I.J. Intelligent Systems and Applications, **2017, 6, 15-23**
Published Online June 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2017.06.02

Grass Fibrous Root Optimization Algorithm

Prof. Dr. Hanan A. R. Akkar
University Of Technology/ Electrical Engineering Department, Baghdad, Iraq
E-mail: dr_hananuot@yahoo.com

Firas R. Mahdi
University Of Technology/ Electrical Engineering Department, Baghdad, Iraq
E-mail: firasrasool1980@gmail.com

***Abstract*—**This paper proposes a novel meta-heuristic optimization algorithm inspired by general grass plants fibrous root system, asexual reproduction, and plant development. Grasses search for water and minerals randomly by developing its location, length, primary root, regenerated secondary roots, and small branches of roots called hair roots. The proposed algorithm explore the bounded solution domain globally and locally. Globally using the best grasses survived by the last iteration, and the root system of the best grass obtained so far by the iteration process and locally uses the primary roots, regenerated secondary roots and hair roots of the best global grass. Each grass represents a global candidate solution, while regenerated secondary roots stand for the locally obtained solution. Secondary generated hair roots are equal to the problem dimensions. The performance of the proposed algorithm is tested using seven standard benchmark test functions, comparing it with other meta-heuristic well-known and recently proposed algorithms.

***Index Terms*—**Grass development, Fibrous root system, Meta-heuristic algorithms, Grass Root Algorithm, Optimization.

I. Introduction

Optimization is a mathematical technique to find the best solution of constrained or non-constrained problems. An optimization problem is how to find variables and parameters that minimize or maximize a function called the objective function. Many optimization problems have more than one local solution. Therefore, it is important to choose a good optimization method that globally looks to the search space to find the best solution without being trapped into local minima solution. Heuristics can be defined as approaches to find optimal or near optimal solutions in a rational computational time without a guaranteed to find an optimal value, while meta-heuristics are a set of intelligent schemes that improve the efficiency of the heuristic procedures. Meta-heuristics can be classified according to the number of candidate solutions used at the same time. Trajectory methods are algorithms based on a single solution at any time and cover a local search based meta-heuristics, while population-based algorithms perform a search with many initial points in a parallel style. Meta-heuristic used in many scientific fields; such as neural network learning [1], and technical learning [2]. One of the most important population-based algorithms are the swarm intelligence algorithms. These algorithms were made of simple agents cooperating locally with each other depending on their environment, Each single element or particle follows one or numerous rules without any centralized structure for controlling its performance. Consequently, local and random interactions among the particles are directed to an intelligent global behavior. Many of these algorithms apply two approaches: global exploration and local exploitation, in which each particle improves its performance by co-operating, sharing information, and compete with other agents to survive. In spite of different nature inspiration sources of these algorithms, they have many similarities such as; initiate randomly, deal with uncertain, simultaneous computation, cooperation, competition, and iterative learning. Meta-heuristics algorithms dislike other exact mathematical methods, have no central control and if an individual fails, it does not affect the performance of the whole group. Therefore, they are more flexible and robust when acting on a complex, multimodal, and dynamic problems [3].

This paper proposes a novel meta-heuristic population-based algorithm that solves complicated optimization problems. The inspiration of the proposed algorithm was the fibrous root system and reproduction process of general grass plants. In Grass Root Algorithm (GRA) the global search performed at each iteration, while the local search performed when the global search is in stack condition or it does not lead to more improvement in the objective function. In order to test the proposed meta-heuristic GRA, seven standard test functions will be used. The obtained results will be compared with other nine well-known, and recently proposed algorithms which are Particle Swarm Optimization (PSO) [4], Differential Evolutionary algorithm (DEA) [5], Bee Colony Optimization (BCO) [6], Cuckoo Search Algorithm (CSO) [7], Wind Driven Algorithm (WDA) [8], Stochastic Fractal Search (SFS) [9], Symbiotic Organisms Search (SOS) [10], Grey Wolf Optimizer (GWO) [11], and Novel Bat Algorithm (NBA) [12].

II. Related Works

Many meta-heuristic optimization algorithms have been proposed in the last few decades. These algorithms optimize an objective function to get the optimum solution using two individual mechanisms; global exploration and local exploitation. In 1995 Kennedy and Eberhart [4] propose an algorithm inspired by social behavior of bird flocking or fish schooling called PSO. It shares many similarities with evolutionary computation techniques. PSO algorithm initialized with a population of random candidate solutions and searches for best solution by updating generations. In 2005 Dervis Karaboga [6] proposed an algorithm motivated by the intelligent behavior of honey bees called Artificial Bee Colony (ABC). ABC algorithm uses only common control parameters such as colony size and maximum cycle number. It provides population-based search techniques in which foods positions are adapted by the artificial bees with time. Bee's aim to explore the positions of food sources that have high nectar amount. In 2010 Z. Bayraktar and et al. [8] propose a novel WDO. It was a population-based iterative meta-heuristic optimization algorithm used for multi-dimensional and multi-modal problems. In WDO, small infinitesimally air parcels navigate over a solution search space follow Newton's second law of motion. WDO employs gravitation and Coriolis forces, provides extra degrees of freedom to fine tune the optimization.

This work proposes and implements a general grass root optimization algorithm GRA, comparing it with other meta-heuristic algorithms through using a variety of test function to evaluate the average mean absolute error, average number of effective iteration, and average effective processing time.

III. Grasses Development and Reproduction

There are three kinds of grasses fibrous root system which is; primary, secondary and hair roots. Primary roots are the first grown roots from the germinating seed. They provide the first few leafs with the required energy and minerals and stay active for a short period until the secondary roots become functional, then they die. Before the primary roots died the plant initiate new roots called secondary roots that develop at the same time new tillers and shoots. As energy is produced by developing leaves, some carbohydrates are partitioned to secondary roots for growth. These roots must compete with other neighbor roots to absorb water and minerals as they take over the function of the primary root. Hair roots are very small branches grow out of the epidermis of secondary roots. They absorb most of the water and minerals for the plant.

There are two major asexual reproduction methods in grasses; stolons which are stems creep along the ground, and rhizomes which are stems grow below the ground. Grasses use stolons and rhizomes to reach farther places and establish new grass culms [13]. Fig. 1 shows the grass asexual two reproduction methods.

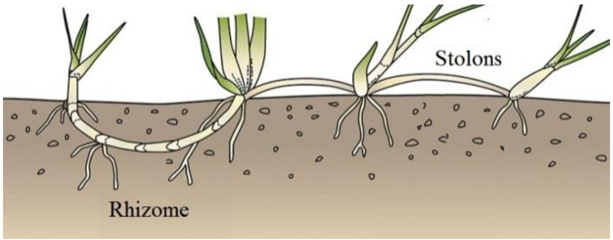

Fig.1. Grass two asexual reproduction methods.

IV. The Proposed Algorithm Govern Rules

Grass plants with stolons and rhizomes perform a global and local search to find better resources by reproduction of new plants and developing its own root system. Both the generated new grasses and roots are developed almost randomly. When a grass or root arrives at a place with more resources, the corresponding plant generates more secondary and hair roots, which makes the growth of the whole plant faster. If a grass plant has been trapped into local position it generates more new grasses by stolons initiated from the best-obtained grass and other grasses survived from the seeding process. On the other hand, new secondary roots with different deviation from the local minimum location will be generated by rhizomes. These secondary roots help plants reaching farther places and escaping from local solution. The rules that govern the grass model are illustrated as follows:

❖ *Global search mechanism includes:*

- Grass plants initiated randomly in the search space and only specified number of these plants survived according to the reached resources. The survived grasses generate a number of new plants deviated randomly from the original plants.
- The best grass obtained so far generates new grass plants using stolons. These newly generated grasses deviated from the best-obtained position with different search steps, usually greater than half of the maximum limit.

❖ *Local search mechanism includes:*

- The best grass obtained from the global search phase generates a random number of secondary roots using rhizomes. Each new generated secondary root contains hair roots equal to a problem dimensions.
- The secondary hair roots modify their length according to the step size vector.
- The best obtained secondary root in the local search phase generates new grass plants.

❖ The best-obtained grass grows faster than other and generates a larger number of deviated grasses and deviated secondary roots.

V. Grass Roots algorithm Mathematical Model

As other meta-heuristic optimization algorithms, GRA starts with initial random population (**P**) in the search space domain of the problem. Each individual grass in the population represents a grass initiated by seeding process. When iteration (*iter*) begins, a new population (**NP**) will be generated. **NP** consists of the fittest obtained grass represented by **GB** = min(*f* (**P**)), where *f* is the objective function, and *D* is the problem dimension, while the second element of **NP** is a number (*GN*) of grasses deviated from **GB** by stolons (**SD**). These stolons **SD** usually deviated from the original grass with step size usually less than the maximum limit of the upper bound **ub**, the last element of **NP** is new grasses equal to (*p–GN-1*) deviated randomly from the survived initial grasses (**SG**), *p* represents the population size.

$$GN = \left(\frac{amse}{amse + minm} \right) \left(\frac{p}{2} \right) \quad (1)$$

$$amse = \frac{1}{ps} \sum_{i=1}^{ps} mse_i \quad (2)$$

$$mse_i = \frac{1}{no} \sum_{j=1}^{no} (ds_j - y_j)^2 \quad (3)$$

$$minm = min\left(\mathbf{mse}\right) \quad (4)$$

$$\mathbf{mse} = \left[mse_1, \ldots, mse_i\,,\ \ldots,\ mse_{ps} \right]^T \quad (5)$$

Where *GN* is the number of generated new grass branches stolons **SD** deviated from **GB**. *amse* is the average of the Mean Square Error (MSE) values of all population **mse**, ds, y are the desired and actual output value. Each new branch grass **SD** deviated from **GB** as in (6), while the survived deviated grasses **SG** is represented by (7).

$$\begin{aligned} \boldsymbol{SD} &= ones\left(GN\,,1\right)*\boldsymbol{GB} + 2*max\left(\boldsymbol{ub}\right)*\ldots \\ &\left(rand\left(GN\,,1\right) - 0.5\right)*\boldsymbol{GB} \end{aligned} \quad (6)$$

$$\begin{aligned} \boldsymbol{SG} &= \boldsymbol{GS} + 2*max\left(\boldsymbol{ub}\right)*\ldots \\ &\left(rand\left(p\ ?\ GN\ -\ ,1\right) - 0.5\right)*\boldsymbol{ub} \end{aligned} \quad (7)$$

Where ones(c,1) represents ones column vector with c rows, and rand(c,1) is a random column vector with c rows ∈ [0,1]. Max(.) is an operator that finds the maximum element in a vector. **GS** is the (*p-GN-1*) highest fitness initial population. The new population (**NP**) will be represented by (8).

$$\mathbf{NP} = \left[\mathbf{GB^T},\ \mathbf{SD^T},\ \mathbf{SG^T} \right]^T \quad (8)$$

Grasses in **NP** will be bounded, evaluate their fitness and compared with the initial **P**. If the fittest new grass of **NP** is better than the old one of **P** then save the fittest new grass as the best solution. Otherwise, calculate the absolute rate of decrease in **mse** between the best obtained so far minimum **mse** (*bm*) shown in (9) and the current iteration minimum **mse** (*minm*) shown in (10). If the rate is less or equals a predefined tolerance value (*tol*) as in (11), then increase a global stack (st_g) counter by one. When the st_g reached its maximum value then move to the local search mechanism.

$$minm = min_{i=1,\ldots,p}\left(\mathbf{mse}\right) \quad (9)$$

$$bm = min_{j=1,\ldots,iter}\left(\mathbf{minmse}\right) \quad (10)$$

$$\left| \frac{minm - bm}{minm} \le tol \right|, \text{ at } j \text{ iteration} \quad (11)$$

The local search mechanism consists of two individual loops; secondary roots, and hair roots loops. The secondary roots will be iterated by a random number usually less than problem dimension *D*. On the other hand, hair roots loops are equal to *D*. Therefore, each secondary root generated by the **GB** will represent a local candidate solution. Each single hair root modifies its location as in (12-15) for a repeated loops less than secondary roots number (*S*).

$$LB_i^{k+1} = agb + GB_i^K + C2*\left(\varepsilon - 0.5\right) \quad (12)$$

k=1,2,...,S and i=1,2,...,D.

$$agb = \frac{1}{D} \sum_{i=1}^{D} LB_i \quad (13)$$

$$\mathbf{C} = [0.02, 0.02, 0.02, 0.2, 0.2, 2, 2, 2, 2, 15] \quad (14)$$

$$C_2 = \boldsymbol{C}\,(1 + \left\|\mu*10\right\|) \quad (15)$$

Where *LB* is the locally modified **GB** element by element, *S* is the number of secondary generated roots where (0 ≤*S*≤ *D*). **C** is the searching step size vector shown in (14), *C2* is a random element of **C** chosen according to the percentage repetition of **C** elements as shown in (15). μ and ε are two random numbers ∈ [0,1]. If the evaluated locally **LB** minimum MSE *lmin* is less than *bm*, then save **LB** as **GB**. Otherwise, calculate the absolute rate of decrease in MSE as in (11). If the rate is less than *tol* then increase local stack counter (st_l) by one, when st_l reached its maximum predefined value then break hair root loop and begin new secondary root loop. After each completed iteration, check if the stopping condition (*GError*) is satisfied, then stop iteration. Otherwise, go to next iteration until reached the *maxit* then stop. Table 1 illustrates the general proposed algorithm pseudo code. The corresponding flowchart will be shown in Fig. 2.

Table 1. GRA pseudo code

```
Initialize maxit, p, D, lb, ub, GError, tol, $st_g$, $st_l$, C, GB,
minm, bm , GN, f.
Initialize random grass population $P \in R^D$
Bound the initial population $lb \le P \le ub$, ub, lb $\in R^D$
For i=1 : popsize // check for best fitness particle in  the initial
                              population //
F= f (P) // f is a predefined objective function //
Calculate the mse(i) for F elements
End For
Sort the grass population (P) ascending according to mse
For iter=1 : maxit // global search  starting //
Evaluate $C_2$ according to (15)
Generate the new population (NP) as in (8)
Bound NP : $lb \le NP \le ub$, NP, ub, lb $\in R^D$
   For i=1: p // check  NP for the best fitness particle loop //
    $F_{new}$= f(NP)
    Mse(i)=$F_{new}$
    End For
Minm=min(Mse) // save minimum mean square error //
 Index the grass with the minm //index the position of the best
                                        particle in the population//
 Evaluate GN deviated from as in (1). //GN is the number of
                                       stolons or grass branches//
 Evaluate SD as in (6).
 Evaluate the GS  from the ascending sorted P
 Evaluate SG as in (7).
 If  Minm < bm then
 bm=Minm
 GB=best indexed grass
 $st_g$=0
 Else If (11) is true then
 increase $st_g$ by 1
 If $st_g$ is at its maximum then
 $st_g$=0 // Begin the local search //
     For q=1:S // secondary  root loop//
     $St_l$=0
         For j=1:D // hair root loop//
         LB=GB //initial LB//
         Evaluate LB as in (12)
         lmin = f(LB)
         If lmin<bm then
         $bm_e$=lmin
         GB=LB
         $st_l$=0
         Else If (12) with lmin instead of minm  is true then
         Increase $st_l$ by one
         Else
         $st_l$=0
         End If
         If $st_l$ is at its maximum then
         Break For
         End If
         End For (j loop)
     End For (q loop)
     End If
     End If
     If bm≤ GError then
     break For (iter loop)
     End If
End For (iter loop)
```

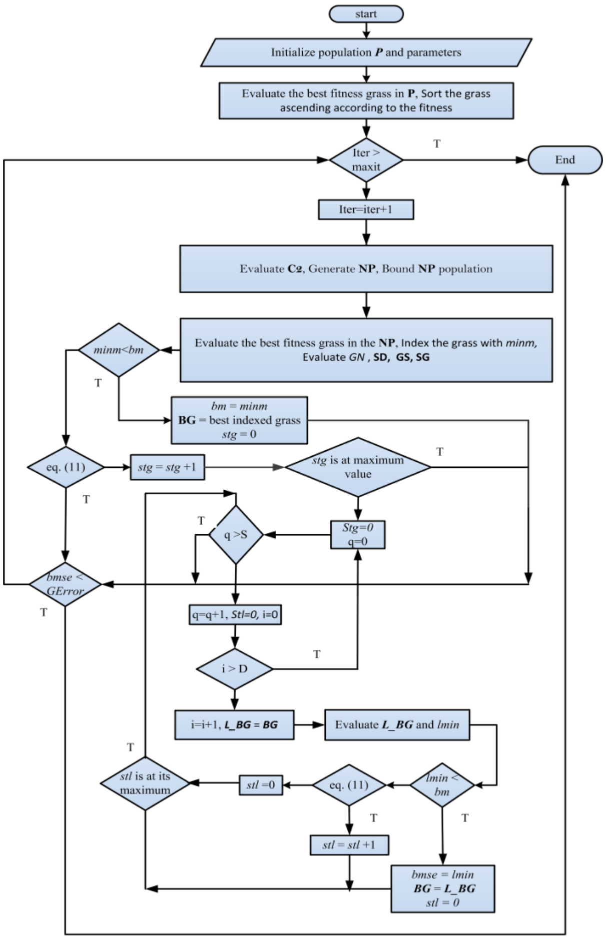

Fig.2. GRA flowchart.

VI. Test Functions Characteristics

Any new algorithm must have a validation for its performance by comparing it with other familiar algorithms over a good set of test functions. A common procedure in this field is to compare different algorithms on a large set of test functions. However, it is important to notice that the efficiency of an algorithm against others cannot be simply measured by means of problems that it solves, if the set of problems are too particular and without diverse properties. Therefore, in order to assess an algorithm, the kind of problems where it performs better than others must be classified, this is only possible if the test functions set is large enough to include an extensive variability of problems, such as multimodal, separable, scalable and multi-dimensional problems. The test functions used in this paper include multimodal functions which are functions with more than one local optimal, unimodal which have only a single optimum value that tests the ability of the proposed algorithm to escape local minima to a global one. The dimensionality of the search space and if the function is separable or not is another important issue related to the test functions.

VII. Experiment and Results

In order to test the proposed algorithm performance, a comparison with other algorithms was carried out. DEA, PSO, BCO are well-known algorithms which have been widely used in optimization, while WDA, CSO, SOS, SFS, NBA, and GWO are more recently algorithms. These algorithms have been compared to GRA over different test functions. The variety of the used test functions and their characteristics give us an indication of the abilities of these algorithms. However, for the purpose of testing and implementation the compared algorithms MATLAB 2013 have been used on a core i3 2.4 GHZ CPU computer with 4GB ram. Each algorithm runs ten times for each test function to calculate the average Mean Absolute Error (MAE). All algorithms are set to have the same; *p*, *maxit*, and *GError*, which are equal to 10, 100, and 0 respectively. The proposed GRA algorithm tolerance *tol* were equal to 1e-2. Table 2 illustrates the obtained algorithms average MAE and the percentage successful of the ten complete iteration cycles for each algorithm. Table 2 uses 7 benchmark test functions to compare the performance of the proposed algorithm with other 9 metaheuristic algorithms. However, the proposed algorithm has succeeded to optimize 85.71% of the test functions with 100% successful iteration cycles, as compared to 42.85% for SFS, 28.57% for SOS, 14.28% for NBA and WDA, and 0 % for GWO, DEA, BCO, CSO, and PSO. A small population size (10 particles) have been used to show the superior of the proposed algorithm over other tested algorithms. When increase the population size algorithms successful epochs rate will be increased. GRA algorithm succeeds to optimize Schwefel, and Michalewics test functions with 100% successful epochs and 50% for Dixon price, while other algorithms get 0% successful epochs for the same functions. Table 3 illustrates the effective average iterations number required by each algorithm to optimize each test function for the carried 10 cycles, while Table 4 illustrates the average effective processing time required to the effective average iteration number showed in Table 3. The shadowed blue cells in Tables 2,3, and 4 represent the succeeded algorithms to optimize the intended test functions. Therefore, the comparison (Bold marker) of the average processing time and average iteration number in Tables 3 and 4 will be restricted to these values only.

Table 2. Tested algorithms effective average AMAE.

Function	AMAE	BCO	CSO	DEA	GRA	GWO	NBA	PSO	SFS	SOS	WDO
Ackley [14]	avg.	3.48	0.256	3.73	5.15E-15	0.769	1.52E-03	3.82	**8.88E-16**	2.87E-09	0.739
	%	0%	70%	0%	**100%**	0%	**100%**	0%	**100%**	**100%**	70%
Rastrigin [15]	avg.	136	37.6	270	5.68E-15	45.5	69.6	328	**0**	**0**	29.7
	%	0%	0%	0%	**100%**	0%	0%	0%	**100%**	**100%**	0%
Schwefel [15]	avg.	5080	8550	9710	**0.00112**	7510	7200	7910	4080	5340	11000
	%	0%	0%	0%	**100%**	0%	0%	0%	0%	0%	0%
Zakharov [14]	avg.	61.1	445	746	1.16E-19	281	411	427	**1.92E-69**	1.76	8.16E-04
	%	0%	0%	0%	**100%**	0%	0%	0%	**100%**	20%	**100%**
Dixon price [14]	avg.	25.2	118	186	**0.335**	1.38	1.05	204	0.671	0.667	0.933
	%	0%	0%	0%	**50%**	0%	0%	0%	0%	0%	0%
Michalewics [15]	avg.	1.29	4.72	7.35	**2.97E-02**	3.79	5.46	5.83	0.797	2.53	5.84
	%	0%	0%	0%	**100%**	0%	0%	0%	0%	0%	0%
Shekel [14]	avg.	3.27	6.49	9.52	**5.01E-03**	4.13	5.41	4.9	0.875	3.47	5.41
	%	10%	0%	0%	**100%**	20%	0%	0%	70%	40%	0%

Table 3. Algorithms effective iterations number.

Fn	BCO	CSA	DEA	GRA	GWO	NBA	PSO	SFS	SOS	WDO
Ackley	100	100	100	**4.3**	100	75.8	100	11	35	100
Rastrigin	100	100	100	**7.5**	100	100	100	11.1	53.8	100
Schwefel	100	100	100	**100**	100	100	100	100	100	100
Zakharov	100	100	100	**9.3**	100	100	100	10.1	100	27
Dixon price	100	100	100	100	100	100	100	100	100	100
Michalewics	100	100	100	**100**	100	100	100	100	100	100
Shekel	100	100	100	**100**	100	100	100	100	100	100

Table 4. Algorithms effective processing time (sec).

Fn	BCO	CSA	DEA	GRA	GWO	NBA	PSO	SFS	SOS	WDO
Ackley	1.14	1.14	2.00	**0.30**	0.62	0.53	0.54	0.31	0.74	0.61
Rastrigin	0.74	0.81	1.22	**0.19**	0.45	0.59	0.43	0.26	0.97	0.49
Schwefel	0.77	0.87	1.23	**5.64**	0.37	0.55	0.43	1.95	1.44	0.38
Zakharov	0.75	0.87	1.40	0.37	0.52	0.70	0.50	**0.31**	1.82	0.34
Dixon price	0.78	0.92	1.43	5.15	0.38	0.50	0.36	1.73	1.34	0.40
Michalewics	0.87	1.64	1.90	**1.16**	0.65	1.30	1.50	2.00	1.53	0.78
Shekel	0.73	0.76	1.33	**0.65**	0.50	0.65	0.48	2.10	1.44	0.48

In Tables 3 and 4, we have seen that the proposed algorithm has gotten the lowest required effective iterations number for all functions succeeded to optimize. It also got the lowest required effective average processing time except for Zakharov test function. Actually, if we let algorithms complete the whole 100 iterations for each cycle. The proposed algorithm will require higher processing time than most of other tested algorithms, this is obviously clear in Dixon price test function. There are two reasons for this high processing time; firstly the proposed algorithm is more complicated than other tested algorithms, and the weakness in the trade off between the global exploration and local exploitation mechanisms. Figures (3-9) show the convergence curves for the tested algorithms, in which GRA converges faster than other algorithms. It also shows that GRA can reach an acceptable solution with fewer generations, fewer population, and fewer iterations as compared to other algorithms.

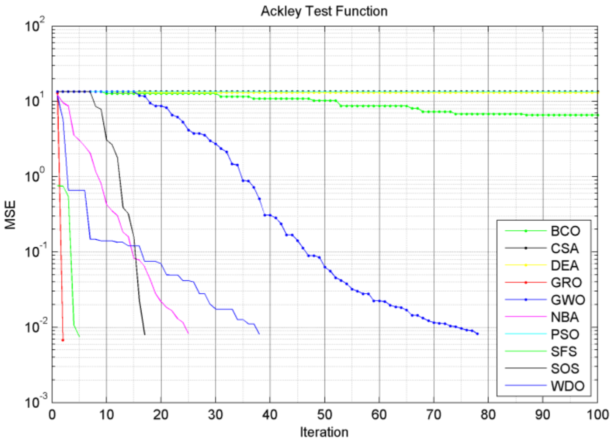

Fig.3. Ackley function algorithms convergence curves.

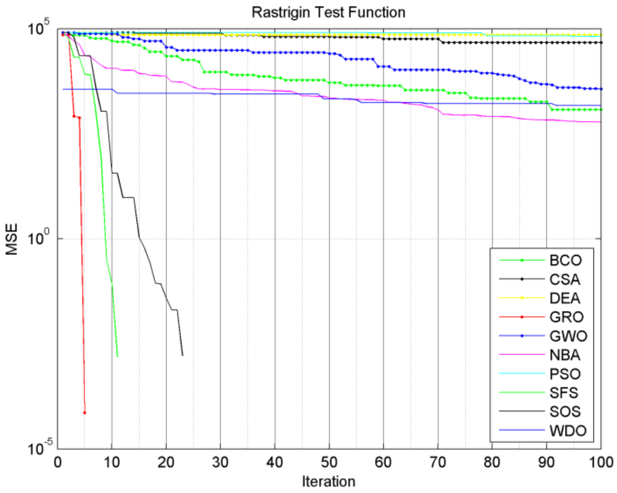

Fig.4.Rastrigin function algorithms convergence curves.

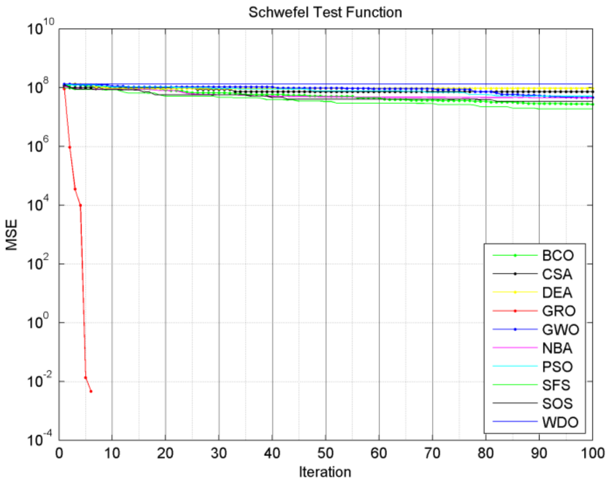

Fig.5. Schwefel function algorithms convergence curves.

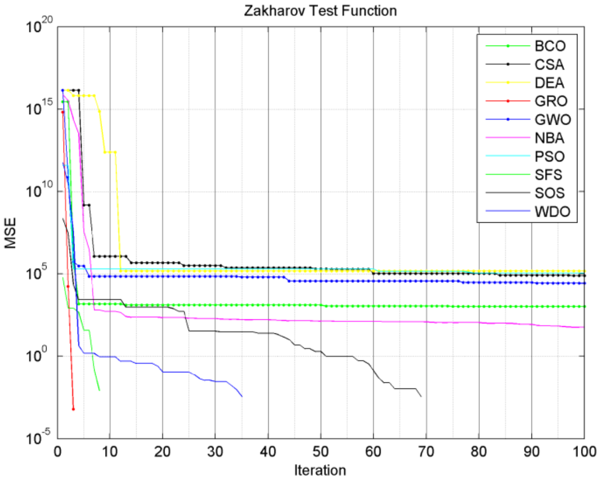

Fig.6. Zakharov function algorithms convergence curves.

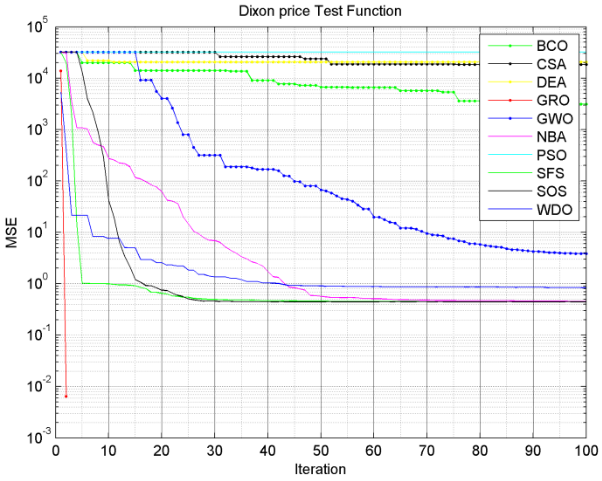

Fig.7. Dixon price function algorithms convergence curves.

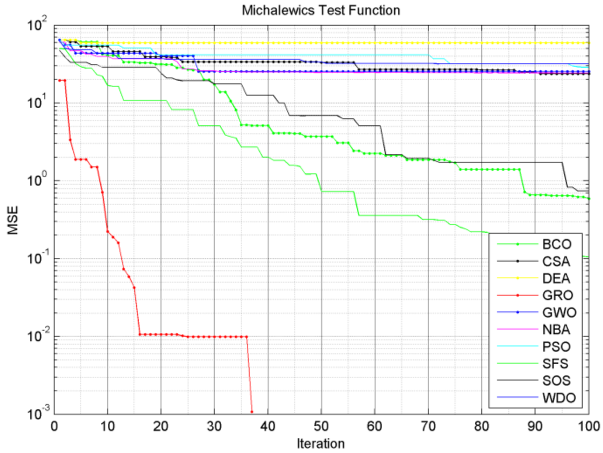

Fig.8. Michalewics function algorithms convergence curves.

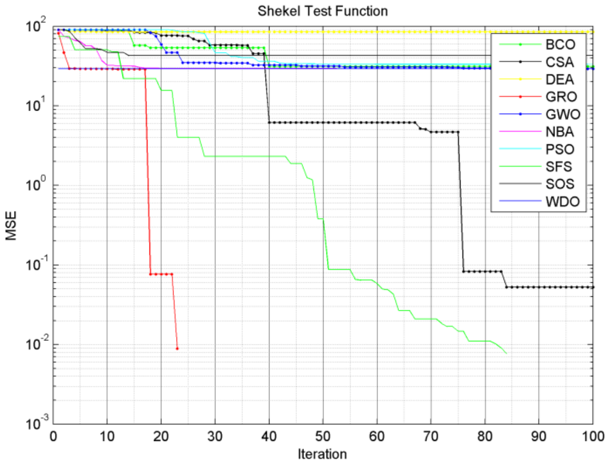

Fig.9. Shekel function algorithms convergence curves.

VIII. Discussion

Any new optimization algorithm must have the ability to explore a specified search space (global search) and at the same time exploit the best-obtained solution (local search). Therefore, any optimization algorithm must equip with the two searching abilities to search new regions, especially around the previous best solution. However, a good tradeoff between the two features leads to a promising optimization algorithm.

GRA has used both abilities exploration and exploitation, each enhanced with two different searching strategies. For exploration, GRA has used the survived grasses deviated from their original positions, and the new grasses generated from the best-obtained grass. At each iteration the survived grass, and the new grasses generated from the best grass (stolons), are simultaneously subjected to changes with different search size steps depending on the problem limits. Stolons length represent the search step size which is smaller or equal to the search space upper limit. The length of these steps should be considered sufficiently large such that the computational agents have a chance to move to any point in the search space and get rid of local minima trap. For exploitation, GRA enhanced with two different searching tools, which are the secondary roots (rhizomes), and the hair roots generated from the secondary roots, both secondary and hair roots are generated from the best global grass obtained so far. These tools are designed to search locally around the best solution of the current iteration and guarantees the adaption of the best solution of current iteration to the next iteration for doing farther search around it. In GRA the ratio of exploration and exploitation is managed by controlling the algorithm parameters, any parameters may indirectly affect this ratio. Maximum global stack and local stack counters in addition to *tol* have the most influence on the balance between exploration and exploitation. Increasing *tol* or maximum global stack counter, or decreasing maximum local stack counter leads to more exploration, while decreasing *tol* or maximum global stack counter, or increasing maximum local stack counter leads to more exploitation. Other parameters that affect the exploitation search of the proposed algorithm are the secondary roots or rhizomes number which represents how many times GRA tries to modify its local solution to get better performance and the searching step size vector C which will affect the length of secondary hair roots that will search around the best solution according to the secondarily generated roots.

IX. Conclusions

This paper proposes a new meta-heuristic optimization algorithm, inspired from the reproduction and root system of the general grass plants. GRA first search the problem using deviated from survived grasses, and reproduced grasses from the best grass for the global search, while secondary roots and hair roots are used for the local search. The performance of the proposed algorithm have been compared to familiar algorithms PSO, BCO, DEA, and other recently proposed algorithms SFS, SOS, NBA, GWO, CSO, WDA using seven test functions which have a variety of different characteristics. The obtained results showed that the proposed algorithm succeeded in optimizing test functions that cannot be optimized by other algorithms according to the chosen parameters and the search space limits, it also showed that GRA has faster convergence with a minimum number of iterations and generations than other compared algorithms.

References

[1] Koffka Khan,Ashok Sahai,"A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context", *International Journal of Intelligent Systems and Applications IJISA*, vol.4, no.7, pp.23-29, 2012.

[2] Harpreet Singh, Parminder Kaur,"Website Structure Optimization Model Based on Ant Colony System and Local Search", *IJITCS*, vol.6, no.11, pp.48-53, 2014.

[3] A. J. Umbarkar, N. M. Rothe, A.S. Sathe,"OpenMP Teaching-Learning Based Optimization Algorithm over Multi-Core System", International Journal of Intelligent Systems and Applications (*IJISA*), vol.7, no.7, pp.57-65, 2015.

[4] Alazzam, H. W. Lewis, "A New Optimization Algorithm For Combinatorial Problems," *Int. J. of Advanced Research in Artificial Intelligence*, Vol.2, Issue 5, pp.63-68, 2013.

[5] J. Kennedy, R. Eberhart, " Particle Swarm Optimization," *IEEE, Int. Conf., Neural Networks Proc.*, Vol.4, 1942 – 1948, 1995.

[6] F. Xue, Sanderson, A.C., Graves, R.J., "Multi-Objective Differential Evolution Algorithm Convergence Analysis and Applications," *IEEE conf., Evolutionary Computation*, vol. 1, Edinburgh, Scotland, pp. 743–750, 2005.

[7] D. Karaboga, B. Basturk, "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm," *Journal of Global Optimization*, Vol.39, Issue 3, pp. 459-471, 2007.

[8] X.S. Yang, S. Deb, "Cuckoo Search via Levy Flights", *IEEE conf. , Proc. of World Congress on Nature & Biologically Inspired Computing, (NABIC)*, pp. 210-214, 2009.

[9] Bayraktar, Z., Komurcu, M. , Werner, D.H., "Wind Driven Optimization (WDO): A Novel Nature-Inspired Optimization Algorithm and its Application to Electromagnetics," *IEEE Int. conf., Antennas and Propagation Society International Symposium (APSURSI)*, Toronto, pp. 1-4, 2010.

[10] H. Salimi, "Stochastic Fractal Search: A Powerful Meta-heuristic Algorithm," *Knowledge-Based Systems*, Vol. 75, 1-18, 2015.

[11] M. Cheng, D. Prayogo, "Symbiotic Organisms Search: A New Metaheuristic Optimization Algorithm," *Computers & Structures*, Vol. 139, pp. 98–112, 2014.

[12] S. Mirjalilia, S. M. Mirjalilib, A. Lewisa, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.

[13] Menga, X.Z. Gaob, Y. Liuc, H. Zhanga, "A Novel Bat Algorithm with Habitat Selection and Doppler Effect in Echoes for Optimization," *Expert Systems with Applications*, Vol. 42, no. 17–18, 6350–6364, 2015.

[14] C. Stichler, Grass Growth and Development, Texas Cooperative Extension, Texas A&M University.

[15] M. Jamil, and X.Yang, "A Literature Survey of Benchmark Functions for Global Optimization Problems", *Int. Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, Aug., 2013.

[16] X.S. Yang, "Appendix A Test Problems in Optimization," in Engineering Optimization: An Introduction with Metaheuristic Applications, 1st ed., New Jersey, USA, *John Wiely & Sons, Inc.,* pp. 261-166, 2010.

Authors' Profiles

Hanan A. R. Akkar received her Bachelor's Degree from the Electrical and Electronics Engineering Department at the University of Technology in 1988. She received her Master's degree and Ph.D. degree from the Electrical and Electronics Engineering Department at the University of Technology in 1994 and 1998, respectively. She has been Professor in the Department of Electrical Engineering at the University of Technology in the filled of ANN, FL, GA and swarm intelligent based on FPGA and electronic circuits. Currently she is Head of scientific committee in the Electrical Engineering Department at University of Technology.

Firas R. Mahdi received his Bachelor's Degree from the Electrical and Electronics Engineering Department at the University of Technology in 2004 Iraq, Baghdad. He received his Master's degree and Ph.D. degree from the same university, at the Electrical and Electronics Engineering Department in 2008 and 2017, respectively. His major field of study was in the artificial intelligence, evolutionary algorithms, and robotics controllers.

How to cite this paper: Hanan A. R. Akkar, Firas R. Mahdi,"Grass Fibrous Root Optimization Algorithm", International Journal of Intelligent Systems and Applications(IJISA), Vol.9, No.6, pp.15-23, 2017. DOI: 10.5815/ijisa.2017.06.02