ASIAN JOURNAL OF CIVIL ENGINEERING (BHRC) VOL. 16, NO. 2 (2015)
PAGES 249-268

DYNAMIC FUZZY-MEMBERSHIP OPTIMIZATION: AN ENHANCED META-HEURISTIC SEARCH

M. Shahrouzi[1] and A. Kaveh[2*]
[1]Department of Engineering, Kharazmi University, Tehran, Iran
[2]Department of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran 16, Iran

Received: 29 April 2014; **Accepted:** 20 October 2014

ABSTRACT

A new hybrid framework is proposed for optimization of orderable search spaces. It is based on fuzzy-membership of design variable alternatives to the optimal solution. The method has dynamic behavior since the membership values are assigned as any solution candidates arises during the search and they are summed based on their overlaps in the alternatives scope. A trail matrix is also utilized to indirectly share values of the fuzzy memberships further ranked regarding closeness of an individual to the optimal solution both in the objective function and in the design space scope. The method takes benefit of different random, vector-sum and probability-based walks to move new solutions toward the global optimum. Utilizing a generalized cooling procedure, the related thresholds are tuned to choose between different walk types in searching the design space and also a search refinement strategy is developed. Two variants of such a framework is then proposed and compared with each other in addition to some well-known procedures including genetic algorithm and particle swarm optimization. Test results with some treated problems, reveals the superior performance of the proposed algorithm and its special feature in adaptive tuning the diversity index during the search.

Keywords: Meta-heuristic algorithms; information share; fuzzy logic; structural optimization.

1. INTRODUCTION

Application of meta-heuristic algorithms in various engineering field has experienced a considerable growth in the last three decades. It is due to simplicity of their programming and generality in solving gradient-free and discrete optimization problems. However, there are some challenges in their reliability to obtain global optimum result and the required

[*]E-mail address of the corresponding author: alikaveh@iust.ac.ir (A. Kaveh)

computational effort for such a purpose. Several meta-heuristic algorithms have recently been developed by investigators to solve different problems [1-6].

The metaheuristic algorithms can be categorized as bio-inspired, nature-based and other heuristics in the original form; however, hybridization techniques are also used to combine some of their features for an efficient search in a particular problem. A main challenge to develop a proper search method is how to provide proper balance between diversification/intensification or global/local search features of that algorithm. It is desired to make early access of the search agents to all regions in the design space, meanwhile not to miss and overpass true optima in narrow valleys of it at the end. Because of different constraints and design spaces in different problems, the method is desired to be capable of changing its diversity balance adaptively case by case.

This paper presents a novel hybrid framework aiming to provide required features in searching discrete orderable problems. It takes benefit of cooling strategy to tune the balance between probability thresholds which are used to switch between explorative random walks and exploitative movements. Meanwhile the concept of fuzzy-membership functions is extended to the available alternatives space combined with an indirect share strategy using artificial pheromone trail and ranking based on closeness of the newcomer designs to the best-so-far one. First the concepts of this method, called Dynamic Fuzzy-membership Optimization (DFO), are introduced followed by its general formulation. Then, two variants out of the general framework is developed and further tested with some benchmark problems. Finally a comparison between the results of the employed algorithms and well-known genetic and particle swarm optimization is performed. The performance and behavioral features of the proposed algorithm variants are then declared and discussed.

2. FUZZY MEMBERSHIP IN THE VARIABLES' SCOPE

The search space of many optimization problems is defined using a fixed-length design vector. It is called a signal in BA [7], a chromosome in GA [8], a trip in ACO [9], a position in PSO [10]. The entire search space can thus be constructed by choosing possible values for each component of such a design vector and generating their combinations within the fixed-length vector. Each combination has its individual fitness value. However in some problems, a preference order may be distinguished among available values for each design variable with indirect effect on the resulting value of the objective or fitness function. They are here-in-after called orderable search spaces.

The design space of a truss-like sizing problem is an example of such spaces in which increasing area of each element will result in increasing the total structural weight. In such a case, one can arrange the section areas assignable to a member in ascending (or descending) order. The optimal design vector, however, depends on the problem constraints rather than only to the minimal weight.

Every meta-heuristic algorithm applies its way of generating newcomer design-candidate out of the current population of design vectors. Once a newcomer is generated and its fitness is evaluated, the chosen value of its variables in the design vector can be assigned a fuzzy-membership to the optimal solution. That means, a value in the allowable range for every variable is not strictly good or bad, but it by some degree (the fuzzy membership) belongs to the best solution. It will further be shown how such a strategy is applied to accelerate the optimization in the ordered search spaces.

3. INDIRECT INFORMATION SHARE

Choosing the fuzzy membership function is a real challenge when the optimum is not yet found. Some methods [11] use a triangular function to highlight position of the value belonging to the best-so-far solution during optimization. Here, we use an ant-based fuzzy membership assignment. Let's first review some essential features of ant colony algorithms prior to employ them in our work.

Ant Colony Optimization (ACO), is a common term for a set of algorithms which mimic the following behavior of natural ants in finding their routes to the goal [9]. Each ant likely chooses a route in which more pheromone is deposited by previous ants that have already passed it. Since the pheromone is continuously evaporated, the more pheromone in a route means more deposit in unit time; that occurs for shorter routes.

Some amount of the deposited pheromone is decreased during evaporation while new deposits usually take place by other ants. In this way the ants (as search agents) indirectly share information about shorter routes to their goal via pheromone trail. Hence, a short-term memory of best routes is constructed in every loop of the search.

The idea is employed here to assign pheromone, τ, to the corresponding values in the matrix of alternative solutions, Alt. Let each variable x_i in a design vector $\underline{X}$ can choose m values between x_i^{LB} and x_i^{UB} when there is n variables in $\underline{X}$. In this case, Alt is a m by n matrix. Once an optimization loop is completed and the best-so-far $\underline{X}^{best}$ is updated, τ for any component of Alt is updated using the following relation:

$$\tau_{ij} = (1-\alpha)\tau_{ij} + \Delta\tau_{ij} \tag{1}$$

$$\Delta\tau_{ij} = u(Alt_{ij}, x_j^i, \sigma) \times g^i \tag{2}$$

$$g^i = \frac{1}{(\left\| \underline{X}^i - \underline{X}^{best} \right\| + \varepsilon)} \times \frac{Fitness^i - Fitness^{Min}}{Fitness^{Max} - Fitness^{Min}} \tag{3}$$

Where α is the pheromone evaporation ratio and u denotes a fuzzy-membership function. According to this relation, fuzzy-membership is centered at $\underline{X}_i$ and amplified with its normalized fitness and also closeness of $\underline{X}_i$ to the $\underline{X}^{best}$. A tiny positive value ε is added to the norm of such a distance to prevent singularity at $\underline{X}^{best}$. The pheromone packet $\Delta\tau_{ij}$ is distributed in range of indices $k-\sigma$ to $k-\sigma$, where $Alt_{kj} = x_j^i$ for any jth variable in the design vector. It is evaporated in this range by $1-\alpha$ for $\underline{X}^{best}$. The matrix of remained pheromone values after deposit/evaporation is called $AF = \left[\tau_{ij}\right]$ in the present work.

Triangular function is a common choice for u, however, a normal distribution is offered here-in-after, Fig. 1. Therefore it is taken $N(Alt_{ij}, x_j^{best}, \sigma)$; a normal distribution centered at x_j^{best} with the standard deviation σ as employed for pheromone deposition in continuous problems:

$$N(x,a,\sigma) = e^{-\frac{(x-a)^2}{2\sigma^2}} \tag{4}$$

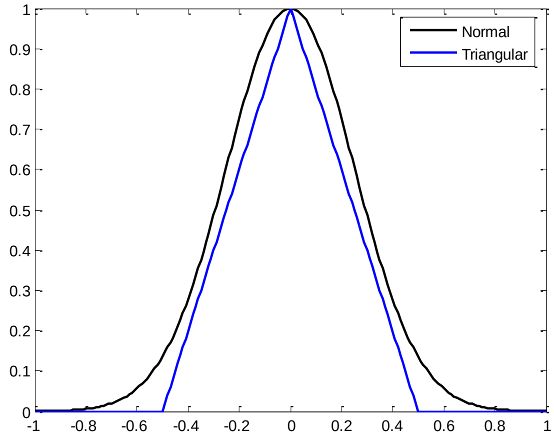

Figure 1. Sample fuzzy-membership functions: normal vs. triangular distribution

4. SEARCH REFINING STRATEGY

Bandwidth perturbation is a common search tool in meta-heuristic algorithms. It is performed in BA local search phase, HS pitch adjustment [12] and mutation in variable band GA [13].

In this paper, similar idea is implemented indirectly by controlling the standard deviation, σ, in the normal-distributed fuzzy membership. The parameter definition is extended to $\sigma_j(t)$ as:

$$\sigma_j(t) = \beta(x_j^{UB} - x_j^{LB})\, f_{cooling}(t) \tag{5}$$

The constant β determines the maximum band with respect to the allowable range of the j[th] variable. This band is, however, decreased according to the envelope function, $f_{cooling}(t)$, which starts from unity. Consequently, the effective range of pheromone deposition is decreased as the search iteration, t, gets closer to its maximum, t_{max}.

The role of $f_{cooling}(t)$ in the present algorithm is somehow like *cooling* procedure in the well-known Simulated Annealing Algorithm [14], since both are used to decrease the ratio of exploration to local search as the search progresses to its end in order not to loose the solutions found at later iterations.

5. PROBABILITY-BASED WALKS

Each heuristic algorithm has its own method of generating new-comer solutions based on search experience in the current population. The present work not only takes benefit of directional walks [15] but also uses the following ant-based probability threshold for this purpose:

$$q_{kj} = \frac{\tau_{kj}}{\sum_j \tau_{kj}} \tag{6}$$

It is used as probability of choosing the Alt_{kj} in the alternatives matrix for the j[th] position in any current member of population; x_j^i via Roulette-Wheel selection role. Such a procedure constitutes an intensification operator in the present algorithm. The other one is to change the $\underline{X}_i$ location in the vector search space toward the current global best; $\underline{X}^{best}$ adding a portion of their distance to the former vector:

$$\underline{V}^{i,t} = r_1 c_i \underline{V}^{i,t-1} + r_2 c_s (\underline{X}^{best} - \underline{X}^{i,t}) \tag{7}$$

$$\underline{X}^{i,t} = \underline{V}^{i,t} + \underline{X}^{i,t-1} \tag{8}$$

Where c_i and c_s stand for inertial and social constants, respectively and r_1, r_2 are randomly generated number between 0 to 1. The algorithm selection criterion is thus as follows:

$$x_j^{i,t} = \begin{cases} x_j^{LB} + r(x_j^{UB} - x_j^{LB}) & if \quad r < p_1 \\ Alt_{kj} \; with\ probability \; q_{kj} & if \quad r < p_2 \wedge r > p_1 \\ x_j^{i,t} + V_j^{i,t-1} & otherwise \end{cases} \tag{9}$$

While r is a random number in the range 0 to 1. The balance between different selection roles are tuned using probability thresholds p_1 and p_2 that are more discussed in the next section.

6. ALGORITHM PARAMETERS

Population size; $N_{Population}$ and the maximum number of iterations; $t_{\max}$ are common parameters for this algorithm. The others are α, β, c_i, c_s probability thresholds; p_1, p_2 and shape of the function $f_{cooling}(t)$. The choice of these extra parameters determines which variant of the algorithm to be further used in practice. Two variants are given in Table 1:

Table 1: Control parameters for two variants of the DFO algorithm

Method	α	β	c	c_s	u	g^i	$f_{cooling}(t)$	p_1	p_2
DFOv1	0.1	0.25	1	2	Normal Distribution	$\frac{1}{(\left\Vert \underline{X}^i-\underline{X}^{best}\right\Vert+\varepsilon)}\times\frac{Fitness^i-Fitness^{Min}}{Fitness^{Max}-Fitness^{Min}}$	$1-(\frac{t-1}{t_{\max}-1})$	$f_{cooling}(t$	1- $f_{cooling}(t)$
DFOv2	1	0.25	0	0	Triangular Distribution	$\frac{Fitness^i-Fitness^{Min}}{Fitness^{Max}-Fitness^{Min}}$	$1-(\frac{t-1}{t_{\max}-1})$	0	1- $f_{cooling}(t)$

Performance of DFO is then compared with a number of well-known meta-heuristics using the same $N_{Population}=25$ and $t_{\max}=200$ treating some test problems in this study as follows. Each design variable range is also subdivided into 200 equal spaces for DFO implementation.

For the sake of true comparison, the randomly initiated population in the first method is saved and identically used for the second so that the resulting convergence curves have the same starting point. Thus, a fitness improvement factor; FI, is defined for each run of an optimization algorithm as:

$$FI=\frac{BestSoFarFitness(\mathrm{t}_{\max})-BestSoFar\ Fitness(\mathrm{t}_1)}{BestSoFar\ Fitness(\mathrm{t}_1)} \tag{10}$$

In addition, a diversity index; DI, is defined for the population in every iteration as:

$$DI=mean(\frac{SD_j}{x_j^{UB}-x_j^{LB}}) \tag{11}$$

7. ILLESTRATIVE EXAMPLES

Test problem 1

As the first example, a pure convex search space is considered by De Jong's first function to be minimized, Fig. 2. There are no local optima but one global optimum 0 at $\underline{X}=\underline{0}$ for this minimization problem:

$$f(\underline{X})=\sum_{i=1}^{N}x_i^2 \tag{12}$$

Meanwhile each design variable for *N=2* dimensions is in the range $-5.12\ to\,5.12$ as reported in literature [16].

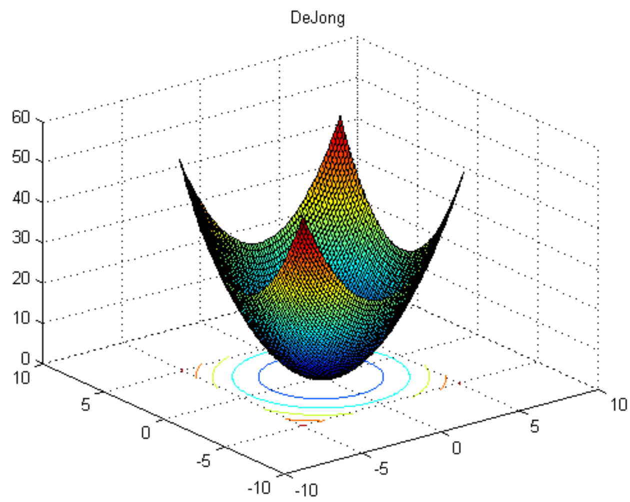

Figure 2. DeJong's test function

Test problem 2

In this problem, it is aimed to minimize the Griewangk's function with the following relation in *N=2* dimensions, Fig. 3:

$$f(\underline{X}) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{j=1}^{N}\cos\left(\frac{x_j}{\sqrt{j}}\right) + 1 \tag{13}$$

The design vector is limited to the range $[-100,100]^2$. The function has several local optima in detail scale with one global optimum at $f(\underline{0}) = 0$.

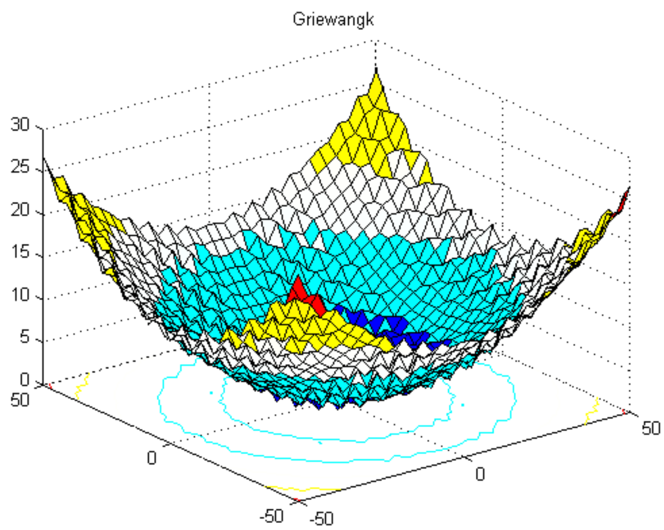

Figure 3. Griewangk's test function

Test problem 3

Ackley's function is selected here to test a multimodal search space with one highlighted global optimum and several neighboring local optima, Fig. 4. With the following relation in *N=2* dimensions, minimum of $f(\underline{X})$ is to be searched when $\underline{X}$ is limited to [-32.768, 32.768]^2. The global minimum is located at (0,0).

$$f(\underline{X}) = -20\exp(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} - \exp(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi . x_i)) + 20 + e \tag{14}$$

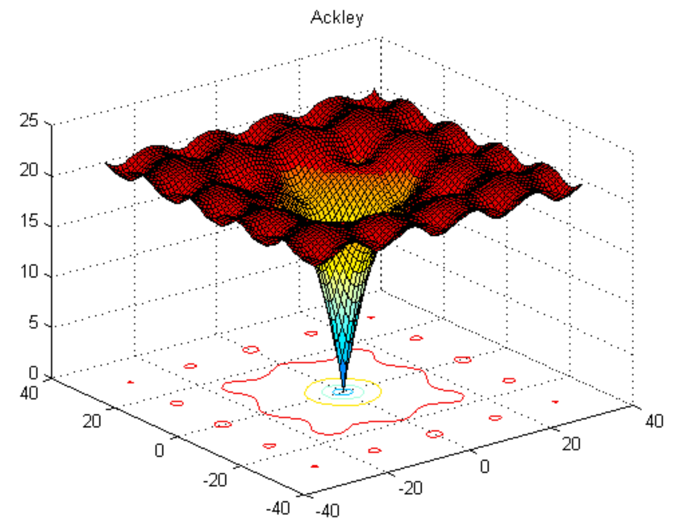

Figure 4. Ackley's test function

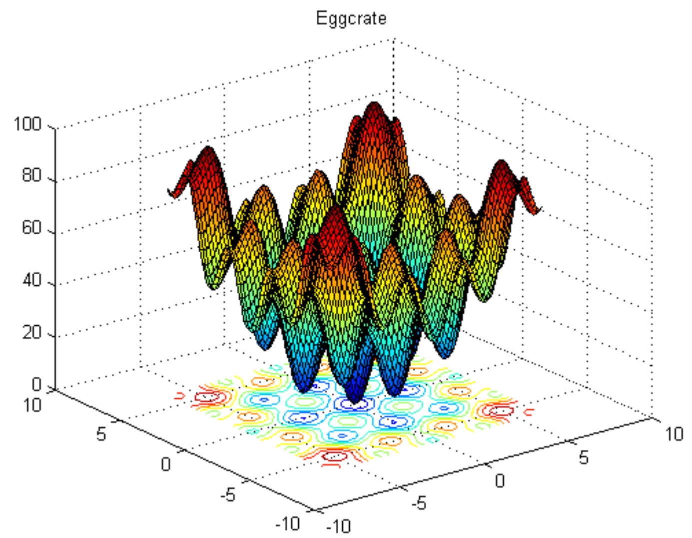

Figure 5. The Eggcrate test function

Test problem 4

In order to test the algorithms in severe hill-climbing, the Eggcrate function is selected here with its global minimum at (0,0) and several local optima with close objective function, Fig. 5. The design variable is limited to vary within range $\pm 2\pi$ in *N=2* dimensions. Eggcrate function is given by the following relation:

$$f(\underline{X}) = \sum_{i=1}^{N} x_i^2 + 25\sum_{i=1}^{N} \sin^2(x_i) \tag{15}$$

Test problem 5: A 3-Bar Truss Design

The 3-bar truss design is considered here as its objective function can be plotted in Fig. 6 due to having only 2 area section variables $\underline{X} = (x_1, x_2)$. This was introduced by Nowcki-1974 [17] as a base structural sizing problem for weight minimization. Using $l = 100cm$, $\sigma = 2kN / cm^2$ and vertical load $P = 2kN$ the problem is defined as:

$$\begin{aligned}
&\textit{Minimize } f(\underline{X}) = l \times (2\sqrt{2x_1} + x_2)\\
&\quad \textit{Subject to}\\
&\quad g_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \le 0\\
&\quad g_2 = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \le 0\\
&\quad g_3 = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \le 0\\
&\quad 0 \le x_1 \le 1\\
&\quad 0 \le x_2 \le 1
\end{aligned} \tag{16}$$

Fig. 7 shows the search space of a narrowly constrained problem.

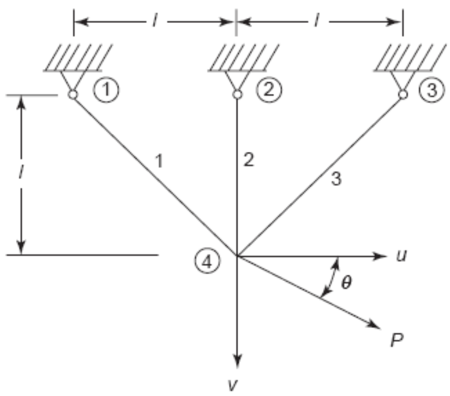

Figure 6. Schematic of the 3-bar design problem [18]

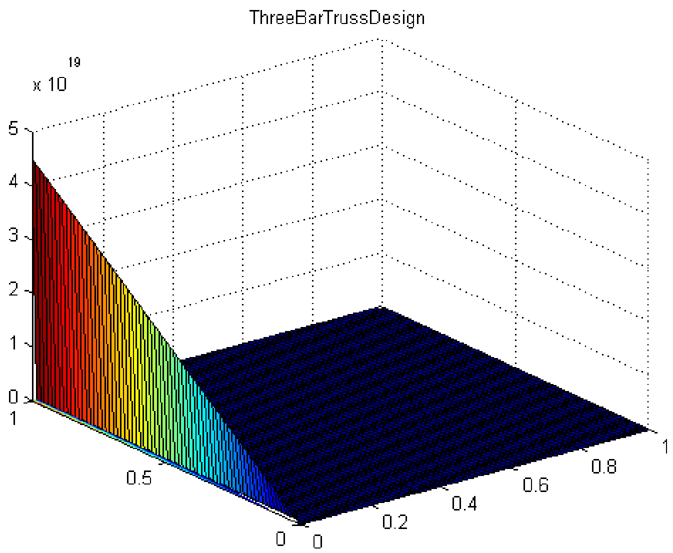

Figure 7. Search space of 3-bar design problem

Test problem 6: Design of a coil spring

It is another sample of constrained structural problem for minimizing the coil spring's weight subject to behavioral constraints of on shear stress, surge frequency and deflection, Fig. 8. However, additional variables' bounds constraints; $x_i^{LB} \le x_i \le x_i^{UB}$, should also be satisfied. The design variables x_1 to x_3 include the mean coil diameter *D*, the wire diameter *d* and the number of active coils *N* as shown in Fig. 8, respectively.

$$\begin{aligned}
&\textit{Minimize} \qquad f(X) = (x_3+2)x_2x_1^2 \\
&\textit{Subject to}: \\
&g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0, \\
&g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \le 0, \\
&g_3(X) = 1 - \frac{140.45 x_1}{x_2^3 x_3} \le 0, \\
&g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \le 0
\end{aligned}$$

$$\begin{aligned}
X^{LB} &= [x_1^{LB}, x_2^{LB}, x_3^{LB}] = [0.05, 0.25, 2.00], \\
X^{UB} &= [x_1^{UB}, x_2^{UB}, x_3^{UB}] = [2.00, 1.30, 1.50]
\end{aligned} \tag{17}$$

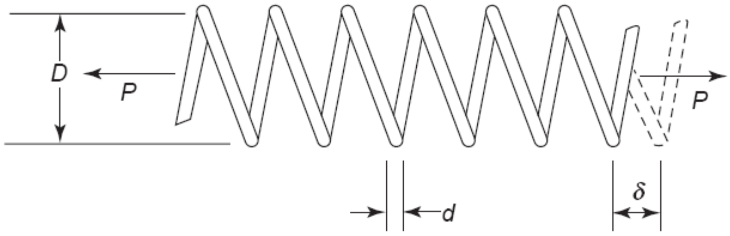

Figure 8. Schematic of the coil spring under static loading [18]

Test problem 7: A welded beam design problem

As the second example, fabrication cost of a welded beam of Fig. 9, is to be minimized subjected to constraints on bending stress σ, shear stress τ, buckling load P_c, end deflection δ and side constraint as given below. Design variables are $x_1 = h$, $x_2 = l$, $x_3 = t$ and $x_4 = b$ as depicted in Fig. 9. The overall fabricating cost of set-up, welding, labor and used material is introduced by the following objective function:

$$\textit{Minimize} \qquad f(X) = 1.10471 x_1^2 x_1 + 0.04811 x_3 x_4 (14.0 + x_2) \tag{18}$$

Constraints are listed as:

Subject to:

$$g_1(X) = \tau(X) - \tau_{\max} \le 0$$
$$g_2(X) = \sigma(X) - \sigma_{\max} \le 0$$
$$g_3(X) = x_1 - x_4 \le 0$$
$$g_4(X) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0$$
$$g_5(X) = 0.125 - x_1 \le 0$$
$$g_6(X) = \delta\{X\} - \delta_{\max} \le 0$$
$$g_7(X) = P - P_c(X) \le 0$$
$$\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$
$$\tau' = \frac{P}{\sqrt{2} x_1 x_2}, \; \tau'' = \frac{MR}{J},$$
$$R = \sqrt{(\frac{x_2}{2})^2 + (\frac{x_1 + x_3}{2})^2}, \; M = P(L + \frac{x_2}{2})$$
$$J = 2\left\{\sqrt{2} x_1 x_2 [\frac{x_2^2}{12} + (\frac{x_1 + x_3}{2})^2]\right\},$$

$$\sigma(X) = \frac{6PL}{x_4 x_3^2},\ \delta(X) = \frac{4PL^3}{E x_3^3 x_4},$$

$$P_C(X) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

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

$$x_i^{LB} \le x_i\ \le x_i^{UB},$$

$$X^{LB} = [x_1^{LB}, x_2^{LB}, x_3^{LB}, x_4^{LB}] = [0.1, 0.1, 0.1, 0.1],$$

$$X^{UB} = [x_1^{UB}, x_2^{UB}, x_3^{UB}, x_4^{UB}] = [2.0, 10.0, 10.0, 2.0]$$

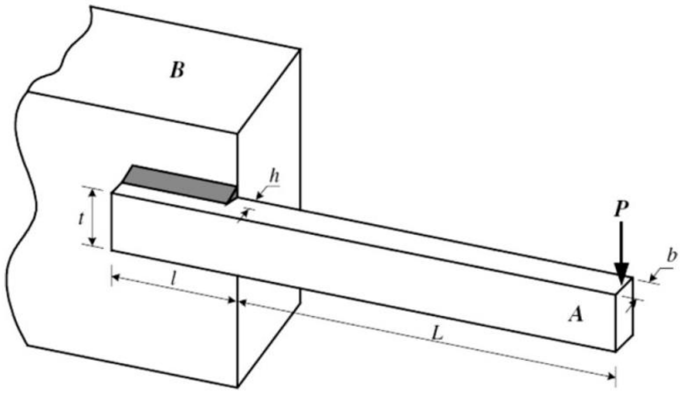

Figure 9. Schematic of the welded beam problem [19]

In this paper the treated algorithms search for the highest fitness function in each test problem defined with the following formulation:

$$\begin{aligned} &Maximize\, F(\underline{X}) = -f(\underline{X})(1 + k_{penalty} \times INF(\underline{X})) \\ &INF(\underline{X}) = \sum_i Q_i(\underline{X}) \\ &Q_i(\underline{X}) = \begin{cases} g_i(\underline{X}) & \text{if}\quad g_i(\underline{X}) > 0 \\ 0 & otherwise \end{cases} \end{aligned} \tag{19}$$

Whereas, Q_i denotes the amount of i^{th} constraint violation and $k_{penalty} = 50$ is the constant penalty coefficient**.**

Comparison of DFO variants

In this section, the proposed variants of DFO; i.e., DFOv1 and DFOv2 are compared with each other treating the referred test problems. Results of the most successive run regarding elitist fitness among a number of trials are provided in Tables 2 and 3 while statistical calculations including mean and standard deviation are also proposed as desired. Several parameters are evaluated including the optimal/elitist fitness F^*, the corresponding objective function OF^* and its infeasibility; $INF(X^*)$ and the design vector; $\underline{X}^*$, fitness improvement index; FI, iteration of last fitness improvement, LI and the number of fitness evaluations up to it; NFE.

Table 2: Evaluation of the best DOFv2 results

Test Problem:	1 (Mean:SD)	2 (Mean:SD)	3 (Mean:SD)	4 (Mean:SD)	5 (Mean:SD)	6 (Mean:SD)	7 (Mean:SD)
F*	0	0	-8.9E-16	0	-268.289 (-280.3:14.65)	-0.0130 (-0.0228 :0.0196)	-2.9814 (-3.7968:0.7218)
INF(X*)	0	0	0	0	0	0	0
OF*	0	0	8.88E-16	0	268.2889	0.0130	2.9814
NFE	550 (757:267)	5 (50:0)	50 (50:0)	650 (713:233)	5000 (3331:1096)	525 (995:722)	1000 (2100:915)
LI	22	2	2	26	200	21	40
FI	1	1	1	1	0 (0.016:0.026)	0.999 (0.992:0.007)	0.982 (0.695:0.327)
CPU time (s)	5.046	4.968	4.968	4.953	5.031	7.016	9.219
X*	0,0	0,0	0,0	0,0	0.770,0.505	0.050,0.315,14.545	0.432,2.377,6.584,0.480

Table 3: Evaluation of the best DOFv1 results

Test Problem:	1 (Mean:SD)	2 (Mean:SD)	3 (Mean:SD)	4 (Mean:SD)	5 (Mean:SD)	6 (Mean:SD)	7 (Mean:SD)
F*	0	0	-8.9E-16	0	-263.946 (-264.22 : 0.26)	-0.0102 (-0.0119 :0.0019)	-1.9556 (-2.1579:0.2489)
INF(X*)	0	0	0	0	0	0	0
OF*	0	0	8.88E-16	0	263.9457	0.0102	1.9556
NFE	2650 (2930:180)	4300 (2720:1427)	2600 (2227:706)	3150 (2510:542)	2950 (2797:1403)	4000 (4367:355)	4600 (4437:320)
LI	106	172	104	126	118	160	184
FI	1	1	1	1	0.074 (0.071:0.046)	0.994 (0.995:0.002)	0.540 (0.799:0.328)
CPU time (s)	7.312	7.281	7.328	7.312	7.312	10.547	13.921
X*	0,0	0,0	0,0	0,0	0.790,0.405	0.050,0.365,9.215	0.242,3.218,8.366,0.252

As can be observed from Tables 3 and 4 that both algorithms have captured the global optimum of first 4 test problems, however, taking different computational efforts. DOFv2 has shown much higher convergence rate than DFOv1; for example while it has captured the optimum by 50 fitness evaluations (2 iterations) it took more than 2500 (100 iterations) for the other variant of DFO in the 2nd and 3rd test functions. Besides, close fitness in the local optima of the Eggcrate function has delayed such a convergence to 26 and 126 iterations for the best trial of these methods. Testing with the 3-bar example, however, reveals a serious weakness of the DFOv2 in overriding a local optimum. It has not only shown no fitness improvement during 200 iterations of DFOv2 in this trial, but also has been stopped in an average result of -280.3 compared with -264.22 for DFOv1. In addition, the standard deviation of elitist fitness for DFOv1 vs. DFOv2 has a meaningful difference of 0.26 vs. 14.65 for this test problem. Similar observation can be driven from Tables 3 and 4 for the other two constrained test problems 6 and 7 (Spring design and Welded Beam design problem). Considering the best achieved FI, it can be realized that the 5th problem (3-bar design) has been more complex for both methods; however, DFOv1 has still shown more stable convergence regardless of the initial population.

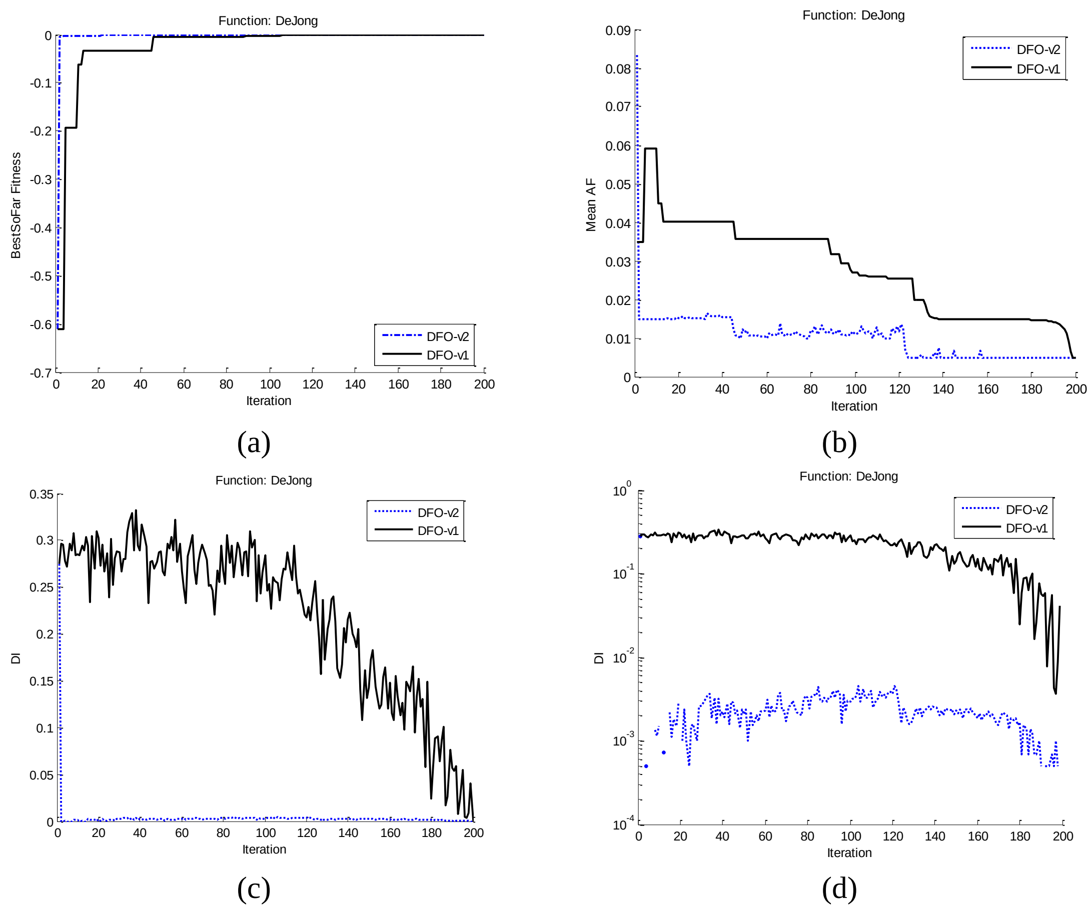

Figure 10. Comparison of DFOv1 and DFOv2 for the test problem 1

It is worth to trace the diversity variation to seek the reason of such behavior. Consider results of a trial run with the most fitness improvement by DFOv2 in test problem 2, Fig. 10. It is evident that DFOv1 has rapidly converged to the optimal fitness of 0 while DFOv2 has a more smooth and gradual convergence to this value. This observation is confirmed by tracing average pheromone trail (Mean AF) in Fig. 10b. It rapidly tends to its minimum for DFOv2, which means most of the population has found the best objective function. Note that in DFOv2 algorithm, $\alpha = 1$ leads to complete pheromone evaporation for the best individual, where no deposit is performed for the others. However, the condition is much different for DFOv1 when *MeanAF* first increases and then experiences a gradual decrease up to the last iteration when no further variation from already found optimum is desired.

Similar behavioral difference is declared considering DI history in Fig. 10c. Thus despite the DFOv2, the diversity of population individuals in searching optimum does not suddenly drop in early iterations of DFOv1. Instead for DFOv1, DI has in average a little fluctuation till later iterations where the region of true optimum is identified and then it tends to zero more rapidly. Such a compared behavior can be better seen using semi-log scale in Fig. 10d.

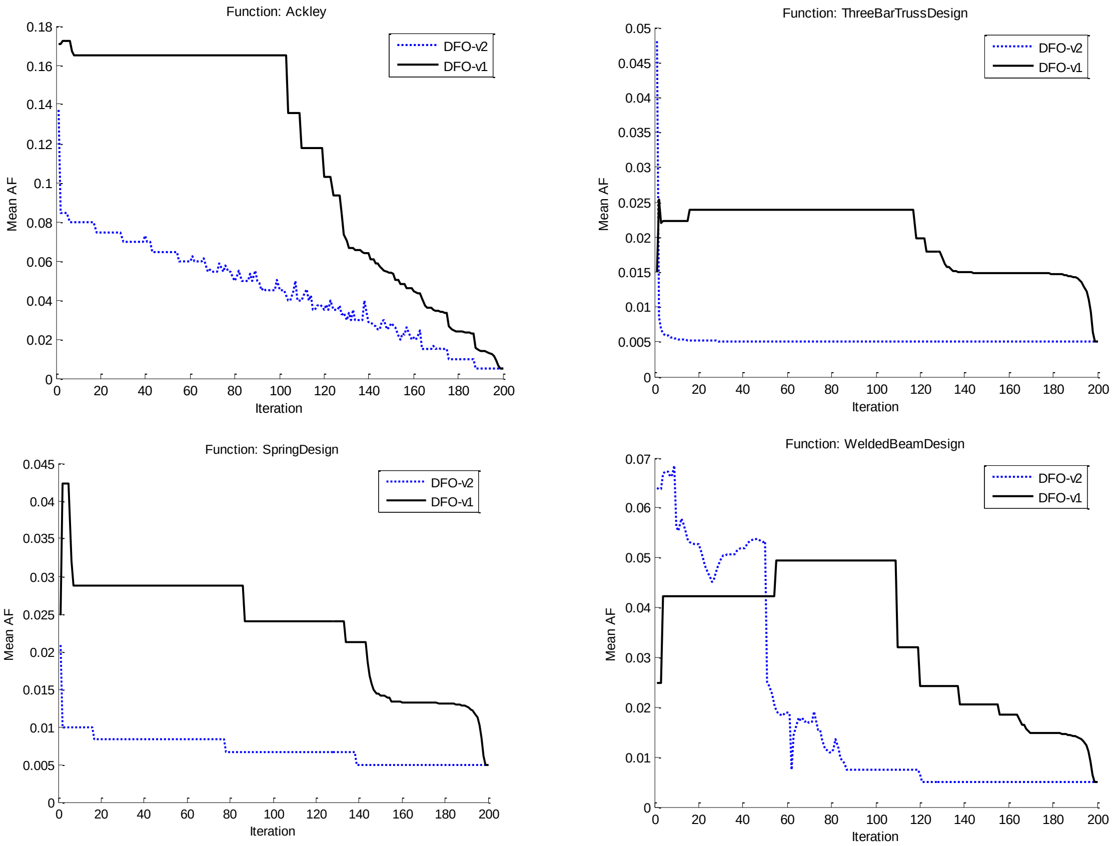

Figure 11. Comparison of DFOv1 and DFOv2 tracing mean AF

Another issue to investigate is the capability of the algorithms to change diversity variation and fuzzy-membership trails adaptively as the problem changes. Fig. 11 demonstrates history of mean trial for more the test problems 3, 5, 6, 7. It is evident that the fuzzy-membership values have resulted in trail variation which dynamically changes as the design space changes. For example, such mean value for DOFv2 is linearly decreased in the problem 3, stepwise in the problem 6, suddenly in the problem 5 and experiences a mixed linearly with sudden local drops and increases in test problem 7. Such a pattern not only depends on the problem but also depends on the implemented algorithm. For DFOv1 treating Ackley function, after an early peak it is almost constant up to half of iterations and then drops linearly with various dips. The early growth and step-wise variation is highlighted treating the problem 6 with a final S-shaped drop. The initial growth and width of the steps have been increased as the constraint narrowness or complexity or sharp-hills of the design space increases like in test problems 5 and 7.

Figure 12 shows the history of diversity index, DI, for the same problems and methods. As can be realized DFOv1 has considerable superior capability than DFOv2 in saving and either increasing DI as needed in the treated problem. In all the examples, DFOv1 has saved much more diversity than DFOv2 in the early iterations for better exploration in this phase and has gradually decreased it as the search progress to its final iterations in order not to loose already found optima in such an exploitative phase. The observation provides sufficient reasoning for results of Tables 3 and 4 in more quality of the achieved final optima by DFOv1 and also rapid/premature convergence of DFOv2 (for example in the test problem 5).

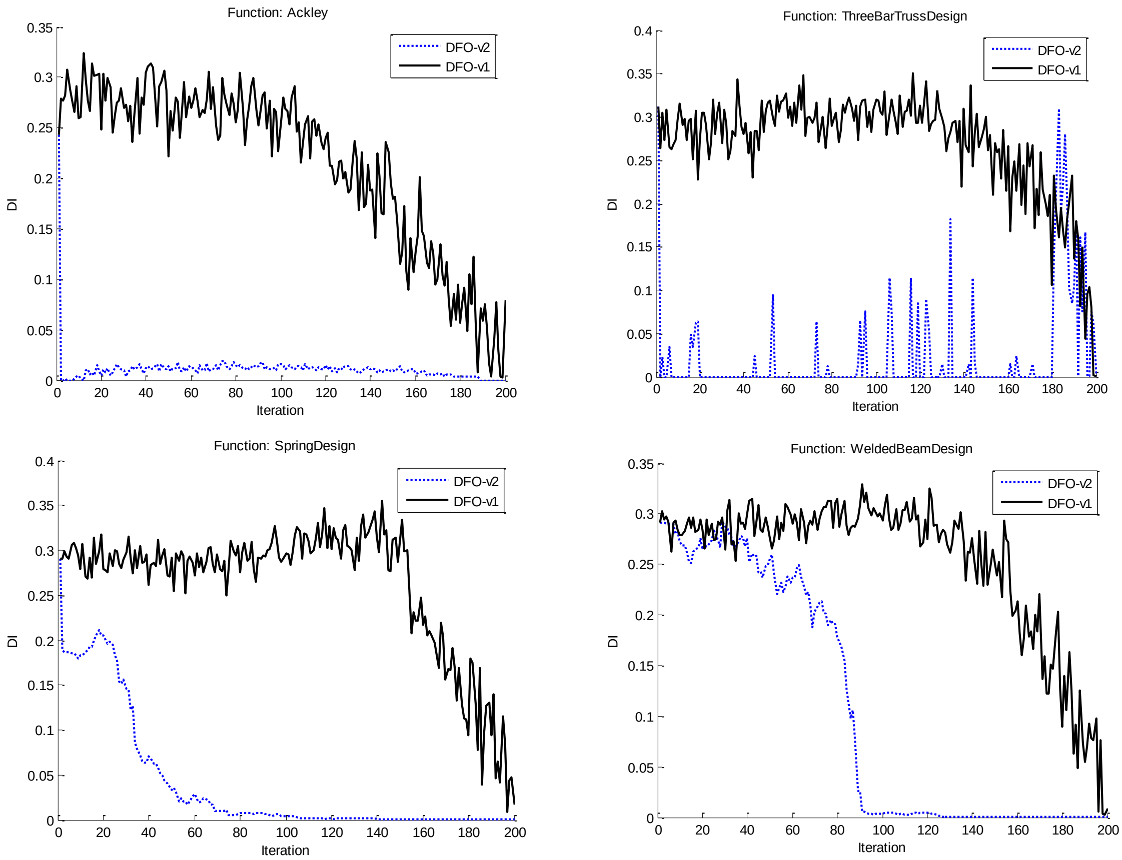

Figure 12. Comparison of DFOv1 and DFOv2 tracing DI.

Comparison with other algorithms

Two baseline algorithms are considered here for further comparison with DFOv1; these are *Genetic Algorithm*, GA and *Particle Swarm Optimization*, PSO [20]. Although many improved versions of these algorithms have been introduced by investigators, at the moment standard formulations of them are considered in this work. The genetic operators include tournament selection and a direct real-coded one-point crossover and mutation [21] with probability thresholds; 85% and 5%, respectively. For the standard PSO the following formulation:

$$\underline{V}^{i,t} = c_i \underline{V}^{i,t-1} + rand \times c_c(\underline{X}^{i,Pbest} - \underline{X}^{i,t}) + rand \times c_s(\underline{X}^{Gbest} - \underline{X}^{i,t}) \tag{20}$$

The inertial, cognitive and social constants are taken $c_i = 1, c_c = 2, c_s = 2$, respectively while *Pbest* stands for previous best experience of each particle and *Gbest* denotes the global best found so far. Once the velocity of each particle is determined by Eq.20, its position is updated using Eq. (8). The size of population and number of iterations are taken the same as DFO.

Comparison of DFOv1 vs. GA and DFOv1 vs. PSO are performed distinctly. Since for each comparison case the initial population is randomly generated and taken the same for the compared algorithms, the FI factor ratio between them is evaluated here to study effectiveness of algorithms in achieving better final solution. According to Table 4, it is declared that DFOv1 has better performance than GA both in best and average results. In addition, such superiority is more highlighted treating more complex test problems 3 and 5. Note that search space of the Ackley test function has a narrow symmetric global optimum valley while the constrained test problem 5 has a non-symmetric yet narrow shape about its minimum.

Comparison of DFOv1 with PSO in Table 4 reveals similar results; however, PSO has been better than GA in most of the treated problems except the Ackley's function. Sample plots of convergence history and DI trace in Figs. 13 and 14 for the best FI ratio confirms such a matter. According to Fig. 13, GA diversity index experiences many high fluctuations after early drop below DI history of DFOv1, which has prevent it to form proper converging to the fittest individual in spite of DFOv1. In the other hand, the average DI of PSO not only has not dropped during the search up to the final iteration. It is while DFOv1 after its smooth DI growth in early iterations has decreased DI near the end of optimization when local search to the true optimum is desired. Such a behavior has enabled DFO to continue its fitness improvements up to achieve fitter solution than the other two algorithms. These results show importance of maintaining higher DI in exploration and balancing it when intensifying to the final solution.

Table 2: Evaluation of the best DOFv1 vs. other algorithms results

Test Problem:	1 (Mean:SD)	2 (Mean:SD)	3 (Mean:SD)	4 (Mean:SD)	5 (Mean:SD)	6 (Mean:SD)	7 (Mean:SD)
FI_{DFO}/ FI_{GA}	1.046 (1.007:0.014)	1.629 (1.159:0.177)	1.556 (1.262:0.147)	1.067 (1.019:0.023)	5.523 (1.892:1.392)	1.485 (1.062:0.015)	1.596 (1.090:0.185)
FI_{DFO}/ FI_{PSO}	1.051 (1.001:0.015)	1.579 (1.124:0.175)	2.276 (1.319:0.341)	1.062 (1.019:0.020)	1.124 (1.033:0.038)	1.008 (1.002:0.002)	1.378 (1.076:0.115)

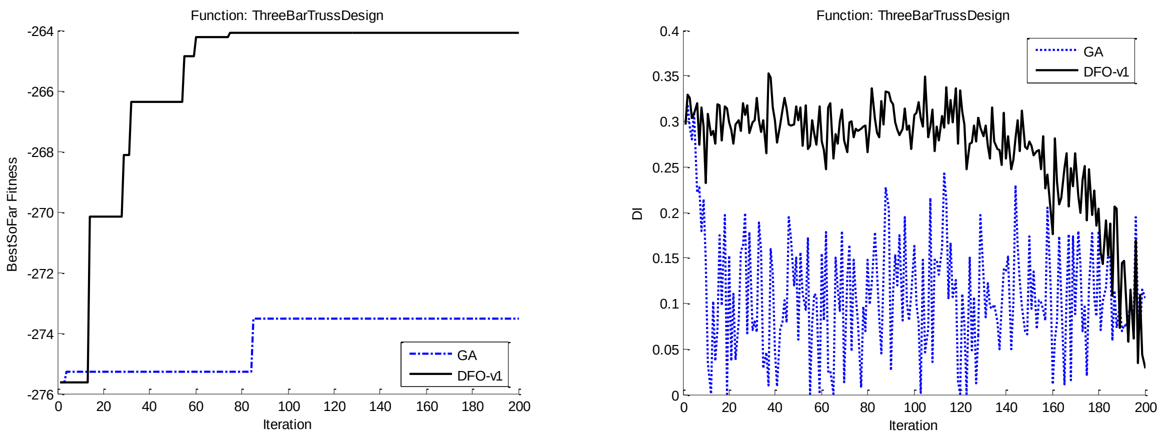

Figure 13. Comparison of GA vs. DFOv2 in 3-bar truss design problem

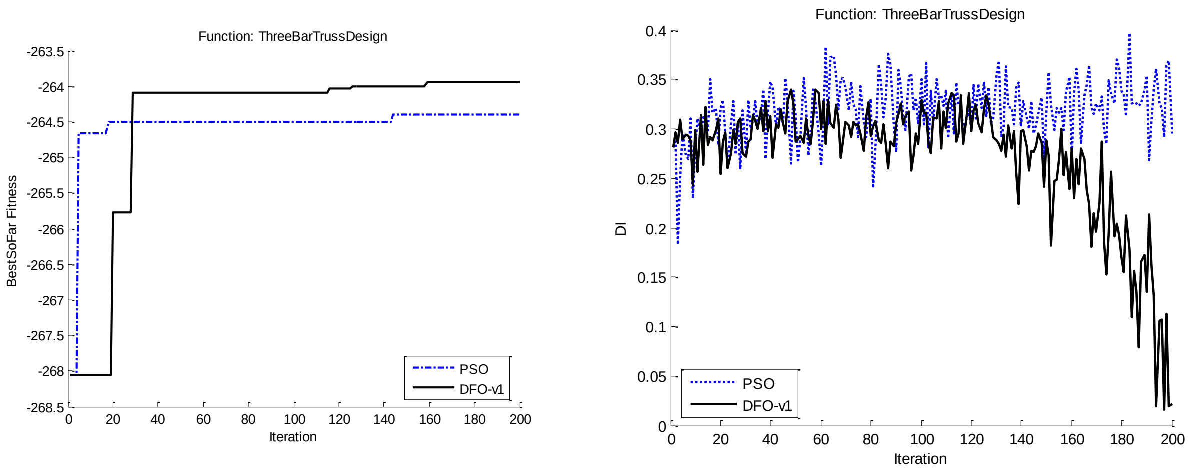

Figure 14. Comparison of PSO vs. DFOv2 in 3-bar truss design problem.

8. CONCLUDING REMARKS

In this paper a hybrid framework is developed to provide dynamic balance between diversification and intensification during the optimization process. It integrates vector-sum walks with probability-based walks through the design space and utilizes a special type of fuzzy-membership share for chosen alternatives between the search agents in a dynamic manner. Indirect information share between individuals is thus combined in real-time process with closeness to the optimal solution regarding both design space scope and fitness scope. A complete variant of the algorithm is then developed called DFOv1 using a normal-distribution fuzzy function while the other one, DFOv2, has suppressed random search portion and vector-sum walks and variables closeness norm function. Two variants are then compared to study such effects on their behavior. The performance of the algorithms are compared not only using the elitist fitness and number of fitness evaluations as effectiveness and efficiency measures but also by characteristic curves of diversity index and trail update histories.

Based on the achieved tests and result statistics in the treated examples, it is found that DFOv2 has very high convergence rate in simpler convex spaces but suffers from too rapid diversity drop to overpass local optima in more complex or constrained examples. In contrary, complete version of DFOv1 was more effective in such examples by saving proper diversity up to near final iterations and gradual intensifying to the final solution by reducing it at the end. Such a behavioral difference was confirmed regarding the mean trail based on fuzzy-membership overlaps during the search between the two algorithms. The complete version imposes a trial update curve like its diversity index history; that is a trail growth at early explorative iterations followed by step-wise/gradual trail drop as the search progress to the end. Because of evaporation procedure the mean trail is decreased when all the individuals intensify to a final solution. In the other hand the DFOv2 variant could not save proper diversity and trial amount up to near final iterations which describes why it was capable of premature convergence to local optima. It is worth notifying that variation and diverse positions of the individuals in every population of the search agents will enforce a dynamic fuzzy value integrated within the trail matrix to further guide new walks in the search space.

Treating different search spaces and test problems it is thus concluded that using complete trial update procedure will result in dynamic behavior so that the proposed algorithm can adapt its DI history with the search problem changes. Such a feature enabled DFOv1 in obtaining higher quality results in all treated cases particularly with zero infeasibility in the constrained examples.

Further comparison with genetic algorithm and particle swarm optimization confirms the special capability of the proposed algorithm in tuning diversity as the search progress while the others showed a monotonic average index. Regarding superiority of the proposed complete version of DFO in the treated examples, it is worth considering it as a powerful tool by a new procedure in tuning local and global search comparable with other existing meta-heuristic frameworks. Investigation of further issues such as the type of the employed fuzzy function or cooling procedure and algorithm adaptive performance in more practical examples is of course a future scope of our work.

REFERENCES

1. Atashpaz-Gargari E, Hashemzadeh F, Rajabioun R, Lucas C. Colonial competitive algorithm: A novel approach for PID controller design in MIMO distillation column process, *International Journal of Intelligent Computing in Cybernetics*, No. 3, **1**(2008) 337-55.
2. Rashedi E, Nezamabadipour H, Saryazdi S. GSA: a gravitational search algorithm, *Info Sciences*, No. 13, **179**(2009) 2232-48.
3. Sadollah A, Bahreininejad Eskandar H, Hamdi M. Mine blast algorithm for optimization of truss structures with discrete variables, *Computers and Structures*, **102-103**(2012) 49-63.
4. Chu S, Tsai P, Pan J. Cat swarm optimization. Lecture Notes in Computer, *Science* Springer-Verlag, **4099**(2006) 854-8.
5. Yang S. Firefly algorithm stochastic test functions and design optimization,

International Journal of Bio-Inspired Computation, No. 2, **2**(2010) 78-84.
6. Alatas B. A novel chemistry based metaheuristic optimization method for mining of classification rules, *Expert Systems with Applications*, No. 12, **39**(2012) 11080-8.
7. Yang XS, Gandomi AH. Bat algorithm: a novel approach for global engineering optimization, *Engineering Computations*, No. 5, **29**(2012) 464-83.
8. Holland J. Adaptation in natural and artificial systems, an introductory analysis with application to biology, control and artificial intelligence. Ann Arbor, The University of Michigan Press, 1975.
9. Dorigo M, Di Caro G, Gambardella LM. Ant algorithms for discrete optimization, *Artific Life*, **5**(1999) 137-72.
10. Kennedy J, Eberhart R. Particle Swarm Optimization, *Proceedings of the International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942-48.
11. Kaveh A, Farhoodi N. A new optimization method: Dolphin Echolocation, *Advances in Engineering Software*, **59**(2013) 53-70.
12. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search, *Simulation*, No. 2, 76(2001) 60-8.
13. Kaveh A, Shahrouzi M. Simulated annealing and adaptive dynamic variable band mutation for structural optimization by genetic algorithms, *Asian Journal of Civil Engineering*, **7**(2006) 655-74.
14. Kirkpatrick S, Gelatto CD, Vecchi MP. Optimization by simulated annealing, *Science*, **1983**(220) 671-80.
15. Shahrouzi M. Pseudo-random directional search: a new heuristic for optimization, *International Journal of Optimization in Civil Engineering*, No. 2, **1**(2011) 341-55.
16. Molga M, Smutnicki C. Test functions for optimization needs, www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf, 1995
17. Nowcki H. Optimization in pre-contract ship design. In Computer Applications in the Automation of Shipyard Operation and Ship Design in: Y. Fujita, K. Lind and TJ. Williams (Eds.), North- Holland, Elsevier, New York, **2**(1974) 327-38
18. Arora JS. Introduction to Optimum Design, 2nd Ed, Elsevier, 2004.
19. Kaveh A, Talatahari S. Novel heuristic optimization method: charged system search, *Acta Mechanica*, Nos. 3-4, **213**(2010) 267-89.
20. Kennedy J, Eberhart R. Swarm Intelligence, Academic Press, London, 2001.
21. Kaveh A, Shahrouzi M. Dynamic selective pressure using hybrid evolutionary and ant system strategies for structural optimization, *International Journal for Numerical Methods in Engineering*, No. 4, **73**(2008) 544-63.