Journal of Theoretical and Applied Information Technology
28th February 2017. Vol.95. No 4


ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# HYBRIDIZING INVASIVE WEED OPTIMIZATION WITH FIREFLY ALGORITHM FOR UNCONSTRAINED AND CONSTRAINED OPTIMIZATION PROBLEMS

[1]HYREIL A KASDIRIN, [2]N. M. YAHYA, [1]M. S. M. ARAS, [3]M. O. TOKHI

[1]Senior lecturer, Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, MALAYSIA

[2]Senior lecturer, Faculty of Electrical Engineering, Universiti Malaysia Pahang, MALAYSIA

[3]Professor, School of Engineering, London South Bank University, UNITED KINGDOM

E-mail: [1]hyreil@utem.edu.my, [3]tokhim@lsbu.ac.uk

**ABSTRACT**

This study presents a hybrid invasive weed firefly optimization (HIWFO) algorithm for global optimization problems. Unconstrained and constrained optimization problems with continuous design variables are used to illustrate the effectiveness and robustness of the proposed algorithm. The firefly algorithm (FA) is effective in local search, but can easily get trapped in local optima. The invasive weed optimization (IWO) algorithm, on the other hand, is effective in accurate global search, but not in local search. Therefore, the idea of hybridization between IWO and FA is to achieve a more robust optimization technique, especially to compensate for the deficiencies of the individual algorithms. In the proposed algorithm, the firefly method is embedded into IWO to enhance the local search capability of IWO algorithm that already has very good exploration capability. The performance of the proposed method is assessed with four well-known unconstrained problems and four practical constrained problems. Comparative assessments of performance of the proposed algorithm with the original FA and IWO are carried out on the unconstrained problems and with several other hybrid methods reported in the literature on the practical constrained problems, to illustrate its effectiveness. Simulation results show that the proposed HIWFO algorithm has superior searching quality and robustness than the approaches considered.

**Keywords:** *Hybrid algorithm, invasive weed optimization, firefly algorithm, unconstrained problem, practical design problem.*

## 1. INTRODUCTION

In science and engineering applications, many problems that are encountered can be considered as optimization problems. These optimization problems can be either constrained or unconstrained. Regardless of the complexity and high dimensionality issues, and computational cost of current numerical methods, solving those optimization problems is still a challenge. Recent biologically inspired algorithms are shown to be capable of solving such problems more efficiently. In recent years, the biologically inspired algorithms have been adopted to solve hard optimization problems and they have shown great potential in solving complex engineering optimization problems (Yang and He, 2013). Numerous biologically inspired algorithms have been developed, and these include population-based algorithms such as particle swarm optimization (PSO), ant colony optimization (ACO), firefly algorithm (FA), invasive weed optimization (IWO) and artificial plant optimization algorithm (APOA). The success of these methods depends on their ability to maintain proper balance between exploration and exploitation by using a set of candidate solutions and improving them from one generation to another generation. Exploitation refers to the ability of the algorithm to apply knowledge of previously discovered good solutions to better guide the search towards the global optimum. Exploration, on the other hand, refers to the ability of the algorithm to investigate unknown and less promising regions in the search space to avoid getting trapped in local optima.

Swarm intelligence based algorithms represent an important class of population-based optimization algorithms, and the firefly algorithm falls within this category. The algorithm is inspired from social behaviour of firefly (Yang, 2010), and is much simpler in concept and implementation than other swarm algorithms because it has the advantage of finding optimal solution with its exploitation capability. For that reason, it has attracted much attention to solve various optimization problems (Hachino et al., 2013; Marichelvam et al., 2013; Nikman et al., 2012; Olamaei et al., 2013; Sayadi et al., 2013). However, the algorithm is subject to getting easily trapped in local optima and is not efficient in achieving global solution.

Another class of population-based optimization model is inspired from common ecological phenomena. One of the promising recent developments in this field is the IWO algorithm, which was initially proposed by Mehrabian and Lucas (2006). The algorithm is inspired by the natural ecological phenomenon and mimics the behaviour of weeds occupying suitable place to grow, reproduce and colonize the area. It has robustness, adaptation, and randomness features and is simple but effective with accurate global search ability. The algorithm has been applied to many engineering and non-engineering fields (Zaharis et al., 2013; Nikoofard, 2012; Pahlavani et al., 2012).

A drawback of the FA is that it always gets trapped in local optima (Farahani et al., 2012). On the other hand, Yin et al (2012) has stressed that the drawbacks of IWO are that it suffers specifically from low solution precision, tuning to get stuck in local optima and premature convergence. Instead of improving the algorithm, many researchers tend to use a hybrid method by combining two or more algorithms in a complementary manner to resolve drawbacks of the constituent algorithms. Several works have been reported on hybridizing with FA such as hybrid with levy flight (Yang, 2010c), ACO (El-Sawy et al., 2013), differential evolution (Abdullah et al., 2012) and genetic algorithm (Farhani et al., 2012). Consequently, IWO also has been hybridized with other metaheuristic algorithms to improve its capability such as with cultural algorithm (Zhang et al., 2008), PSO (Hajimirsadeghi and Lucas, 2009), evolutionary algorithm (Zhang et al., 2010), memetic algorithm (Sengupta et al., 2012) and with group search optimizer (Roy et al., 2013).

In this paper, a new hybrid algorithm based on the population diversity of IWO and the swarm population based on FA is proposed, and referred to hybrid invasive weed-firefly optimization (HIWFO) algorithm. The proposed HIWFO algorithm integrates IWO with FA to solve unconstrained and practical constrained optimization problems. The performance of the HIWFO is demonstrated through tests with a set of benchmark functions of unconstrained problems and four practical constrained problems. The organization of the paper is as follows; Sections 2 and 3 describe the original IWO and FA algorithms, respectively. In section 4, the HIWFO algorithm is introduced and described. Section 5 describes the experimental set-up and presents performance investigations with benchmark functions of unconstrained and practical constrained problems. The analysis and evaluation of the results are also elaborated in the section. Finally, conclusions drawn from the work are presented in section 6.

## 2. INVASIVE WEED OPTIMIZATION

IWO is an ecologically inspired optimization algorithm based on colonizing of weeds, introduced by Mehrabian and Lucas (2006). The IWO algorithm mimics the natural behaviour of weeds in colonizing and searching a suitable place for growth and reproduction. Weeds are vigorously invasive and robust plants able to adapt to changes in the environment, making them a threat to agriculture. The robustness, adaptation and randomness of the algorithm are shown by imitating a natural phenomenon of invasive weeds.

In the IWO algorithm, the process simulates the survival of weeds colony, where it begins with initializing the initial plant in the search area. The plant is spread randomly in the search place. Each member is able to produce seeds. However, production of seeds depends on their relative fitness in the population. The worst member produces a minimum number of seeds ($s_{min}$) and the best produces the maximum number of seeds ($s_{max}$) where the weeds production of each member is linearly increased. After that, the seeds are randomly scattered over the search space near to its parent plant. The scattering process uses a normally distributed random number with standard deviation (SD) given as

$$\sigma_{iter} = \left[\frac{iter_{max} - iter}{iter_{max}}\right]^n (\sigma_{max} - \sigma_{min}) + \sigma_{min} \quad (1)$$

where $iter_{max}$ is maximum number of iterations, $iter$ is current iteration, $n$ is the nonlinear modulation index, $\sigma_{max}$ is usually initial SD and $\sigma_{min}$ is the final SD in the optimization process. The seeds with their respective parent plants are considered as potential solution for subsequent generations. In order to maintain the size of population in the search area, the algorithm conducts a competitive exclusion strategy, where an elimination mechanism is employed; if the population exceeds maximum size only the plants with better fitness are allowed to survive. Those with better fitness produce more seeds and with high possibility of survival and become reproductive. The process continues until the maximum number of iterations is reached and the plant with best fitness is closest to the optimal solution. Algorithm 1 shows pseudo code of the IWO algorithm.

## 3. FIREFLY ALGORITHM

FA is a population-based optimization algorithm and in the family of swarm intelligence algorithms introduced by Yang (2008; 2009; 2010). It is inspired by the social behaviour of a group of fireflies that interact and communicate via the phenomenon of bioluminescence produced in the insect body.

Yang (2008, 2010) suggests that each firefly will produce its own light intensity that determines the brightness of the firefly. The variation of light intensity produced is associated with the encoded objective function. For a firefly to move to another brighter firefly, assuming that a firefly $j$ is more attractive than firefly $i$, the movement of firefly $i$, towards firefly $j$ is determined by;

$$x_{i+1} = x_i + \beta_0 e^{-\gamma r^2}(x_j - x_i) + \alpha \epsilon_i \qquad (2)$$

where the third term is a randomization term which consists of randomization coefficient, $\alpha$ with the vector of random variable, $\epsilon_i$ from Gaussian distribution. Algorithm 2 shows pseudo code of the firefly algorithm.

## 4. HYBRID INVASIVE WEED FIREFLY OPTIMIZATION ALGORITHM

Based on the introduction of IWO and FA in the previous section, the combination of the two approaches is described in this section. The idea of this hybridization is to obtain a more robust optimization technique, especially to compensate for deficiencies of the individual algorithms. Therefore, in this work, a hybrid algorithm is proposed by inducing FA into IWO, referred to as hybrid invasive weed firefly optimization (HIWFO) algorithm. The strategy utilizes the spatial dispersion of IWO and firefly movement to explore new areas in the search space and exploit the population, respectively. Therefore, it can overcome the lack of exploration of the original FA and improve the low solution precision of the IWO. In other words, hybridization not only improves the performance, it also improves the accuracy of the constituent algorithms. This combination improves the capability of optimization procedure by updating the solution to accelerate the convergence speed for more accurate fitness values with less computational time.

The biggest advantage of IWO algorithm constitutes its capability of global exploration and diversity search. In the algorithm, the initial weeds are dispersed over the search space randomly to produce new seeds. Selection of better plants (spatial dispersion) from the population consisting of weeds and seeds continues until the maximum number of plants is reached. The spatial dispersion in the algorithm strives to improve the population diversity to avoid premature convergence and make the algorithm more robust. The optimization algorithm is enhanced by cooperation of FA so that each seed in the iteration can move towards the best individual in the current iteration. Hence, the enhanced algorithm not only ensures the individual diversity by IWO, but also improves the optimization accuracy and the speed of the algorithm.

The boundary re-adjustment scheme is placed after the movement process at the end of the iteration to ensure the population is within the search space. The action also helps each member of the population to stay within the boundary and ready for the next iteration. Therefore, the steps of the proposed HIWFO algorithm are best described as follows:

[Step 1] Initialization

Initialize the parameters of invasive weed and firefly algorithm, the dimension and boundary limit of the search space. Initialize the population of the hybrid algorithm. A population of initial seeds of plant is dispersed over a search space with random positions. By using the designated objective function, each seed's fitness value could be calculated based on its initial position.

[Step 2] Update the following parameters:

The production and distribution of weed(s) by plant. Each plant produces seeds and this increases linearly from the minimum to its maximum possible seeds production.

$$weed_{x_i} = \frac{f_{x_i} - f_{min}}{f_{max} - f_{min}}(s_{max} - s_{min}) + s_{min} \qquad (3)$$

where $f_{x_i}$ is the weed's fitness at current population, $f_{max}$ is the maximum fitness of the current population, $f_{min}$ is the minimum fitness of the same population, $s_{max}$ and $s_{min}$ respectively represent the maximum and the minimum values of a seed. The parameter of light absorption coefficient, γ, attraction coefficient, β and randomization coefficient, α remain constant as suggested by Yang (2009).

[Step 3] Reproduction loop: Iteration = iteration + 1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Each seed grows into plant in the population capable of reproducing seeds but according to its fitness, where the fitter plants produce more seeds.

[Step 4] Spatial dispersion

The seeds generation is randomly distributed in the search area according to normal distribution with zero mean and standard deviation (SD). The normalized SD per iteration, $\sigma_{iter}$ is as given in equation (1).

[Step 5] Competitive exclusion

The population of plants is controlled by the fitness of the plants. If the population has reached its maximum size, the elimination process runs on the poor fitness plants where only plants with better fitness are allowed to survive. This elimination process or competitive exclusion is employed from generation to generation until it reaches its maximum number of generations / iterations of the algorithm. At the end of the algorithm, the seeds and their respective parents are ranked together and have chance to grow in the search area and reproduce seeds as mentioned in step (2). Those with better fitness produce more seeds and have high possibility of survival and become reproductive. The processes continue until the maximum number of iterations is reached and the plant with best fitness is expectedly closest to the optimum solution.

[Step 6] Improve the local search by localization.

The fitness value of each plant is equal to the light intensity of the firefly algorithm. Therefore, the firefly algorithm's mechanism is started. The position of the plant, $x_{i+1}$is updated by using equation (2) in a highly random manner. The plant with lower fitness value essentially has low light intensity, and will approach and move towards higher light intensity.

[Step 7] Boundary checking mechanism

With the random movement in Step 6 members of the population will have tendency to move beyond the boundary. The boundary checking mechanism is used to avoid any member of the population jump out of the boundary of the problem.

[Step 8] The result of the algorithm for the iteration is updated and if the maximum number of iterations has not reached, the next generation of the plant starts in the loop.

The main steps of the proposed HIWFO approach can be summarized in pseudo code as in Algorithm 3.

## 5. EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the experimental results assessing the performance of the proposed hybrid algorithm. Two types of tests are considered. The first set of tests involves four well-known unconstrained optimization problems that consist of unimodal and multimodal benchmark functions and the second set of tests involves test used four structural engineering applications that deal with continuous variables in constrained optimization problems.

The algorithms are implemented and tested using a personal computer (PC) with processor CPU Intel (R) Core (TM) i5-2400 with Windows 7 Professional operating system, frequency of 3.10 GHz and memory installed of 4.00 GB RAM. The program is coded in MATLAB R2012a. Each problem is tested with 30 independent runs with a minimum number of function evaluations of 30000 per run.

### 5.1 Test 1: Unconstrained Optimization Problems

This section examines the set-up test for unconstrained optimization problems. Four well-known benchmark functions, shown in Table 1, are used to evaluate the performance of HIWFO in solving unconstrained optimization problems. In Table 1, D represents the number of dimensions and for this test, three variations, namely D = 10, 30 and 50 are used. The variable range, fitness optimum, and type of problem whether U = unimodal function or M = multimodal function are also shown in Table 1. All benchmark functions have their global optima as 0.

The benchmark function that has single optimum is called unimodal (U) whereas if it has more than one optimum, it is called multimodal (M). Multimodal functions are used to test the ability of the algorithm to escape from local optima and locate a good near-global optimum. Therefore, for the case of multimodal functions especially in high dimensions, the final results are very important than the convergence rates. The experiment also looks at how effective the algorithm could be extended for higher dimension problems, although this also will involve increased computational complexity.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The tests and performance results of the proposed hybrid algorithm are also compared with the performance of original FA and IWO algorithms. Table 2 shows the parameter sets used in the tests where $\sigma_{initial}$ and $\sigma_{final}$, represent the initial and final values of SD respectively, $s_{max}$ and $s_{min}$, represent the maximum and the minimum values of a seed respectively, $\gamma$, light absorption coefficient, $\beta$, attraction coefficient, and $\alpha$, randomization coefficient used in the algorithms. As noted in Table 2, the algorithms also used the same population size, n and the maximum number of iterations for a fair comparative evaluation. The initial population for each algorithm is randomly positioned in the search space.

In the tests, 30 independent runs of the three algorithms were carried out on each function with three different dimensions (i.e., D = 10, 30 and 50). The average of the final solutions, the best solution and their respective standard deviations are noted.

Table 3 compares the algorithms with the quality of optimum solution over the four benchmark functions used. The mean and standard deviation of 30 independent runs for each of the three algorithms are shown in Table 3, where the best mean solution in each case has been marked in bold font.

Table 4 shows the performance comparison of the best values and worst results of the three algorithms for functions $f_1$ – $f_4$. From Tables 3 and 4 it can be seen that HIWFO achieved better results in both low and high dimensions for all the benchmark functions in terms of search precision and robustness.

The rates of convergence of the algorithms achieved with the benchmark functions are shown in Figure 1, where only results for 30 dimension functions are shown as representative sample. It is noted that the proposed hybrid algorithm, outperformed the classical FA and IWO in reaching the optimal solution. For Figures 1(a), 1(b) and 1(c), the test functions each has one local optimum point, whereas the functions in Figures 1(d), 1(e) and 1(f) each has many local optima. The classical FA seems to have got trapped at the local optimum especially in case of De Jong, Rosenbrock, Rastrigin and Griewank functions. The IWO got easily trapped in the local optima for Rastrigin and Griewank functions. On the other hand, compared to FA and IWO, the HIWFO algorithm improved the situation and also showed exploitation of local search with faster convergence. The hybrid algorithm further showed tendency to get better result as the number of iterations increased. Based on the results in Tables 3 and 4 and Figure 1, it is clear that HIWFO outperformed the original FA and IWO in the unconstrained benchmark tests.

## 5.2 Test 2: Practical Constrained Optimization Problems

The performance of the proposed algorithm is tested and the result presented in this section using four typical engineering constrained design problems that have widely been used in the literature. The performance of the algorithm is also assessed in comparison to those of four known hybrid algorithms, namely co-evolutionary particle swarm optimization approach; CPSO (He and Wang, 2007), integration PSO wih DE; PSO-DE (Lui et al., 2010), hybrid charges system search and PSO; CSS-PSO (Kaveh and Talatahari, 2011) and hybrid glowworm swarm optimization; HGSO (Zhou et al., 2013) and with FA (Gandomi et al, 2011) to verify the reliability and validity of the algorithm. Generally, a constrained optimization problem is best described as follows:

$$\text{Minimize } f(\vec{x}), \vec{x} = [x_1, x_2, \dots, x_n] \tag{5}$$

Subject to:

$$g_i(x) \leq 0, for\ i = 1, \dots, q \tag{6}$$

$$h_j(x) = 0, for\ j = 1, \dots, m \tag{7}$$

However, for the equality constraints handling, the equations are transformed into inequalities of the form

$$|h_j(x)| - \varepsilon \leq 0, for\ j = 1, \dots, m \tag{8}$$

where a solution $\vec{x}$ is regarded as feasible solution if and only if $g_i(x) \leq 0$ and $|h_j(x)| - \varepsilon \leq 0$ with $\varepsilon$ a very small number. The presence of constraints in any optimization problem may have significant effect on the performance of the optimization algorithm. In this paper, penalty function method is used to solve the constrained optimization problem. The penalty function method is a popular method used as compared to most traditional algorithms that are usually based on the concept of gradient. This method is easy to implement and is often chosen due to its simplicity (He and Wang, 2007). With this method, the constrained optimization problem is transformed to unconstrained optimization problem that is simpler to solve. The proposed hybrid algorithm handles the practical optimization problems with constraints as described below.

### 5.2.1 Welded beam design problem

The welded beam structure is often used as benchmark problem for testing optimisation methods with constraints problems where it was first described by Coello (2000) is often used as benchmark for testing optimization methods with constrained problems. The problem is designed to find the minimum fabricating cost *f(x)* of the welded beam subject to constraints on shear stress ($\tau$), bending stress in the beam ($\theta$), buckling load on the bar ($P_c$), end deflection of the beam ($\delta$) and side constraint. In this problem, there are four optimization design variables to be considered, that is the thickness of the weld ($h$), the length of the welded joint ($l$), the width of the beam ($t$) and the thickness of the beam ($b$). The mathematical formulation of the cost function, their respective constraint functions and variable regions are as shown in Appendix A1.

He and Wang (2007), Lui et al (2010), Kaveh and Talahari (2010) solved this problem using PSO-based hybrid methods. Zhou et al (2013) used hybrid glowworm swarm optimization (HGSO) to solve this problem. Gandomi et al (2011) examined the handling of FA with this constrained structural optimization problem. Table 5 shows the statistical results obtained with the different approaches and with the proposed hybrid algorithm. It can be noted that the best feasible solution found by HIWFO algorithm was better than the best solutions found by other approaches with relatively small standard deviation, although PSO-DE and HGSO were better in the average searching quality and worst solution.

### 5.2.2 Tension / compression spring design problem

The tension / compression spring design is also one of the practical benchmark problems, The problem is well described by Belegundu (1982) and Arora (1989), where the design is to minimize the weight of a tension / compression spring subject to constraints on minimum deflection, shear stress and surge frequency. For this problem, the design variables are the mean coil diameter, $D$ ($x_1$), the wire diameter, $d$ ($x_2$) and the number of active coils, $N$ ($x_3$). The cost function, their respective constraints and the variable regions are as shown in Appendix A2.

This problem has been solved by using co-evolutionary particle swarm optimization (CPSO) algorithm (He and Wang, 2007) and hybrid PSO with differential evolution (PSO-DE) algorithm (Lui et al, 2010). Moreover, Kaveh and Talahari (2011) employed charged system with PSO, Zhou et al (2013) used hybrid glowworm swarm optimization (HGSO) to solve this problem. Table 6 presents statistical results obtained with the proposed hybrid algorithm and the algorithms reported by the researchers mentioned. It is noted that the best feasible solution and the mean solution obtained by HIWFO algorithm were better than those previously reported. The standard deviation of the proposed algorithm was also relatively very small.

### 5.2.3 Pressure vessel design problem

The pressure vessel design problem is a practical problem often used as benchmark for testing optimization methods. The objective is to find the minimum total cost of fabrication, including the costs from a combination of welding, material and forming. The thickness of the cylindrical skin, $T_s$ ($x_1$), the thickness of the spherical head, ($T_h$) ($x_2$), the inner radius, $R$ ($x_3$), and the length of the cylindrical segment of the vessel, $L$ ($x_4$) were included as optimization design variables of the problem. The cost function, constraint functions and ranges of variables are stated in Appendix A3.

The problem has been solved using co-evolutionary PSO (He and Wang, 2007), PSO-DE (Lui et al, 2010), hybrid charged system with PSO (Kaveh and Talahari, 2011) and HGSO (Zhou et al, 2013). Gandomi et al. (2011) examined the handling of FA with constrained structural optimization problems. Table 7 shows the best solutions obtained with these algorithms and the HIWFO. It can be seed in Table 8, that the best solution found by HIWFO was better than the best solutions found by the hybrid techniques considered. Table 8 also shows that FA performed slightly better in the best and average searching results as compared with HIWFO, however, the proposed method achieved better quality on the worst result and lower standard deviation.

### 5.2.4 Speed reducer design problem

The speed reducer problem is also one of the practical problems used as benchmark problem for testing optimization methods. In this constrained optimization problem, the design is to minimize the weight of speed reducer subject to constraints of bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts. The minimum cost function, their respective constraint functions and ranges of variables are stated in Appendix A4.

In the literature, Lui et al (2010) used hybridizing PSO with differential evolution (PSO-DE), Kaveh and Talahari (2011) employed charged system with PSO and Zhou et al (2013) used hybrid glowworm swarm optimization (HGSO) to solve this problem. The statistical simulation results obtained by the approaches mentioned with the proposed hybrid method are listed in Table 8. It can be seen that the best solution and the average search quality of HIWFO algorithm were better than those of other mentioned methods. However, the proposed hybrid method showed the largest standard deviation as compared with the other methods.

## 6. CONCLUSION

A new hybrid algorithm based on hybridization of the invasive weed and firefly algorithms has been proposed to solve unconstrained and constrained optimization problems. The hybridization of the algorithms has been achieved by embedding the FA method into IWO algorithm structure to enhance the local search capability of IWO that already has very good exploration capability. Simulation results based on four well-known unconstrained problems have demonstrated the effectiveness, efficiency and robustness of the proposed method. In addition, based on the simulation results and comparisons of the practical constrained problems, it can be concluded that the HIWFO algorithm offers superior search quality and robustness. The parameters of invasive weed and firefly algorithms can be modified to further enhance their search capability. Moreover, incorporating suitable adaptive parameters of the algorithm could further improve the diversity mechanism in the HIWFO algorithm to further balance the exploration and exploitation abilities to achieve better performance. Furthermore, future work will look at solving real world optimization problems using this hybrid technique.

## ACKNOWLEDGEMENTS:

Hyreil A. Kasdirin and M. S. M. Aras are researchers from Center for Robotic and Industrial Automation (CeRIA), Center of Excellence, Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka (UTeM) and financially supported by the Ministry of Education of Malaysia.

## REFERENCES:

[1] S. Abdullah, M. Deris, S. Mohamad, S.Z. Mohd Hashim, "A new hybrid firefly algorithm for complex and nonlinear problem", Distributed Computing and Artificial Advances in Intelligent and Soft Computing 151, 2012, pp. 673 – 680.

[2] R. Akhbari, K. Ziarati, "A Cooperative Approach to Bee Swarm Optimization", Journal of Information Science and Engineering 27, 2011, pp. 799 – 818.

[3] J.S. Arora, "Introduction to Optimum Design", McGraw-Hill, New York, 1989.

[4] A. D. Belegundu, and J. S. Arora, "A study of mathematical programming methods for structural optimization", Report No. CAD-SS-82.5, Division of Materials Engineering, University of Iowa City, Iowa, August 1982.

[5] C.A. Coello, "Use of A Self-adaptive Penalty Approach for Engineering Optimization Problems", Computers in Industry 41, 2000, pp. 113 – 127.

[6] K. Deep, and K. N. Das, "A novel hybrid genetic algorithm for constrained optimization". *International Journal of System Assurance Engineering and Management*, *4*(1), 2013, pp. 86-93.

[7] A.A. El-Sawy, E.M. Zaki, R.M. Rizk-Allah, "A novel hybrid ant colony optimization and firefly algorithm for solving constrained engineering design problems", Journal of Natural Sciences and Mathematics 6 (1), 2013, pp 1 – 22.

[8] S.M. Farahani, A.A. Abshouri, B. Nasiri, M.R. Meybodi, "Some hybrid models to improve firefly algorithm performance", International Journal of Artificial Intelligent 8 (S12), 2012.

[9] A.H. Gandomi, X.S. Yang, A.H. Alavi, "Mixed variable structural optimization using firefly algorithm", Computers and Structures 89, 2011, pp. 2325 – 2336.

[10] A.H. Gandomi, X.S. Yang, S. Talahari, A.H. Alavi, "Firefly algorithm with chaos", Communications in Nonlinear Science and Numerical Simulation 18 (1), 2013, pp. 89 – 98.

[11] L. Guo, G.G. Wang, H. Wang, D. Wang, "An effective hybrid firefly algorithm with harmony search for global numerical optimization", The Scientific World Journal 2013 (2013), Article ID 125625, 9 pages, http://dx.doi.org/10.1155/2013/125625.

[12] H. Hajimirsadeghi, and C. Lucas, "A hybrid IWO/PSO algorithm for fast and global optimization", In proceeding International IEEE Conference Devoted to 150 Anniversary of Alexander Popov, Saint Petersburg, Russia. IEEE EUROCON 2009, pp. 1964 – 1971.

[13] Q. He, and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", Engineering Applications of Artificial Intelligence 20, 2007, pp. 89 – 99.

[14] A. Kaveh, and S. Talahari, "Hybrid charges system search and particle swarm optimization for engineering design problems", Engineering Computations 28 (4), 2011, pp. 423 – 440.

[15] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization", Applied Soft Computing 10, 2009, pp. 629 – 640.

[16] R. Mehrabian, and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization", Ecological Informatics 1 (4), 2006, pp. 355 – 366.

[17] A.H. Nikoofard, H. Hajimirsadeghi, A. Rahimi-Kian, and C. Lucas, "Multiobjective invasive weed optimization: Application to analysis of Pareta improvement models in electricity markets", Applied Soft Computing 12, 2012, pp. 100 – 112.

[18] P. Pahlavani, M.R. Delavar, and A.U. Frank, "Using a modified invasive weed optimization for a personalized urban multi-criteria path optimization problem", International Journal of Applied Earth Observation and Geoinformation 18, 2012, pp. 313 – 328.

[19] R.M. Rizk-Allah, E.M. Zaki, and A.A. El-Sawy, "A novel hybrid ant colony optimization and firefly algorithm for solving constrained engineering design problems", Journal of Natural Sciences and Mathematics, Qassim University, 6 (1), Jan 2013, pp. 1 – 22.

[20] R.M. Rizk-Allah, E.M. Zaki, and A.A El-Sawy, "Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems", Applied Mathematics and Computation 224, 2013, pp. 473 – 483.

[21] S. Roy, S.M. Islam, S. Das, and S. Ghosh, "Multimodal optimization by artificial weed colonies enhanced with localized group search optimizers", Applied Soft Computing, 2013, pp. 27 – 46.

[22] A. Sadollah, H. Bahreininejadi, Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems", Applied Soft Computing 13, 2013, pp. 2592 – 2612.

[23] T. Sengupta, A. Chakraborti,, A. Konar, and A. K. Nagar, "An Intelligent Invasive Weed Optimization: a Q-learning Approach". In *Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Jan 2012.

[24] M. Tuba, I. Brajevic, and R. Jovanovic, "Hybrid seeker optimization algorithm for global optimization", International Journal Applied Mathematics & Information Sciences 7 (3), 2013, pp. 867 – 875.

[25] J. Y. Wu, "Solving Constrained Global Optimization Problems by Using Hybrid Evolutionary Computing and Artificial Life Approaches". Mathematical Problems in Engineering, Hindawi Publishing Corporation, Article ID 841410, 36 pages, 2012. doi:10.1155/2012/841410.

[26] X.S. Yang, "Firefly algorithms for multimodal optimization. Stochastic algorithm: Foundations and Applications", SAGA 2009, Lecture Notes in Computer Sciences, 5792, 2009, pp. 169 – 178.

[27] X.S. Yang, "Firefly, levy flights and global optimization", Research and Development in Intelligent Systems XXVI, 2010, pp. 209-218.

[28] X.S. Yang, and X. He, "Firefly algorithm: recent advances and applications", International Journal Swarm Intelligence 1 (1), 2013, pp. 36 – 50.

[29] X.S. Yang, "Nature-inspired metaheuristics algorithms", Luniver Press, United Kingdom, 2008.

[30] Z. Yin, M.I. Wen, and C. Ye, "Improved invasive weed optimization based on hybrid genetic algorithm", Journal of Computational Information Systems 8 (8), 2012, pp. 3437 – 3444.

[31] M. Younes, "A novel hybrid FFA-ACO algorithm for economic power dispatch", CEAI 15 (2), 2013, pp. 67 – 77.

[32] D. Z. Zaharias, C. Skeberis, T. D. Xenos, P. I. Lazaridis, and J. Cosmas, "Design of a novel antenna array beamformer using neural network trained by modified adaptive dispersion invasive weed optimization based data", IEEE Transactions on Broadcasting 59 (3), Sept. 2013, pp. 455 – 460.

[33] X. Zhang, Y. Niu, G. Cui, and Y. Wang, "A modified invasive weed optimization with crossover operation". In *Intelligent Control and Automation (WCICA), 2010 8th World Congress on* , IEEE, 2010, pp. 11-14.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[34] X. Zhang, J. Xu, G. Cui, G., Y. Wang, and Y. Niu, "Research on invasive weed optimization based on the cultural framework". In *Bio-Inspired Computing: Theories and Applications, 2008. BICTA 2008. 3rd International Conference on*, IEEE, 2008, pp. 129-134.
[35] Y. Zhou, Q. Luo, and H. Chen. "A novel differential evolution invasive weed optimization algorithm for solving nonlinear equations systems", Journal of Applied Mathematics, Hindawi Publishing Corporation, Article ID 757391, 18 pages, 2013, http://dx.doi.org/10.1155/2013/757391.

***Algorithm 1** Pseudo Code Of Classical Invasive Weed Optimization Algorithm*

*Input:*
Objective function of $f(x)$,
where $x = (x_1, \dots, x_d)^T$;
Pre-determined parameter, number of minimum seeds, $s_{min}$; number of maximum seeds, $s_{max}$; initial standard deviation, $\sigma_{iter}$; maximum population size $n$;
*Output:*
Begin
Generate initial population of weeds $x_i$, where $x_i (i = 1, \dots, n)$ by randomly initiating a population in the search space. Calculate every individual's fitness, $f(x_i)$,
Rank the initial weeds based on its fitness, $f(x_i)$,
  Calculate the number of seeds produced by each weed with Equation (1);
While ( t < maximum iteration)
{ t; current iteration}
  Update SD with Equation (2);
  Generate seeds over the search space;
  If the number of weeds and seeds > maximum population size, n
    Eliminate the plant with lower fitness;
  End if
  Calculate every individual fitness, $f(x_i)$,
  Rank the initial weeds based on their fitnesses, $f(x_i)$,
  Find the current best individual and its fitness;
End while;
Post process results and visualization;
End procedure;

***Algorithm 2** Pseudo Code Of Classical Firefly Algorithm*

*Input:*
Objective function of $f(x)$,
where $x = (x_1, \dots, x_d)^T$;
Pre-determined parameter, Attractiveness coefficient, $\beta_0$; Absorption coefficient, $\gamma$; Randomization coefficient, $\alpha$, variable boundary and population size $n$;
*Output: Output:*
Generate initial population of fireflies $x_i$, where $x_i (i = 1, \dots, n)$
Begin
Formulate the light intensity, $I(x_d)$;
While ( t < maximum iteration)
{ t; current iteration}
  For firefly $i$ to $n$; {all n fireflies};
    For firefly $j$ to $n$; {all n fireflies};
      Evaluate the distance between two fireflies $(x_i, x_j)$, $r$;
      Evaluate the attractiveness with distance via $e^{-\gamma r^2}$
      If ($I_j > I_i$), move firefly $i$ towards $j$, then;
        Evaluate new solutions, $x_{i+1}$ via Equation (3).
      End if;
    End for $j$;
  End for $i$;
  Update light intensity, $I(x_d)$ based on the update firefly location
  Rank the fireflies and find the current best;
End while;
Post process results and visualization;
End procedure;

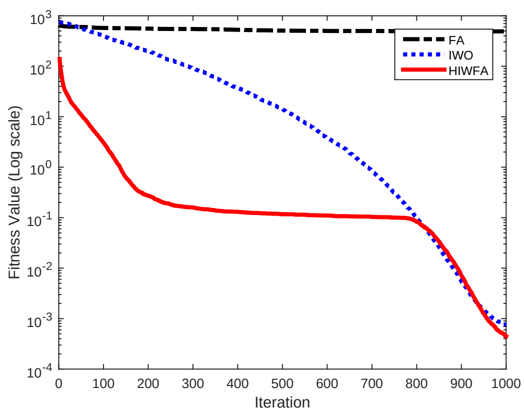
.

***ALGORITHM 3** PSEUDO CODE OF HYBRID INVASIVE WEED FIREFLY OPTIMIZATION ALGORITHM*

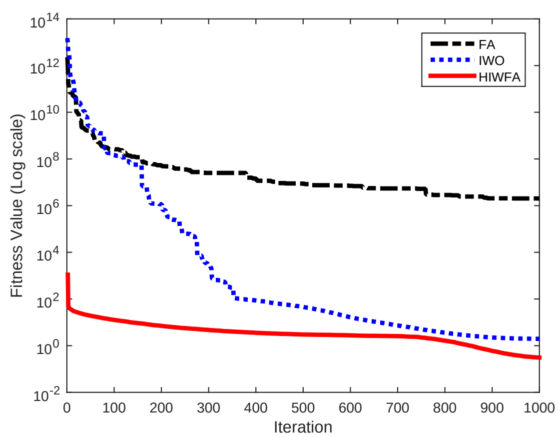
*Input:*
Objective function of $f(x)$,
where $x = (x_1, \dots, x_d)^T$;
Pre-determined parameter;
number of minimum seeds, $s_{min}$; number of maximum seeds, $s_{max}$; initial standard deviation, $\sigma_{iter}$; Attractiveness coefficient, $\beta_0$; Absorption coefficient, $\gamma$; Randomization coefficient, $\alpha$, variable boundary;
maximum population size $n$;
*Output:*
Begin
Generate initial population of weeds $x_i$, where $x_i (i = 1, \dots, n)$ by randomly initiating a population in the search space.
Calculate every individual's fitness, $f(x_i)$ and equally the value with the light intensity, $I(x_d)$;
Rank the initial weeds based on their fitnesses, $f(x_i)$,
Calculate the number of seeds produced by each weed with Equation (1);
While ( t < maximum iteration)
{ t; current iteration}
  Update SD with Equation (1) ;
  Generate seeds over the search space;
  If the number of weeds and seeds > maximum population size, n
    Eliminate the plant with lower fitness;
  End if
  Improve the weeds location using firefly localization
  For firefly $i$ to $n$; {all n weeds / fireflies};
    For firefly $j$ to $n$; {all n weeds / fireflies};
      Evaluate the distance between two fireflies $(x_i, x_j)$, $r$;
      Evaluate the attractiveness with distance via $e^{-\gamma r^2}$
      If ($I_j > I_i$), move firefly $i$ towards $j$, then;
        Evaluate new solutions, $x_{i+1}$ via Equation (2).
      End if;
    End for $j$;
  End for $i$;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

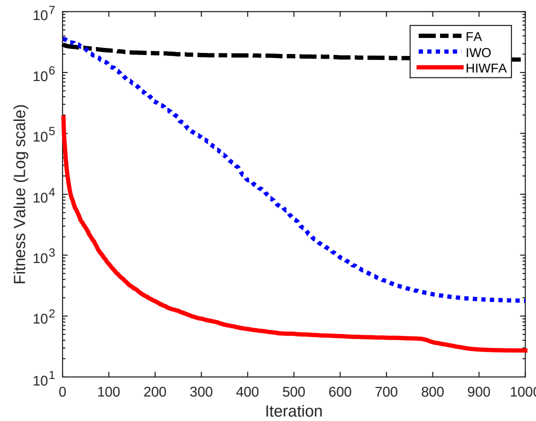
Update the weeds location by using boundary mechanism;
If $x_{i+1}$ exceeds its boundary, set $x_{i+1}$ to its boundary
End if
Calculate every individual's fitness, $f(x_i)$,
Rank the initial weeds based on their fitnesses, $f(x_i)$,
Find the current best individual and its fitness;
End while;
Post process results and visualization;
End procedure;



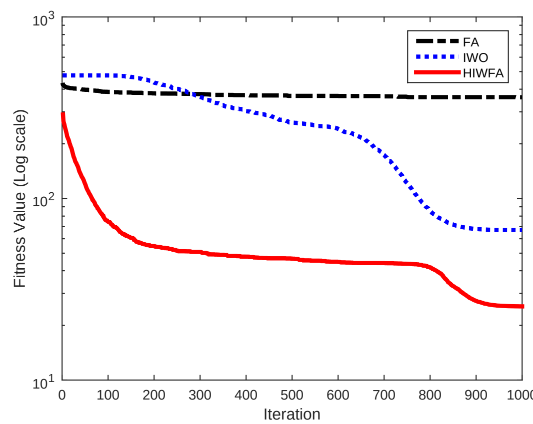

*(a) De Jong Function (f1)*



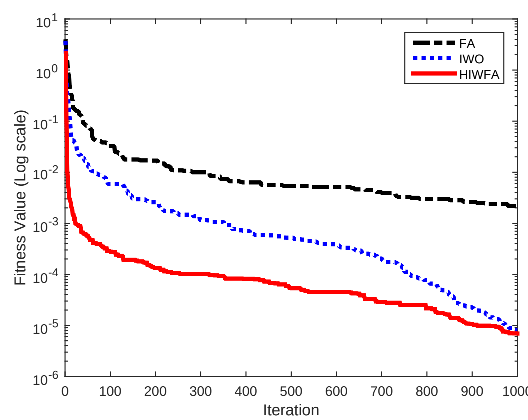

*(b) Schwefel's Problem 2.22 (f2)*




*(c) Rosenbrock Function (f3)*

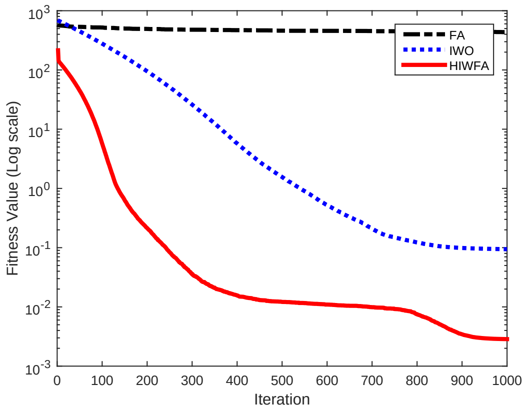



*(d) Rastrigin Function (f4)*




*(c) Ackley Function (f5)*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195




*(d) Griewank Function (f6)*

*Figure 1: Algorithm convergence in 30 dimensions' benchmark function tests.*

*TABLE 1: BENCHMARK FUNCTIONS USED IN THE TESTS*

| Name | Formulation | Variable range | f(min) | Unimodal / Multimodal |
|---|---|---|---|---|
| Sphere | $f_1(x) = \sum_{i=1}^{D} x_i^2$ | $[-10, 10]^D$ | 0 | U |
| Schwefel's Problem 2.22 | $f_2(x) = \sum_{i=0}^{D} \lvert x_i \rvert + \prod_{i=0}^{D} \lvert x_i \rvert$ | $[-10, 10]^D$ | 0 | U |
| Rosenbrock | $f_3(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | $[-10, 10]^D$ | 0 | U |
| Rastrigin | $f_4(x) = 10D + \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i)]$ | $[-5.12, 5.12]^D$ | 0 | U |
| Ackley | $f_5(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right) + 20 + e$ | $[-32, 32]^D$ | 0 | M |
| Griewank | $f_6(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \frac{x_i}{\sqrt{i}} + 1$ | $[-600, 600]^D$ | 0 | M |

*Table 2: Parameter values used by the algorithms in the tests*

| Algorithm | Population, n | Max Iteration | $\sigma_{initial}$ | $\sigma_{final}$ | $s_{max}$ | $s_{min}$ | β | γ | α |
|---|---|---|---|---|---|---|---|---|---|
| HIWFO | 40 | 1000 | 5 | 0.005 | 5 | 0 | 1.0 | 1.0 | 0.2 |
| FA | 40 | 1000 | - | - | - | - | 1.0 | 1.0 | 0.2 |
| IWO | 40 | 1000 | 5 | 0.005 | 5 | 0 | - | - | - |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 3: Performance comparison for unconstrained optimization problems*

| | | FA | | | IWO | | | HIWFA | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $f$ | D | Fitness | Std dev | Time | Fitness | Std dev | Time | Fitness | Std dev | Time |
| $f_1$ | 10 | 3.886E+01 | 8.337E+00 | 1.413E+01 | 4.344E-05 | 1.140E-05 | 1.338E+00 | **2.260E-05** | 3.097E-05 | 6.602E+00 |
| | 30 | 4.881E+02 | 3.033E+01 | 1.454E+01 | 7.554E-04 | 8.575E-05 | 1.516E+00 | **4.471E-04** | 5.660E-04 | 7.189E+00 |
| | 50 | 1.029E+03 | 6.028E+01 | 1.503E+01 | **2.736E-03** | 3.284E-04 | 1.694E+00 | 3.746E-03 | 1.931E-03 | 7.778E+00 |
| $f_2$ | 10 | 1.707E+01 | 2.314E+00 | 1.422E+01 | 1.739E-02 | 1.671E-03 | 1.512E+00 | **1.025E-02** | 1.028E-02 | 6.119E+00 |
| | 30 | 2.029E+06 | 3.962E+06 | 1.465E+01 | 1.967E+00 | 3.629E+00 | 1.705E+00 | **3.097E-01** | 2.344E-01 | 6.804E+00 |
| | 50 | 2.414E+16 | 9.065E+16 | 1.510E+01 | 5.157E+01 | 5.309E+01 | 1.998E+00 | **1.346E+00** | 9.190E-01 | 7.512E+00 |
| $f_3$ | 10 | 2.406E+04 | 1.042E+04 | 1.413E+01 | **4.588E+00** | 5.309E+01 | 1.708E+00 | 5.555E+00 | 1.335E+00 | 6.775E+00 |
| | 30 | 1.632E+06 | 2.311E+05 | 1.472E+01 | 1.781E+02 | 3.596E+02 | 1.725E+00 | **2.722E+01** | 2.274E+00 | 7.419E+00 |
| | 50 | 4.658E+06 | 3.654E+05 | 1.519E+01 | 1.766E+02 | 4.727E+02 | 1.994E+00 | **4.936E+01** | 6.706E+00 | 8.082E+00 |
| $f_4$ | 10 | 6.591E+01 | 8.850E+00 | 1.473E+01 | 1.022E+01 | 3.355E+00 | 1.374E+00 | **4.423E+00** | 1.933E+00 | 6.512E+00 |
| | 30 | 3.609E+02 | 1.637E+01 | 1.509E+01 | 6.697E+01 | 1.655E+01 | 1.618E+00 | **2.545E+01** | 7.418E+00 | 7.111E+00 |
| | 50 | 6.899E+02 | 1.790E+01 | 1.489E+01 | 1.590E+02 | 3.261E+01 | 1.835E+00 | **5.631E+01** | 1.335E+01 | 7.649E+00 |
| $f_5$ | 10 | 2.962E-03 | 1.996E-03 | 1.325E+01 | 9.096E-06 | 7.765E-06 | 1.520E+00 | **4.607E-06** | 3.727E-06 | 6.455E+00 |
| | 30 | 2.177E-03 | 1.925E-03 | 1.375E+01 | 7.956E-06 | 5.771E-06 | 1.790E+00 | **6.938E-06** | 6.465E-06 | 7.124E+00 |
| | 50 | 1.941E-03 | 1.871E-03 | 1.442E+01 | 9.360E-06 | 8.798E-06 | 1.972E+00 | **4.599E-06** | 3.811E-06 | 7.672E+00 |
| $f_6$ | 10 | 3.703E+01 | 6.220E+00 | 1.392E+01 | 6.580E-02 | 2.733E-02 | 1.486E+00 | **3.175E-02** | 3.395E-02 | 6.584E+00 |
| | 30 | 4.351E+02 | 3.347E+01 | 1.447E+01 | 9.409E-02 | 2.800E-01 | 1.738E+00 | **2.859E-03** | 6.081E-03 | 7.193E+00 |
| | 50 | 9.369E+02 | 4.386E+01 | 1.482E+01 | 4.120E+01 | 1.903E+01 | 1.899E+00 | **2.020E-03** | 4.103E-03 | 7.882E+00 |

*TABLE 4: STATISTICAL RESULTS OBTAINED USING BENCHMARK FUNCTIONS.*

| | | FA | | | | IWO | | | | HIWFA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *f* | D | Best | Worst | Median | Std dev | Best | Worst | Median | Std dev | Best | Worst | Median | Std dev |
| $f_1$ | 10 | 2.294E+01 | 5.538E+01 | 3.685E+01 | 8.337E+00 | 1.439E-05 | **6.059E-05** | 4.413E-05 | 1.140E-05 | **4.747E-07** | 9.345E-05 | 1.190E-06 | 3.097E-05 |
| | 30 | 4.077E+02 | 5.391E+02 | 4.911E+02 | 3.033E+01 | 6.056E-04 | **9.388E-04** | 7.621E-04 | 8.575E-05 | **1.654E-05** | 1.607E-03 | 6.656E-05 | 5.660E-04 |
| | 50 | 8.973E+02 | 1.146E+03 | 1.036E+03 | 6.028E+01 | 2.099E-03 | **3.616E-03** | 2.720E-03 | 3.284E-04 | **8.077E-04** | 7.339E-03 | 4.008E-03 | 1.931E-03 |
| $f_2$ | 10 | 1.197E+01 | 2.063E+01 | 1.717E+01 | 2.314E+00 | 1.353E-02 | **2.168E-02** | 1.745E-02 | 1.671E-03 | **1.497E-03** | 2.861E-02 | 2.909E-03 | 1.028E-02 |
| | 30 | 1.854E+04 | 1.881E+07 | 3.771E+05 | 3.962E+06 | 1.102E-01 | 1.371E+01 | 1.370E-01 | 3.629E+00 | **1.027E-01** | **1.143E+00** | 2.268E-01 | 2.344E-01 |
| | 50 | 1.133E+12 | 4.959E+17 | 2.482E+15 | 9.065E+16 | **2.933E-01** | 1.847E+02 | 2.531E+01 | 5.309E+01 | 5.041E-01 | **4.587E+00** | 9.585E-01 | 9.190E-01 |
| $f_3$ | 10 | 7.687E+03 | 5.061E+04 | 2.364E+04 | 1.042E+04 | **2.933E-01** | 1.847E+02 | 2.531E+01 | 5.309E+01 | 2.960E+00 | **8.678E+00** | 5.479E+00 | 1.335E+00 |
| | 30 | 1.226E+06 | 2.040E+06 | 1.638E+06 | 2.311E+05 | 2.574E+01 | 1.701E+03 | 2.928E+01 | 3.596E+02 | **1.880E+01** | **2.951E+01** | 2.772E+01 | 2.274E+00 |
| | 50 | 3.976E+06 | 5.347E+06 | 4.617E+06 | 3.654E+05 | 4.650E+01 | 2.623E+03 | 4.914E+01 | 4.727E+02 | **4.434E+01** | **8.408E+01** | 4.898E+01 | 6.706E+00 |
| $f_4$ | 10 | 4.702E+01 | 7.963E+01 | 6.831E+01 | 8.850E+00 | 4.987E+00 | 1.891E+01 | 9.956E+00 | 3.355E+00 | **1.997E+00** | **8.963E+00** | 3.994E+00 | 1.933E+00 |
| | 30 | 3.133E+02 | 3.845E+02 | 3.646E+02 | 1.637E+01 | 3.494E+01 | 1.126E+02 | 6.382E+01 | 1.655E+01 | **9.146E+00** | **3.907E+01** | 2.615E+01 | 7.418E+00 |
| | 50 | 6.500E+02 | 7.241E+02 | 6.945E+02 | 1.790E+01 | 9.410E+01 | 2.255E+02 | 1.592E+02 | 3.261E+01 | **3.682E+01** | **9.756E+01** | 5.390E+01 | 1.335E+01 |
| $f_5$ | 10 | 1.109E-04 | 8.833E-03 | 2.846E-03 | 1.996E-03 | 1.076E-06 | 2.982E-05 | 6.411E-06 | 7.765E-06 | **8.166E-08** | **1.654E-05** | 3.713E-06 | 3.727E-06 |
| | 30 | 1.452E-04 | 8.049E-03 | 1.682E-03 | 1.925E-03 | 1.320E-06 | 2.979E-05 | 6.852E-06 | 5.771E-06 | **1.378E-07** | **2.478E-05** | 3.959E-06 | 6.465E-06 |
| | 50 | 1.013E-04 | 9.153E-03 | 1.483E-03 | 1.871E-03 | 1.515E-06 | 4.049E-05 | 5.658E-06 | 8.798E-06 | **5.335E-08** | **1.862E-05** | 3.988E-06 | 3.811E-06 |
| $f_6$ | 10 | 2.201E+01 | 4.618E+01 | 3.792E+01 | 6.220E+00 | 1.478E-02 | 1.548E-01 | 6.272E-02 | 2.733E-02 | **3.878E-08** | **9.604E-02** | 1.601E-02 | 3.395E-02 |
| | 30 | 3.654E+02 | 4.867E+02 | 4.383E+02 | 3.347E+01 | 7.430E-03 | 1.219E+00 | 1.482E-02 | 2.800E-01 | **8.149E-06** | **2.218E-02** | 8.524E-05 | 6.081E-03 |
| | 50 | 8.448E+02 | 9.962E+02 | 9.376E+02 | 4.386E+01 | 1.347E+01 | 8.203E+01 | 4.223E+01 | 1.903E+01 | **2.304E-04** | **1.566E-02** | 5.227E-04 | 4.103E-03 |

*Table 5: The best solution obtained for welded beam design problem.*

| | Optimal design variables | | | | Min f(x) | | | |
|---|---|---|---|---|---|---|---|---|
| Methods | $x_1(h)$ | $x_2(l)$ | $x_3(t)$ | $x_4(b)$ | Best | Mean | Worst | Std Dev |
| CPSO | 0.20237 | 3.54421 | 9.04821 | 0.20572 | 1.72802 | 1.74883 | 1.78831 | $1.30 \times 10^{-2}$ |
| PSO-DE | 0.20573 | 3.47049 | 9.03662 | 0.20573 | 1.72485 | 1.72485 | 1.72485 | $6.70 \times 10^{-16}$ |
| CSS-PSO | 0.20730 | 3.43570 | 9.04193 | 0.20571 | 1.72338 | 1.74345 | 1.76257 | $7.36 \times 10^{-3}$ |
| HGSO | 0.20573 | 3.47049 | 9.03662 | 0.20573 | 1.72485 | 1.72485 | **1.72485** | $3.60 \times 10^{-12}$ |
| FA | 0.20150 | 3.56200 | 9.04140 | 0.20570 | 1.73121 | 1.87866 | 2.34558 | 0.26780 |
| HIWFO | 0.24748 | 2.77145 | 9.10994 | 0.20670 | **1.71520** | **1.72574** | 1.73841 | $5.01 \times 10^{-3}$ |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 6: The best solution obtained for the tension / compression spring design problem.*

| | Optimal design variables | | | Min f(x) | | | |
|---|---|---|---|---|---|---|---|
| Methods | $x_1(d)$ | $x_2(D)$ | $x_3(N)$ | Best | Mean | Worst | Std Dev |
| CPSO | 0.05173 | 0.35764 | 11.24454 | 0.01267 | 0.01273 | 0.01292 | $5.20 \times 10^{-5}$ |
| PSO-DE | 0.05190 | 0.35671 | 11.28932 | 0.01267 | 0.01267 | **0.01267** | $1.20 \times 10^{-8}$ |
| CSS-PSO | 0.05143 | 0.35106 | 11.60979 | 0.01264 | 0.01275 | 0.01301 | $3.95 \times 10^{-5}$ |
| HGSO | 0.051690 | 0.35672 | 11.28932 | 0.01267 | 0.01267 | 0.01267 | $4.35 \times 10^{-15}$ |
| FA | NA | NA | NA | NA | NA | NA | NA |
| HIWFO | 0.050000 | 0.31916 | 13.76057 | **0.01264** | **0.01265** | 0.01268 | $1.25 \times 10^{-5}$ |

*Table 7: The best solution obtained for pressure vessel design problem.*

| | Optimal design variables | | | | Min f(x) | | | |
|---|---|---|---|---|---|---|---|---|
| Methods | $x_1(Ts)$ | $x_2(Th)$ | $x_3(R)$ | $x_4(L)$ | Best | Mean | Worst | Std Dev |
| CPSO | 0.81250 | 0.43750 | 42.09808 | 176.6405 | 6059.745 | 6850.004 | 7332.879 | 426.000 |
| PSO-DE | 0.81250 | 0.43750 | 42.09844 | 176.6366 | 6059.714 | 6059.714 | 6059.714 | $1.00 \times 10^{-10}$ |
| CSS-PSO | 0.81250 | 0.43750 | 42.14262 | 176.0904 | 6059.684 | 6068.753 | 6103.882 | 13.124 |
| HGSO | 0.81250 | 0.43750 | 42.09844 | 176.6366 | 6059.714 | 6059.714 | 6059.714 | $9.25 \times 10^{-13}$ |
| FA | 0.75000 | 0.37500 | 38.86010 | 221.3655 | 5850.383 | 5937.338 | 6258.968 | 164.547 |
| HIWFO | 0.78365 | 0.38712 | 40.57787 | 197.8209 | 5927.636 | 6099.018 | 6224.648 | 83.828 |

*Table 8: The best solution obtained for speed reducer design problem.*

| | Optimal design variables | | | | | | | Min f(x) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | Best | Mean | Worst | Std Dev |
| PSO-DE | 3.50 | 0.70 | 17.0 | 7.30 | 7.80 | 3.35 | 5.29 | 2996.348 | 2996.348 | 2996.348 | $6.4 \times 10^{-6}$ |
| HGSO | 3.50 | 0.70 | 17.0 | 7.30 | 7.72 | 3.35 | 5.29 | 2994.471 | 2994.471 | 2994.471 | $1.44 \times 10^{-10}$ |
| HIWFO | 3.28 | 0.70 | 17.0 | 7.30 | 7.54 | 3.30 | 5.17 | 2979.524 | 2990.461 | 3005.640 | 6.415 |

***Appendix A1. Welded beam design problem***

*Cost function*

$$\min f(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$$

*Constraint functions*

$$g_1(x) = \tau(\{x\}) - \tau_{max} \le 0$$
$$g_2(x) = \sigma(\{x\}) - \sigma_{max} \le 0$$
$$g_3(\{x\}) = x_1 - x_4 \le 0$$
$$g_4(x) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0$$
$$g_5(x) = 0.125 - x_1 \le 0$$
$$g_6(x) = \delta(\{x\}) - \delta_{max} \le 0$$
$$g_7(x) = P - P_c(\{x\}) \le 0$$

*where*

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2}x_1 x_2}, \tau'' = \frac{MR}{J}, M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}, J = 2\left\{\sqrt{2}x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(x) = \frac{6PL}{x_4 x_3^2}, \delta(x) = \frac{4PL^2}{E x_3^3 x_4}, P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000\ lb, L = 14\ in, E = 30 \times 10^6 psi, G = 12 \times 10^6 psi$$

*Ranges of variables*

$$0.1 \le x_1, x_4 \le 2, 0.1 \le x_2, x_3 \le 10$$

***Appendix A2. Tension / compression string design problem***

*Cost function*

$$\min f(x) = (x_3 + 2) x_2 x_1^2$$

*Constraint functions*

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0$$
$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0$$
$$g_4(x) = \frac{(x_1 + x_2)}{1.5} - 1 \le 0$$

*Ranges of variables*
$0.05 \le x_1 \le 2, 0.25 \le x_2 \le 1.3, 2 \le x_3 \le 15$

***Appendix A3. Pressure vessel design problem***
*Cost function*
$$min\, f(x) = 0.6224x_1x_2 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$
*Constraint functions*
$$g_1(x) = -x_1 + 0.0193x_3 \le 0$$
$$g_2(x) = -x_2 + 0.00954x_3 \le 0$$
$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$
$$g_4(x) = x_4 - 240 \le 0$$

*Ranges of variables*
$0 \le x_1, x_2 \le 99, 10 \le x_3, x_4 \le 200$

***Appendix A4. Speed reducer design problem***
*Cost function*
$$min\, f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$
*Constraint functions*
$$g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \le 0$$
$$g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0$$
$$g_3(x) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0$$
$$g_4(x) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0$$
$$g_5(x) = \frac{[(745(^{x_4}/_{x_2x_3}))^2 + 16.9 \times 10^6]^{1/2}}{110x_6^3} - 1 \le 0$$
$$g_6(x) = \frac{\left[(745(^{x_5}/_{x_2x_3}))^2 + 157.5 \times 10^6\right]^{1/2}}{85x_7^3} - 1 \le 0$$
$$g_7(x) = \frac{x_2x_3}{40} - 1 \le 0$$
$$g_8(x) = \frac{5x_2}{x_1} - 1 \le 0$$
$$g_9(x) = \frac{x_1}{12x_2} - 1 \le 0$$
$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0$$
$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0$$

*Ranges of variables*
$2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28, 7.3 \le x_4 \le 8.3, 7.3 \le x_5 \le 8.3, 2.9 \le x_6 \le 3.9, 5.5 \le x_7 \le 5.5$