist competitive algorithm (ICA) [13]. ICA has outperformed many of the already existing stochastic and direct search global optimization techniques. The hybrid MICA-IWO method has been used for handling optimal reactive power dispatch (ORPD) problem. Other recently hybrid method based on IWO is the hybrid IWO/WDO [14] algorithm. In IWO/WDO, wind driven optimization (WDO) [15] algorithm has been combined with IWO algorithm for nulling pattern synthesis of uniformly spaced linear and non-uniform circular array antenna.

In this paper, we will combine PSO and IWO global optimization algorithms, and propose the novel hybrid algorithm based on these algorithms which are jointly called as HIWOPSO. As PSO has Swarm intelligence, this could provide more variant population for IWO which could help in finding better places by using the previous experiences. To avoidance of trapping in local solutions and searching the large area greatly, the mutation function has been used which applied for checking other places and give chance for them. Proposed method is tested on the benchmark functions and compared with other famous algorithms. The statistical analysis Friedman test [16] is performed on the results and compared with other algorithms results. The Convergence diagram and normality diagrams for the statistical analysis is also presented.

In the real world, many problems have been proposed and optimization problems are one of them [17]. The optimization problems are single or multi-objective. The multi-objective is the problem with more than one objective function (m>c1) and single objective is a problem with one objective function (m=1). The main goal in this procedure is to seek the global minimum or maximum. The function may have more than one minimum or maximum which is called as the local, but only one of them is the global maximum or minimum. The point x* is the global minimum if f(x*)_f(x) for all the x in the searching space S. Optimization problem may consist of one or more mathematical functions which need to be optimized. The general form of the optimization problem is indicated in Eq. (1).

$$\text{MinimizeF } (f1(x);\ldots;\ fm(x)),\ \ x = (x1,\ldots,xn) € S. \qquad (1)$$

Where n is the decision variables, m is the number of objectives, x is decision vector and S is searching space. If the problem has one objective function (m=1), then it should be indicated as Eq. (2).

$$\text{minimizef}(x), \qquad x = (x_{,\ldots,}x_n)\ € \ S. \qquad (2)$$

The rest of the paper is organized as follows: Section 2 illustrates the PSO and IWO algorithms, and section 3 discusses the HIWOPSO algorithm, its parameters and boundary control. Section 4 presents 26 benchmark test functions applied for the experiments. Finally, the last section presents the concluding remarks.

II. Related Works

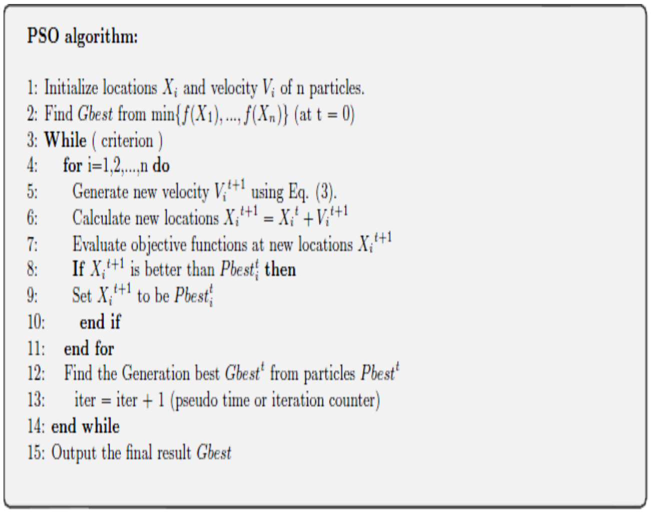
The PSO is the one of the Metaheuristic algorithms which is originated from the nature. This algorithm was introduced by Kenndy and Eberahart in 1995 [9]. The PSO is originated from the birds and folks migration behavior, living in small and large numbers of groups. The birds use a method for finding food and migration, which has been used in this algorithm. In this method, only the birds know their distance from food, but they don't know the location of the food thus, following the other neighboring birds is the best way for surviving.

The PSO consist of elements with the name of particles which is a probable solution in the searching space. The main steps in the PSO algorithm are as follow: first, particles are distributed randomly in the searching area and PSO starts the process with these particles. In this searching process, particles only follow the one which is nearer to the goal and has better fitness value. Each particle has a velocity which is represented by V_i and calculated by Eq. (3) in the D-dimensional searching space. Particles are under the effect of personal ($pbest_i^t$) and swarm experiences ($Gbest_i^t$) and the position is updated by Eq. (4).

$$v_i\ [t+1] = wv_i\ [t]\ \ +c_1r_1(x_{i,best}[t] - x_i[t])\ \ +c_2r_2(x_{g,best}[t] - x_i[t])\ (3)$$

$$x_i[t+1] = x_i[t] + v_i[t+1] \qquad (4)$$

In Eq. (3) and (4), Xi represents the ith particle of the population, c1 and c2 are the learning coefficients, r1 and r2 are random values between [0 1], ! is the inertia weight, and Vi is the ith member of particles velocity. $pbest_i^t$ and $Gbest_i^t$ are the personal best and generation best.



PSO algorithm:

```
Initialize locations $X_i$ and velocity $V_i$ of n particles.
Find $Gbest$ from min$\{f(X_1),...,f(X_n)\}$ (at t = 0)
While ( criterion )
  for i=1,2,...,n do
    Generate new velocity $V_i^{t+1}$ using Eq. (3).
    Calculate new locations $X_i^{t+1} = X_i^t + V_i^{t+1}$
    Evaluate objective functions at new locations $X_i^{t+1}$
    If $X_i^{t+1}$ is better than $Pbest_i^t$ then
    Set $X_i^{t+1}$ to be $Pbest_i^t$
      end if
  end for
  Find the Generation best $Gbest^t$ from particles $Pbest^t$
   iter = iter + 1 (pseudo time or iteration counter)
end while
Output the final result $Gbest$
```

Invasive weed optimization was first introduced by Mehrabian and Locus [11] and it is One of the population based optimization algorithms which is originated from colonial behavior of weeds. The IWO algorithm is a very simple and yet efficient algorithm in finding optimum solution of the objective function, which is implemented based on the natural and basic features of weeds in a colony such as reproduction, growth and competition to survive. In comparison to other algorithms, IWO is simpler and has adequate ability and convergence rate to the global optimum solution of the objective function. Some of the major features of this algorithm which specifies it from other methods are reproduction, space distribution and exclusive competition.

In order to simulate behavior of a weed, we have the following algorithm, according to [11].

Step 1: Initial population production: a population of N0 seeds is randomly distributed in an n dimensional space.

Step 2: Reproduction: each seed grows and turns into a mature plant and then, begins seed production for newer generation. The amount of seeds produced by a plant increases linearly between two possible values of minimum (S_{min}) and maximum (S_{max}) possible amounts of produced seeds. The amount of produced seeds for the ith plant in every repeat is dependent to its goal value (F_i), its best (F_{best}) and worst (F_{worst}) goal values in that repeat and is calculated with the following equation:

$$Numseed(i) = \left[S_{min} + (S_{max} - S_{min})\frac{f - f_{worst}}{f_{best} - f_{worst}}\right] \quad (5)$$

Step 3: Distribution space the randomness and assimilation of the algorithm are related to this stage. The produced seeds are distributed in the d dimensional search space with normal distribution which has zero mean and different variance of (N(0;$_{\_t}$)$_d$).In this state, the seeds will be near the breeder plant. Although, the standard deviation decreases from initial amount (_$_{initial}$) to final amount (_$_{final}$) in each repeat, in the simulations, non-linear variation of standard deviation causes satisfactory results which are illustrated below:

$$\sigma_t = \left(\frac{T-t}{T}\right)^n \times \left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \quad (6)$$

In Eqs. (13)(15), T represents the maximum number of repeats related to (t) and is the non-linear modulation factor. In this status, the positions of seeds (Sj) for the ith plant (wi) are calculated as follows:

$$S_j = W_i + \mathrm{N}(0, \delta_t)^d,\ 1 <= \mathrm{j} \leq \mathrm{numseed(i)} \quad (7)$$

Step 4: Exclusive competition: by several repeats, the number of plants produced by rapid reproduction reaches its maximum value (W_{max}), in this situation, every plant is permitted to produce seeds by in accordance with reproduction method. The seeds are authorized to spread in search space with correspondence to distribution space method, when the seeds find their position; they form a colony alongside their parent plants. Then members with less propriety are deleted in order to number of members reach its maximum allowed value. In this method, the parent plants combine with their children and the plants with most propriety from the group are preserved and allowed for replacement.

This crowd control mechanism will be imposed on next generations until reaching the final period. Step 5: If the criterion satisfied end otherwise, return to Step 2.

III. Proposed Model

The swarm intelligence is some kind of artificial intelligence which has been established based on group behaviors in decentralized and self-organizing systems. This systems usually included population of the simple agents that interact locally with each other and their environment. Some samples of this system that can be mentioned are ants groups, birds flock, fishes flock, bacterize bulk and animals herd. In order to use of swarm intelligence, it needs to use the behavior of these systems in the proposed method. So PSO algorithm has been using with IWO algorithm to give the behavior of swarm intelligence to agents of IWO algorithm and use it in guiding the solution of the problems to the goal. As previously mentioned, in IWO algorithm, each father (weed) produce some child (seed) that these Childs distribute around the father with a kind of normal distribution. In proposed hybrid method, this distribution could be based on the some kind of normal distribution or guided through the goal by using the swarm intelligence and previously experience of father. Using the previously experiences is like this, if each weed (wi) could give good result, it stores the result as the experience in the experience memory and these experiences inherited to this weed Childs and used in guiding the solutions of the problem like the PSO algorithm. Each child in proposed method (HIWOPSO) inherited two things from their father and that is the velocity (V_i) and experience (w_P). The main process of proposed method is like this: first, the population with size npop0 of weeds (w_i) are distributed randomly around the searching space and their position is initialized randomly. The cost of each weed is computed from the position which calculated previously and at the Start, the value of velocity is zero (V_i = 0). first, The experience of each weed is the value

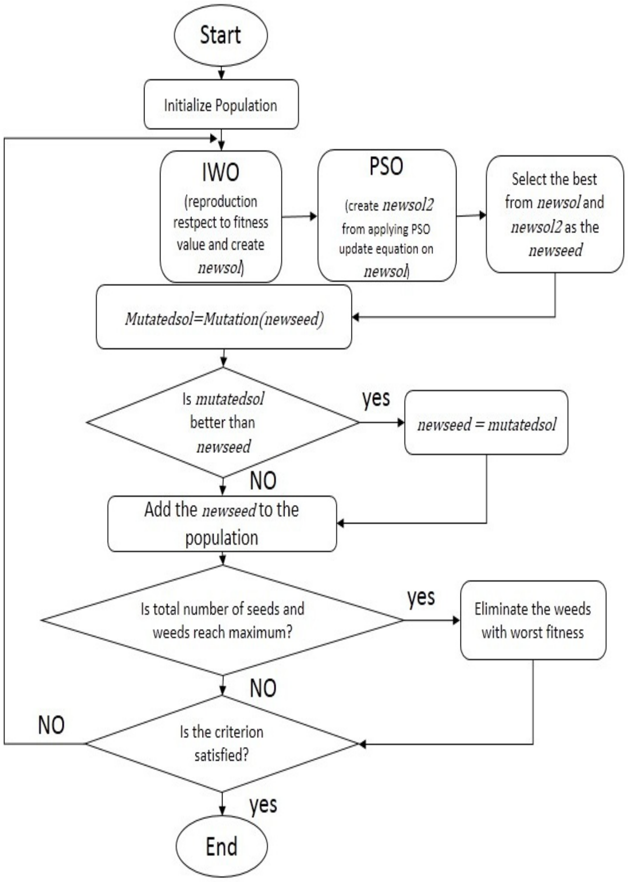

Start
Initialize Population
IWO
(reproduction restpect to fitness value and create newsol)
PSO
(create newsol2 from applying PSO update equation on newsol)
Select the best from newsol and newsol2 as the newseed
Mutatedsol=Mutation(newseed)
Is mutatedsol better than newseed
yes
newseed = mutatedsol
NO
Add the newseed to the population
Is total number of seeds and weeds reach maximum?
yes
Eliminate the weeds with worst fitness
NO
NO
Is the criterion satisfied?
yes
End

Fig. 1. Flowchart of HIWOPSO algorithm

which previously initialized to the position and cost for each weed. After producing population with size of npop0, algorithm enters to the main iteration loop which at start have a section for computing the number of producing child for each weed by Eq. (5) and also the velocity for each weed is computed like the following equation:

$$w_i^{t+1} = w.w_i^t + c_1.r_1.(w_{pi}^t - w_i^t) + c_2 + r_2.(w_{best\ i}^t - w_i^t) \qquad (8)$$

Where w_i is ith weed in population and w_p is personal experience and w_{best} is the global best , other variables are described in PSO algorithm and This equation is same as the PSO velocity update equation. For each of the w_i enters to a loop and one solution is created by the seed distribution formula in IWO (newsol) Eq. (9) and another solution is created by PSO (newsol2) Eq (10).

$$\text{Newsol2} = w_i + \mathrm{N}(0, \delta_t)^d \qquad (9)$$

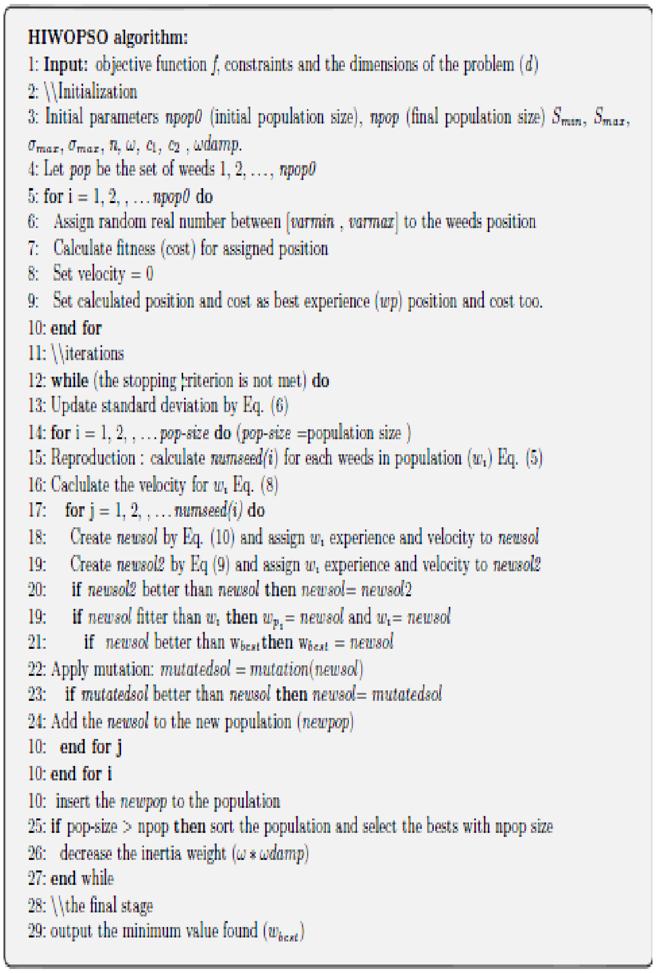
$$\text{Newsol} = w_i + V_i \qquad (10)$$

New solutions newsol and newsol2 are compared with each other and if newsol2 better than newsol then newsol value will be replaced by the newsol2 value. The newsol is compared with w_i experience and if newsol better than w_p it will set as w_i experience and it also if it is better than global best, the global best will be updated with newsol. After using the PSO and IWO for creating new solutions, its time to reduce the probability of trapping in local solutions.

For decreasing this probable event , using mutation function could be help full after producing new solution. The mutation function applied to the new solution (newsol) and the output (mutatedsol) will be compared with newsol. The mutatedsol is compared with newsol and if the mutation creates better solution then the value of newsol will be replaced by the mutatedsol value and newsol inserted to the population. The mutation function described briefly in next subsection. After inserting the newsol to the population, it must be control the size of population not to exceed from maximum size which was initialized at the start (npop).

The mutation function which has been using in proposed algorithm, uses random selection of dimension and replaces that dimension values with the random values that has been obtained from function domain. For example if the input for the mutation function is the following input=[0.7 0.5 0.6 0.8] with boundary [L U]=[0.4 0.9] and if the mutation function selects random dimensions 2 and 4 then that dimensions values will be changed with random real value from domain [0.4 0.9] and output=[0.7 0.45 0.6 0.65]. Random selection of dimension could be any order of input dimension size for example, if d=30 then random dimension could be f{1 5 18 20} , f1{ 2 3 4 5 6 7 19} and etc... The number of dimension that selected for replacement is determine randomly in each execution of mutation function.

The pseudo code for proposed method described as follows:



```
HIWOPSO algorithm:
Input: objective function $f$, constraints and the dimensions of the problem ($d$)
\\Initialization
Initial parameters npop0 (initial population size), npop (final population size) $S_{min}$, $S_{max}$,
$\sigma_{max}$, $\sigma_{max}$, n, $\omega$, $c_1$, $c_2$ , $\omega damp$.
Let pop be the set of weeds 1, 2, …, npop0
for i = 1, 2, , … npop0 do
  Assign random real number between [varmin , varmax] to the weeds position
  Calculate fitness (cost) for assigned position
  Set velocity = 0
  Set calculated position and cost as best experience (wp) position and cost too.
end for
\\iterations
while (the stopping criterion is not met) do
Update standard deviation by Eq. (6)
for i = 1, 2, , … pop-size do (pop-size =population size )
Reproduction : calculate numseed(i) for each weeds in population ($w_i$) Eq. (5)
Caclulate the velocity for $w_i$ Eq. (8)
  for j = 1, 2, , … numseed(i) do
    Create newsol by Eq. (10) and assign $w_i$ experience and velocity to newsol
    Create newsol2 by Eq (9) and assign $w_i$ experience and velocity to newsol2
    if newsol2 better than newsol then newsol= newsol2
    if newsol fitter than $w_i$ then $w_{p_i}$= newsol and $w_i$= newsol
      if newsol better than $w_{best}$ then $w_{best}$ = newsol
Apply mutation: mutatedsol = mutation(newsol)
  if mutatedsol better than newsol then newsol= mutatedsol
Add the newsol to the new population (newpop)
  end for j
end for i
 insert the newpop to the population
if pop-size > npop then sort the population and select the bests with npop size
  decrease the inertia weight ($\omega * \omega damp$)
end while
\\the final stage
output the minimum value found ($w_{best}$)
```

HIWOPSO algorithm is consists of 2 main parts and these parts are executed in consecutive form in each cycle which are showed at the flow chart Fig. 1. In each part process is performed on weeds of population with size of pop-size and weed position is d dimensional vectors. The main operator for each part effects on time complexity. The sections that alter time complexity are: PSO update equation, IWO position update equation. We can explain complexity analysis of HIWOPSO in worse case and prove the fastest execution of these two algorithms combination like this: As the HIWOPSO algorithms are executed in consecutive form, then the time complexity for IWO is O(nd) and for PSO is also O(nd), therefore the HIWOPSO algorithm is run in O(nd) complexity because of the consecutive form of execution in IWO and PSO algorithms. For analyzing convergence of the Metaheuristic algorithm, Markov chain Monte Carlo method is the one of the preferable methods for this task [18].

Most Metaheuristic algorithms can be sighted in the framework of Markov chain from statically viewpoint. Now if look at the proposed hybrid method closely using the framework of Markov chain Monte Carlo, each algorithm in HIWOPSO essentially forms a Markov chain and the appropriate better solutions which created in each iteration, replace with previous one. Convergence analyzing based on the Markov chain for algorithm HIWOPSO is performed as follows:

Definition 1. Assume that the best weed is shown by X^{*}:= $\{\ x^{*} \in X\colon f(x^{*}) = \min(f(x)\ \mid\ x \in X\ )\}$ where X is probable solution and f is fitness function. The number of best weeds in weeds population is shown by $\omega(N)$: $\mid$ N $\cap\ x^{*}\ \mid$.

Definition 2. Algorithm convergence with probability 1 to the best if this condition is true :

$\lim_{g\to\infty} P\{\ \omega\ \ ((N(g))$>= 1 $\mid$ N(0) N_0}=1 , where g indicates generation number and N0 is random initial population.

Theorem 1. HIWOPSO algorithm converges to its globally best solution with probability 1.

Proof: Let P_0(g)=P{ $\omega(N(g) = 0)\}$ then the probability due to the Bayesian condition final probability of P_0(g+1) is P_0(g+1)= P{ $\omega(N(g+1) = 0) \to$ P{ $\omega(N(g+1) = 0)$ } $\mid$ $\omega(N(g) \neq 0\}$ + P{ $\omega(N(g+1) = 0)\ \mid \omega(N(g) = 0)\}$ Since the best solution replace with previous one in memory, this expression P{ $\omega(N(g+1) = 0)\ \mid\ \omega(N(g) \neq 0$ is true

Hence, P_0(g+1) = P{ $\omega(N(g+1) = 0)\ \mid\ \omega(N(g) = 0) \times$ P{ $\omega(N(g) = 0)\}$.

P{ $\omega(N(g+1) = 1)\ \mid\ \omega(N(g) = 0)\}$ >0 is true because of the HIWOPSO algorithm by two main phases IWO and PSO store the best solution.

Make Γ=min P{ $\omega(N(g+1) = 1)\ \mid\ \omega(N(g) = 0)$ min , g = 0,1,2…
Then
P{ $\omega(N(g+1) = 0)\ \mid\ \omega(N(g) = 0)\}$

=1-P {w(N(g+1) $\neq 0\ \ \mid \omega(N(g) = 0)\}$
=1-P {w(N(g+1) $\geq 1\ \ \mid \omega(N(g) = 0)\}$
≥ 1-P {w(N(g+1) $= 1\ \ \mid \omega(N(g) = 0)\}$≤ 1-Γ <1

Therefore,
0<= P_0(g+1)= P{ $\omega(N(g+1) = 0)\}$≤(1-Γ)× P{ $\omega(N(g) =$ 0)}=(1-Γ)× P_0(g).
such that, 0≤ P_0(g+1)≤(1-Γ)× $P_0(0)$.
Hence, 0<= P_0(g+1) ≤ (1-Γ)×(1-Γ)× P_0(g-1)≤…≤$(1-\Gamma)^{g+1}$×P_0(0).

Given that $\lim_{g\to\infty}(1-\Gamma)^{g+1} = 0$ and 0 ≤ $P_0(0)$≤ 1.

Hence 0 ≤ $\lim\limits_{g\to\infty} P_0(g)$≤ $\lim_{g\to\infty}(1-\Gamma)^{g}$ ×P_0(0)=0 , $\lim\limits_{g\to\infty} P_0(0) = 0$

Then
$\lim\limits_{g\to\infty}$ P{w(N(g))≥ 1 $\mid$ w (N(0))=N_0}

1-$\lim\limits_{g\to\infty}$ P{w(N(g))= 0 $\mid$ w (N(0))=N_0}

1-$\lim\limits_{g\to\infty} P_0(0) = 1$.

There for, when g→ ∞ , P{w(N(g))≥ 1 → 1. HIWOPSO algorithm could reach to best solution and assurance convergence with probability 1.

Parameter adjustment is a non-negligible task which is required to be performed properly in order to get a better result in solving various problems. Besides, the parameter adjustment also is necessary for controlling the boundary whenever the algorithm finds a new solution [19]. The (HIWOPSO) needs boundary control for a weed wi, because its position is required to be in the searching space, which is a boundary between [L U], (L is the lower bound and U is the upper bound of the searching space). The method that controls the boundary is as follows:

p = Max(X,L); q = Min(p, U) (11)

Where Min and Max are the functions that select the minimum and maximum among the input pairs, X is the input weed position and q is the output which has been controlled in the boundary range [L U]. Since proposed algorithm is combination of two algorithms IWO and PSO then it is consisted of these two algorithms parameters. HIWOPSO parameters with initial settings are mentioned in Table 1.

IV. Discussion and Evaluation

In this paper, the proposed algorithm has been evaluated with a set of benchmark functions which are a subgroup of unimodal and multimodal functions. These functions have various dimensions such as 2, 4, 10 and 30. Tables 3 highlights a multimodal test functions and Table 2 shows unimdal test functions.

TABLE I. PARAMETER SETTINGS

Parameter	definition	PSO	IWO	HIWOPSO
T	Maximum number of iterations	5000	5000	2000
N pop	Maximum population size	30	150	150
npop0	Initial population size	-	10	10
S_{min}	Minimum number of seeds	-	0	0
S_{max}	Maximum number of seeds	-	5	5
n	Nonlinear modulation index	-	2	2
$\sigma_{initial}$	Standard deviation initial value	-	1	1
σ_{final}	Standard deviation in al value	-	0.001	0.001
$c1$	cognitive/local weight	2	-	2
$c2$	social/global weight	2	-	2
ω	Inertia weight	1	-	1
Ω damp	Inertia weight reduction rate	0.9	-	0.9

TABLE II. UNIMODAL TEST FUNCTIONS (D: DIMENSIONS)

Function	D	Range	Min	Formulation
F1(Beale)	2	[-4.5,4.5]	0	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2$ $+(2.625 - x_1 + x_1x_2^3)^2$
F2(Easom)	2	[-100,100]	-1	$f(x) = -\cos(x_1)\cos(x_2)exp(-(x_1-\pi)^2 - (x_2-\pi)^2)$
F3(Matyas)	2	[-10,10]	0	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$
F4(Colville)	4	[-10,10]	0	$f(x) = 100(x_1^2 - x_2)^2 + (x_1-1)^2 + (x_3-1)^2$ $+90(x_3^2 - x_4)^2 + 10.1(x_2-1)^2 + (x_4-1)^2$ $+19.8(x_2-1)(x_4-1)$
F5(Zakharov)	10	[-5,10]	0	$f(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^4$
F6(Schwefel 2.22)	30	[-10,10]	0	$f(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$
F7(Schewefel 1.2)	30	[-100,100]	0	$f(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$
F8(Dixon-price)	30	[-10,10]	0	$f(x) = (x_1-1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_{i-1})^2$
F9(Step)	30	[-5.12,5.12]	0	$f(x) = \sum_{i=1}^{D} (x_i + 0\cdot 5)^2$
F10(Sphere)	30	[-100,100]	0	$f(x) = \sum_{i=1}^{D} x_i^2$
F11(SumSquares)	30	[-10,10]	0	$f(x) = \sum_{i=1}^{D} ix_i^2$
F12(Quartic)	30	[-1.28,1.28]	0	$f(x) = \sum_{i=1}^{D} ix_i^4 + Rand$

Multimodal test functions have many local minimums, and therefore, they are hard to be solved simply because of the trapping in local solutions. The (HIWOPSO) has been evaluated by these functions and search the global minimum for them. In Table 4, the results for algorithms GA [20], DE [21], PSO, BA [22] and IWO are compared with HIWOPSO. Conditions for experiment and parameter settings for these algorithms is explained in reference [23]. The experiment has been performed on the computer with following features: CPU 2.1 GHZ, Ram 8 GB and Matlab 2016 running on computer with windows 7. The NFE= 500,000 (number of function evaluation) was set as Stopping criteria and the values minimum than 1E-12, presented as 0 same as other methods. The mean value and Std Dev (standard deviation) have been calculated from 30 independent runs. In Table 4, the HIWOPSO found the minimum results for most of the functions with best standard deviations. The results of Friedman non-parameteric test [16] are also presented in this table and HIWOPSO could rank the best with value 2.75. The low p-value indicates that the results are remarkably different with each other and figure 2 depicts the results for this test with a bar diagram. For analyzing the procedure of convergence in (HIWOPSO), figure 3 has been presented. This figure shows the Convergence diagram for functions F8, F9, F18 and F19 in algorithms PSO, IWO and (HIWOPSO). As it stands, HIWOPSO has reached the desired minimum with

TABLE III. MULTIMODAL TEST FUNCTIONS (D: DIMENSIONS)

Function	D	Range	Min	Formulation
F13(Schaffer)	2	[-100,100]	0	$f(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2+x_2^2}) - 0.5}{(1+0.001(x_1^2+x_2^2))^2}$
F14(6 H Camel)	2	[-5,5]	-1.03163	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
F15(Bohachevsky2)	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)(4\pi x_2) + 0.3$
F16(Bohachevsky3)	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$
F17(Shubert)	2	[-10,10]	-186.73	$f(x) = (\sum_{i=1}^{5} i\cos(i+1)x_1 + i)$ $(\sum_{i=1}^{5} i\cos((i+1)x_2 + i))$
F18(Rosenbrock)	30	[-30,30]	0	$f(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$
F19(Griewank)	30	[-600,600]	0	$f(x) = \frac{1}{4000}\left(\sum_{i=1}^{D} (x_i - 100)^2\right)$ $-\left(\prod_{i=1}^{D} \cos\left(\frac{x_i - 100}{\sqrt{i}}\right)\right) + 1$
F20(Ackley)	30	[-32,32]	0	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e$
F21(Bohachevsky1)	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0\cdot 3\cos(3\pi x_1)$ $-0.4\cos(4\pi x_2) + 0.7$
F22(Booth)	2	[-10,10]	0	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
F23(Michalewicz2)	2	[0,π]	-1.8013	$f(x) = -\sum_{i=1}^{D} sin(x_i)(\sin(ix_i^2/\pi))^{20}$
F24(Michalewicz5)	5	[0,π]	-4.6877	$f(x) = -\sum_{i=1}^{D} sin(x_i)(\sin(ix_i^2/\pi))^{20}$
F25(Michalewicz10)	10	[0,π]	-9.6602	$f(x) = -\sum_{i=1}^{D} sin(x_i)(\sin(ix_i^2/\pi))^{20}$
F26(Rastrigin)	30	[-5.12,5.12]	0	$f(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$

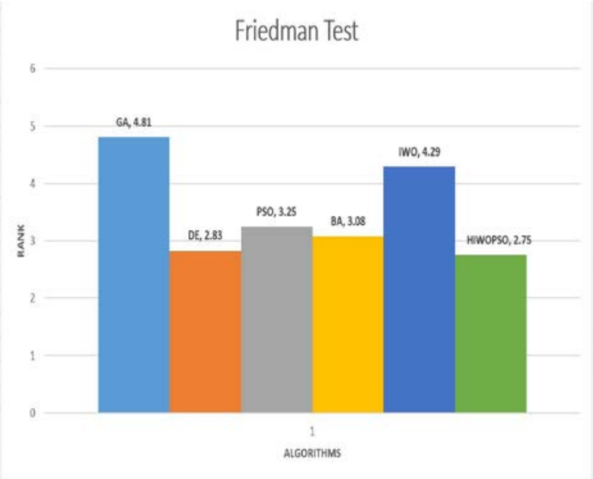


Fig. 2. Bar diagram for non-parametric Friedman test results for Functions F1-F26

minimum CPU Time and faster against the other algorithms IWO and PSO. Also, the test of normality

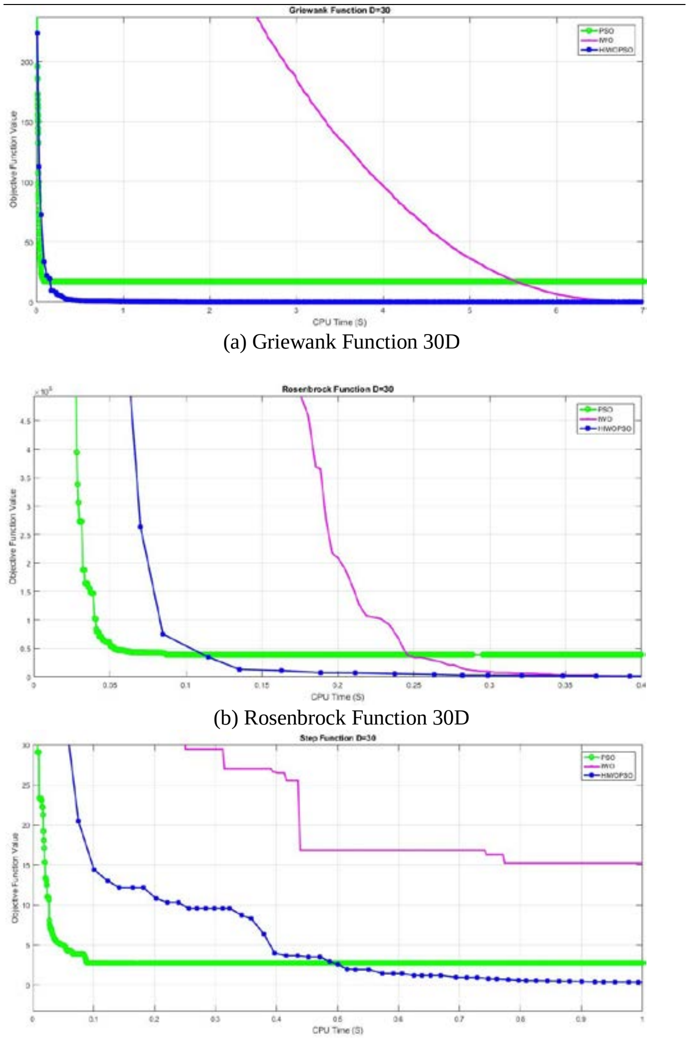
TABLE IV. HIWOPSO COMPARISON WITH GA, DE, PSO, BA, AND IWO (UNIMODAL FUNCTION SET), BOLD VALUES REPRESENT

Function	Criteria	GA	DE	PSO	BA	I	HIWOPSO
(F1)	Mea	0	0	0	1.88	2.	0
	Std	0	0	0	1.94	4.	0
(F2)	Mea	-1	-1	-1	-	-1	-1
	Std	0	0	0	4.50	0	0
(F3)	Mea	0	0	0	0	0	0
	Std	0	0	0	0	0	0
(F4)	Mea	0.01494	0.04	0	1.11	7.	0
	Std	0.00736	0.08	0	0.46	4.	0
(F5)	Mea	0.01336	0	0	0	1.	0
	Std	0.00453	0	0	0	3.	0
(F6)	Mea	11.0214	0	0	0	2.	1.42E-03
	Std	1.38686	0	0	0	0.	2.242585E-03
(F7)	Mea	7.40E+0	0	0	0	1.	147.401395
	Std	1.14E+0	0	0	0	3	448.571186
(F8)	Mea	1.22E+0	0.66	0.66667	0.66	0.	0.66667
	Std	2.66E+0	E-9	E-8	1.16	0	0
(F9)	Mea	1.17E+0	0	0	5.12	2.	0
	Std	76.56145	0	0	0.39	1.	0
(F10)	Mea	1.11E+0	0	0	0	2.	0
	Std	74.21447	0	0	0	4.	0
(F11)	Mea	1.48E+0	0	0	0	2.	0
	Std	12.40929	0	0	0	4.	0
(F12)	Mea	0.18070	0.00	0.00116	1.72	1.	3.66E-03
	Std	0.02712	0.00	0.00028	1.85	0.	0.001401347
(F13)	Mea	0.00424	0	0	0	7.	0
	Std	0.00476	0	0	0	6.	0
(F14)	Mea	-1.03163	-	-1.03163	-	-	-1.03163
	Std	0	0	0	0	0	0
(F15)	Mea	0.06829	0	0	0	1.	0
	Std	0.07822	0	0	0	7.	0
(F16)	Mea	0	0	0	0	1.	0
	Std	0	0	0	0	2.	0
(F17)	Mea	-186.73	-	-186.73	-	-	-186.73
	Std	0	0	0	0	0	0
(F18)	Mea	1.96E+0	18.2	15.0886	28.8	5	0.03137
	Std	3.85E+0	5.03	24.1701	0.10	3	0.02992
(F19)	Mea	10.63346	0.00	0.01739	0	0.	0
	Std	1.16146	0.00	0.02081	0	0.	0
(F20)	Mea	14.67178	0	0.16462	0	0.	5.62617E-05
	Std	0.17814	0	0.49387	0	0.	0.0001823
(F21)	Mea	0	0	0	0	4.	0
	Std	0	0	0	0	3.	0
(F22)	Mea	0	0	0	0.00	3.	0
	Std	0	0	0	0.00	2.	0
(F23)	Mea	-1.8013	-	-1.57287	-	-	-1.8013
	Std	0	0	0.11986	0	0	0
(F24)	Mea	-4.64483	-	-2.4908	-	-	-4.6877
	Std	0.09785	0.01	0.25695	0	0	0
(F25)	Mea	-9.49683	-	-4.0071	-	-	-9.6602
	Std	0.14112	0.06	0.50263	0	0.	0
(F26)	Mea	52.92259	11.7	43.9771	0	2	0
	Std	4.56486		2.53817		2.	0

of Kolmogorov-Smirnova and Shapiro-Wilk [16] has been performed for the four functions of F6, F7, F12 and F20 which are hard to be solved, and here, standard deviation is not zero for them. Table 5 presents the results for this test and the p-value, df (degree of freedom), and the statistics for this test are also presented. Accordingly, the p-value which is higher than the significant _ =0.05 is considered a normal distribution and the lower than that value is supposed to be an abnormal distribution. Based on the p-value and the test of normality of Kolmogorov-Smirnova, the function of F7 result is normal and other functions have the abnormal distribution. Figure 4 shows the normal and abnormal distributions, the histogram, QQ-plot and Box-plot for the two functions, F7 and F12. In this figure, F7 is a normal distribution and F12 is considered abnormal. As can be inferred, in normal distributions, the results are in one diagonal line for QQ-plot while in abnormal distributions, this fact does not hold true.

TABLE V. TEST OF NORMALITY KOLMOGOROV-SMIRNOVA AND HAPIRO-WILK FOR FUNCTIONS F7,F25 AND F12

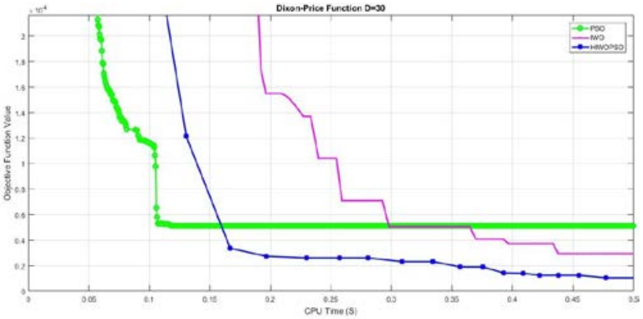
	Kolmogorov-Smirnova			Shapiro-Wilk		
	Statistic	df	p-value	Statistic	df	p-value
F6	0.239999	30	0.000	0.858999	30	0.000
F7	0.1259999	30	0.103	0.858999	30	4.0532E-3
F12	0.6268000	30	0.000	0.3579999	30	0.000
F20	0.3219999	30	0.000	0.4839999	30	0.000



(a) Griewank Function 30D

(b) Rosenbrock Function 30D

(c) Step Function 30D



(d) Dixon-Price Function 30D

Fig. 3. Convergence diagram for functions F8, F9, F18 and F19 in algorithms PSO, IWO and HIWOPSO

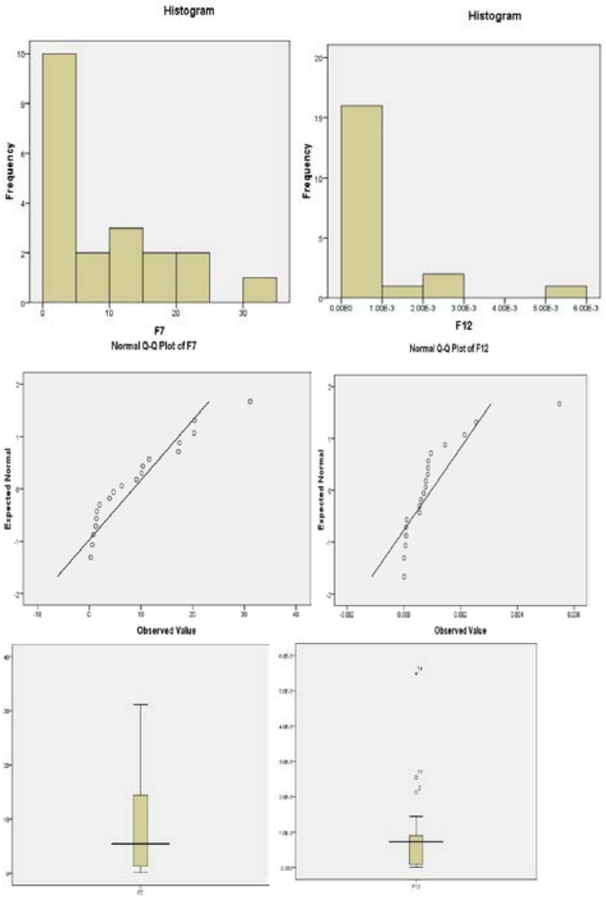


Fig. 4. HIWOPSO algorithm. Left plots are normal for F7 and the Right are abnormal for F12

as studies presented in section 2. Figure (4) and (5) shows these comparisons.

V. Conclusion and Future Works

The IWO is, arguably, one of the most efficient nature inspired Metaheuristic algorithms, which has outperformed most of the algorithms in solving the various optimizing numerical problems. Furthermore, one of the practical Metaheuristic algorithms which have been used most widely in the optimization is the PSO. The PSO algorithm is a simple and practical algorithm which is more amenable to combination with others. In the current study, the PSO and IWO algorithms are combined in order to design a new hybrid method for seeking the global solution. The proposed hybrid algorithm begins to search by creating solutions by IWO and PSO algorithms and used the better one as the input to the mutation function. Mutation output is compared with the input and better one selected as the new seed. The experiment based on benchmark functions and non-parametric ranking showed that the proposed hybrid is more dominant and competent than other famous algorithms. Moreover, based on the results of the test of normality and convergence, this proposed hybrid algorithm mostly had abnormal distributions for results and converged to the optimum solutions in minimum CPU Time.

According to the description and assumptions made in this study, the following works are suggested:1.Using compound method for solving engineering problems2.using Compound method for data clustering 3.resolving the distribution of load in power plants according to posed algorithm.

References

[1] C. Grosan, A. Abraham, Intelligent systems a modern approach, Intelligent Systems Reference Library, Springer Berlin Heidelberg, 17 (2011).

[2] X-S Yang, Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics. Cambridge International Science Publishing , (2008).

[3] P. Hansen, N. Mladenovic, J. Brimberg, J. Moreno-Prez, Variable neighbor- hood Search .In :Handbook of Metaheuristic s.Springer US, (2010) 61 86.

[4] B. Dogan, T. □ Olmez, A new Metaheuristic for numerical function optimization: Vortex search algorithm, Information Sciences, 293 (2015) 125{145.

[5] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Advances in Engineering Software, 95 (2016) 51-67.

[6] A. Sadollah , A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm for optimization of truss structures with discrete variables, Computers & Structures, 103 (2012) 49-63.

[7] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm a novel Metaheuristic optimization method for solving constrained engineering optimization problems, Computers & Structures, 111 (2012) 151-166.

[8] H. Salimi, Stochastic fractal search: A powerful Metaheuristic algorithm, Knowledge-Based Systems, 75 (2015) 1-18.

[9] J. Kennedy, R. Eberhart, Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. Perth, Australia, (1995) 1942-1948.

[10] T. KAY CHEN , Advances in Swarm Intelligence, First International Conference, ICSI 2010, Beijing, China, June 12-15, 2010, Proceedings, Part I, Springer-Verlag Berlin Heidelberg. 6145 (2010).

[11] Mehrabian AR, Lucas C. A novel numerical optimization algorithm inspiredfrom weed colonization. Ecol Inform 2006;1:35566.

[12] Mojtaba Ghasemi, Sahand Ghavidel, Mohammad Mehdi Ghanbarian, Amir Habibi, A new hybrid algorithm for optimal reactive power dispatch problem with discrete and continuous control variables, Applied Soft Computing, Volume 22, September 2014, Pages 126-140

[13] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, Proc. IEEE Cong. Evol.Comput. (2007) 46614667.

[14] Santosh Kumar Mahto, Arvind Choubey, A novel hybrid IWO/WDO algorithm for nulling pattern synthesis of uniformly spaced linear and non-uniform circular array antenna, AEU – International Journal of Electronics and Communications, Volume 70, Issue 6, June 2016, Pages 750-756

[15] Bayraktar Z, Komurcu M, Bossard JA, Werner DH. The wind driven optimization technique and its application in electromagnetic . IEEE Trans Antennas Propag2013;61:274557.

[16] S. Garcia, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior : a case study on the CEC'2005 Special Session on Real Parameter Optimization. Journal of Heuristics, 15 (2009) 617-644.

[17] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, (2004).

[18] H. Duan, D. Wang, X. Yu, Markov Chains and Martingale Theory Based Convergence Proof of Ant Colony Algorithm and Its Simulation Platform. in Intelligent Control and Automation, 1 (2006) 3057{3061.

[19] W. Chu, X. Gao, S. Sorooshian, Handling boundary constraints for particle swarm optimization in high-dimensional search space. Information Sciences, 181 (2011) 4569-4581.

[20] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, (1989).

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

[22] D. Pham, A. Ghanbarzadeh, E. Ko, S. Otri, S. Rahim, M. Zaidi, The bees algorithm a novel tool for complex optimization problems, in: D. Pham, E. Eldukhri, A. Soroka (Eds.), Intelligent Production Machines and Systems, Elsevier Science Ltd, Oxford, (2006) 454-459.

[23] M. Cheng, L. Lien, Hybrid artificial intelligence based pba for benchmark functions and facility layout design optimization, Journal of Computing in Civil Engineering, 26 (2012) 612-624.