

An Improved Tree Seed Algorithm for Optimization Problems

Murat Aslan, Mehmet Beskirli, Halife Kodaz, and Mustafa Servet Kiran

Abstract—Various heuristic algorithms have been proposed in the literature for solving optimization problems. Tree-seed algorithm (TSA) is inspired from relation between trees and seeds that a population-based evolutionary algorithm. when create process of a seed occur in TSA, the position updating of each dimension of the seed is calculated separately. In scope this study, Some changes have been implemented to original TSA. A new operator was added to the position update equation of original TSA when create a seed from tree. This operator is calculated by dynamically according to the dimension of the problem. As the dimension of the problem increases, the value of the this operator decreases. In addition, we determined an upper and a lower bound for the update process of the seed. The Improved Tree Seed Algorithm (ITSA) proposed in this study and the TSA have been tested on some benchmark functions in the literature. As a result, when the experimental results are taken into consideration, it is understood that the proposed algorithm ITSA is obtained more effective results for benchmark functions than TSA. Moreover, it is observed that ITSA found quite successful results compared with TSA for large-scale benchmark problems.

Index Terms—Evolutionary calculation, heuristic search, optimization problems, tree seed algorithm.

I. INTRODUCTION

Heuristic algorithms try to catch the best solution among all solutions by using existing knowledge for solving the problem defined as an optimization problem [1]. Linear and dynamic programming based algorithms do not guarantee to find optimum solution for complex NP-hard type problems [2]. For solving the complex optimization problems it is needed to a powerful computation at an exponential time [3]. Hereby, metaheuristic algorithms guarantee that to find an acceptable solution in a reasonable time under the guidance of the existing solutions [4].

Metaheuristic algorithms have been utilized to solve many different problems such as hydrothermal scheduling [5], forecast energy demand [6], travelling salesman problem [7], [8] and graph coloring problem [9], [10]. Optimization problems are generally defined as continuous or discrete optimization problems according to the values of decision variables. The values of decision variables in discrete optimization problems are integer numbers [11]. For example, for travelling salesman problem, each city is represented by integer values such as 1, 2, 3. However, for a problem that is defined as a continuous optimization problem, decision variables take real values [11]. For example, a decision variable can take all real values that can be defined between 0

and 1. Metaheuristic algorithms can be nature inspired or based on social behavior or musical. In addition, there are hybrid algorithms that are improved by utilizing some heuristic algorithms [12]. But the goal of all methods is to achieve acceptable solutions using existing knowledge properly. One of the most important benefits of metaheuristic algorithms is the convergence speed to precise solutions through the use of learning and adaptation techniques for solve real world problems [13]. Algorithms such as Particle Swarm Optimization (PSO) Algorithm [14], Ant Colony Optimization Algorithm (ACO) [15], Artificial Bee Colony (ABC) Algorithm [16], Bat Algorithm [17], Artificial Algae Algorithm (AAA) [18] are some of the metaheuristic algorithms which proposed in the literature.

The Tree-Seed Algorithm (TSA) is a new nature based metaheuristic algorithm that has been proposed by Kiran (2015) for the resolution of continuous optimization problems [19]. The main goal of this study is to improve the performance of TSA. Many approaches have been proposed in the literature by insertion some new operators into the original algorithms. The main reason of insertion new operators into the original algorithms is to increase the solution quality of the current algorithms. it has been improved a lot of algorithms in literature thanks to adding new operators into existing metaheuristic algorithms such as PSO, ACO and Shuffled Frog Leaping Algorithm (SFLA) to increase the performance of them. Jaballah *et al.* (2014) proposed the Improved Shuffled Frog Leaping Algorithm (ISFLA) by inserting two new operators such as c_1 and c_2 into the original SFLA. According to the experimental results of the study of Jaballah *et al.*, ISFLA has proved to be quite successful results compared to original SFLA [2]. Lovbjerg (2002) proposed the improved PSO algorithm by using the strategies of Genetic Algorithm and Hill Climbing algorithm in PSO algorithm. Aslan (2017) proposed an approach based on SFLA using the crossover and mutation operators of the Genetic Algorithm with to the original SFLA. According to the study of Aslan (2017), the changes made on the SFLA achieved more successful results than the original SFLA [9]. Haklı (2016), added new strategies in the original Cuckoo Optimization Algorithm (COA) in his study and applied it on the benchmark functions. He noted that changes he has made on the COA have improved the performance of the COA [20].

In this study, we improve a more effective Tree-Seed Algorithm by making some changes on the original TSA. In the second section, we explained TSA in short. The steps of the Improved Tree-Seed Algorithm (ITSA) are detailed in the third section. In the fourth section, the experimental results are given. In the last section of the study contains some information about the conclusion and future studies.

Manuscript received November 22, 2017; revised January 10, 2018.

The Authors are with the Computer Engineering Department, Selçuk University, Konya, Turkey (e-mail: murataslan@selcuk.edu.tr).

II. TREE-SEED ALGAE ALGORITHM (TSA)

Step 1: Set algorithm parameters
 Set the value of the Search Tendency (ST) parameter to be between (0-1).
 Set the dimension of the problem (D) and number of individuals in the population (N).
 Set termination criteria.

Step 2: Generate the initial population
 Generate the initial population randomly according to the dimension of the problem and the constraints.
 Calculate the fitness value for each individual of the population.
 Select the tree with the best fitness value (BestTree)

Step 3: Searching with Seeds
FOR All trees
 Set number of seeds at random between 1 and N (k)
 FOR $i=1$ to k
 FOR $j=1$ to D
 IF ($rand < ST$)
 Update the relevant dimension according to Eq. (1).
 ELSE
 Update the relevant dimension according to Eq. (2).
 END IF
 END FOR
 END FOR
 Select the seed with the best fitness value among all seeds
 Compare the best seed with the parent tree (parentTree).
 If the seed is in a better position than the parent tree, then update the position of parent tree by position of the seed.
END FOR

Step 4: Determine the best solution in the population
 Select the tree with the best fitness value (BestTree).

Step 5: Check the termination condition
IF Termination condition is not met.
 Go to Step 3.
ELSE
 Report the best solution.
END

Fig. 1. The framework of the tree-seed algorithm [19].

The Tree-Seed Algorithm (TSA) is a new nature based metaheuristic algorithm that has been proposed by Kiran (2015) for continuous optimization problems. In the Tree-Seed Algorithm, each tree represents a parent individual and each seed represents a child individual generated of a parent tree. In tree-seed algorithm, If the solution quality of the seed is better than the tree which this seed is included, then the position of the tree is updated by replacing the position of the existing tree with the seed individual. TSA try to discover the quality solutions by balancing the two mechanisms called exploration and exploitation. At the exploration phase, the population is generated randomly, dependent on the constraints of the problem. Thus, a wide search space is obtained to achieve high quality solutions. The parent individuals which named Trees are created at the exploration phase. The next phase called exploitation, the seeds are generated from the trees. In the exploitation phase, each dimension of each seed is updated according to the value of the Search Tendency (ST) parameter by the tree with the best fitness value or the ParentTree. Thanks to exploration and exploitation mechanisms of TSA, it is tried to reach quality solutions both in local and global search space. In each iteration of the TSA, seeds are created for each tree with a number k , which will be randomly between 1 and the number of individuals of the population. ParentTree, a tree selected randomly from the population except ParentTree and the tree with the best fitness value (BestTree) is used for generate the position of seeds. When create the seed from a tree, each dimension of the seed is created according to either

Eq. (1) or Eq. (2) dependent the ST value. Eq. (1) and Eq. (2) are given below. The ParentTree means selected tree, the seed represent the child generated from a tree, the i value indicates the i 'th seed generated from the tree, BestTree represents the tree with the best fitness value, Trees indicates the tree population, r value indicates the index of the randomly selected tree, j is related to dimension and rand is represent the randomly value generated between 0 and 1.

$$Seed_{i,j} = ParentTree_j + (BestTree_j - Trees_{r,j}) * (rand - 0.5) * 2 \quad (1)$$

$$Seed_{i,j} = ParentTree_j + (ParentTree_j - Trees_{r,j}) * (rand - 0.5) * 2 \quad (2)$$

When the dimension of a seed is updated according to the ST value, if the rand value is smaller than the ST value, the related dimension is updated according to Eq. (1), otherwise the related dimension is updated according to Eq. (2). The framework of the TSA are given in Fig. 1.

III. IMPROVED TREE SEED ALGORITHM (ITSA)

As the complexity of a problem increases, it is also being difficult to find the optimum solution of this problem. Therefore, in order to obtain effective solutions, we sometimes need to modify some structures of the original algorithms. It is quite difficult to find the ideal solution by original Tree-Seed Algorithm when the dimension of the problem increases. On account of this, some changes have been made on the TSA to improve the performance of TSA. The goal of these changes is to extend both the local and global search space. Some changes were made on the "Searching with Seeds" (Step 3) phase. Acceleration coefficient C parameter which is calculated according to the dimension of the problem was inserted into the original TSA. The steps of the TSA have been preserved except Step 3. In the scope of this study, Eq. (1) and Eq. (2) were rearranged. The equations (3) and (4) given below are the new equations used in this study.

$$\Delta_{i,j} = (BestTree_j - Trees_{r,j}) * (rand - 0.5) * C \quad (3)$$

$$\Delta_{i,j} = (ParentTree_j - Trees_{r,j}) * (rand - 0.5) * C \quad (4)$$

C parameter which used for Eq. (3) and Eq. (4) is calculated by Eq. (5). The D parameter given in Eq. (5) represents the dimension of the problem. The maximum value of the C coefficient is 2 and the minimum value is 1, 15. At the end of experimental runs with different parameters, it is noticed that the success of the algorithm decreases when the value of parameter C rises above 2 or below 1, 15. Therefore, the minimum and maximum limit values for the C coefficient were selected sequentially 1, 15 and 2.

$$C = 2 - (D^2 * 0.0001) \quad (5)$$

The second change in TSA has boosted the efficiency of the parent tree. The lower and upper bound for the value of $\Delta_{i,j}$ which collected with the relevant dimensions of the parent tree when create a seed from the parent tree were determined. Thus, a restriction is made for $\Delta_{i,j}$. The upper

and lower bounds for $\Delta_{i,j}$ are given in Eq. (6).

$$\text{Min} \leq \Delta_{i,j} \leq \text{Max} \quad (6)$$

Minimum and Maximum values of $\Delta_{i,j}$ are calculated sequentially $\text{Min} = -0.1 * (d_{\text{max}} - d_{\text{min}})$ and $\text{Max} = 0.1 * (d_{\text{max}} - d_{\text{min}})$. d_{max} represents the upper bound of the problem and, d_{min} represents the lower bound. After that $\Delta_{i,j}$ is added to the current parent tree for position of seed. Eq. (7) shows the position of a seed.

$$\text{Seed}_{i,j} = \text{ParentTree}_j + \Delta_{i,j} \quad (7)$$

IV. EXPERIMENTAL RESULTS

In this study, TSA and proposed algorithm (ITSA) have been tested on some benchmark functions. Both algorithms were tested under the same conditions. The numbers of the population were selected as 10,20,30,40 and 50 respectively. For each number of populations, independent tests were executed with the dimension of the problem being 20,50, and 100 respectively. The ST value was chosen as 0,1 as suggested by Kiran (2015) [19]. The C parameter is calculated to be dynamic according to the dimension of the

problem. C coefficient is determined by Eq. (5) is given above. Algorithms are run 30 times for each function separately and standard deviation values (Std.), Best values and Average values are given in the Table 2. For experiments, the termination condition is chosen as the maximum number of function evaluations (MaxFEs). MaxFEs is calculated according to the Eq. (8) proposed by Suganthan *et al.* [21]. Algorithms are written and run in Matlab 2010a environment. For experiments we used a computer with Intel i5 2.23 GHz processor and 8 GB memory.

$$\text{MaxFEs} = D * 10,000 \quad (8)$$

A. Benchmark Functions

The benchmark functions used in this study for experiments are named from F1 to F5 and are given in Table I. All of the benchmark functions given in Table I are minimization problems. The number of the functions are given in the Fn. column, the name of the benchmark functions are given in the Name column, the boundary value ranges of the functions are given in the Search Range column and the equation of the functions are given in the Function column. F1 and F4 are unimodal benchmark functions and F2, F3 and F5 are multimodal benchmark functions.

TABLE I: BENCHMARK FUNCTIONS USED IN EXPERIMENTS

Search Range	Function
$[-100,100]^D$	$F1(x) = \sum_{i=1}^N x_i^2$
$[-5.12,5.12]^D$	$F2(x) = \sum_{i=1}^N x_i^2 - 10 \cos(2\pi x_i) + 10$
$[-600,600]^D$	$F3(x) = \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
$[-10,10]^D$	$F4(x) = \sum_{i=1}^{N-1} [100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
$[-32,32]^D$	$F5(x) = 20 + e + -20 \exp\left(-0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}\right) - \exp\left(\frac{1}{N} \sum_{i=1}^N \cos(2\pi x_i)\right)$

B. Obtained Results

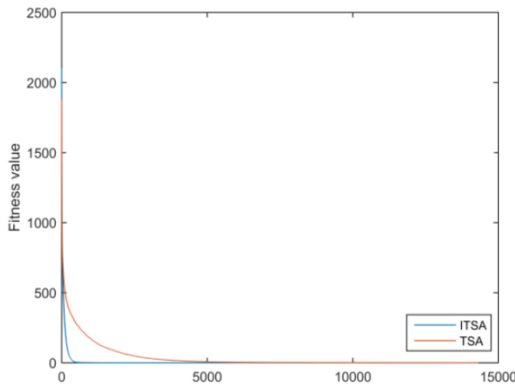


Fig. 2. Convergence of Sphere function.

Experimental results of ITSA and TSA are given in Table II for the different population numbers and different dimension numbers. When Table II is examined, it is observed that the proposed algorithm ITSA has found more successful results than TSA in generally. The most important

disadvantage of TSA is the number of seeds to be created from a tree is selected randomly in each iteration. If the number of populations is small, the number of seeds from the tree will be small too. However, if the number of populations is increase, the number of seeds to be created from the tree will also increase. If the number of seeds created from the tree increases, it will cause premature termination of the algorithm. So the efficiency of the algorithm will be reduced. Kiran (2015) determined the number of seeds randomly for every iteration between 10% and 25% of population in order to increase the efficiency of the algorithm [19]. But in this study, there is no restriction on the number of seeds to be created from the tree. Since the TSA's termination condition is MaxFEs (number of function evaluations), increasing in the number of seeds will cause the algorithm to prematurely terminate. Both TSA and ITSA have achieved more effective results for experiments with fewer populations. If Table II is examined, it is understood that TSA finds better results for F2 function. However, for the other benchmark functions ITSA finds better results under the conditions of different dimensions and different number of populations. When the

dimension of the problem is selected as 100, the proposed algorithm is quite successful than TSA.

The convergence graphs obtained from experiments when the number of populations is 20 and the dimension of the problem is 100 are given in the following figures. The convergence graph of Sphere function is given in Fig. 2. If the Fig. 2 is analyzed, it is realized that after the fitness value falls below the 1, the proposed algorithm quickly converges towards 0. The convergence of the TSA is slower than ITSA. However, it is seen that after 5000 FEs, convergence of both algorithms fitness value is similar. The convergence graph of Griewank function is given in Fig. 3. The convergence of the Griewank function is similar to the Sphere function. After the value of the objective function decreased to below 500, ITSA quickly converges to 0. The convergence graph of Rosenbrock function is given in Fig. 4. After the value of the objective function decreased to below 2, if Fig. 4 is examined, it is understood that the convergence speed of ITSA is better than TSA. The convergence graph of Ackley function is given in Fig. 5. When the convergence graph for the Ackley function is analyzed, the convergence speeds of TSA and ITSA are quite different from each other. For Ackley

function, the convergence of the TSA for reach the 0 value is quite slow. TSA converges the 0 value after approximately 15000 FEs. However, according to Fig. 5, it is seen that ITSA quickly converges towards 0. ITSA converges the 0 value after approximately 1000 FEs.

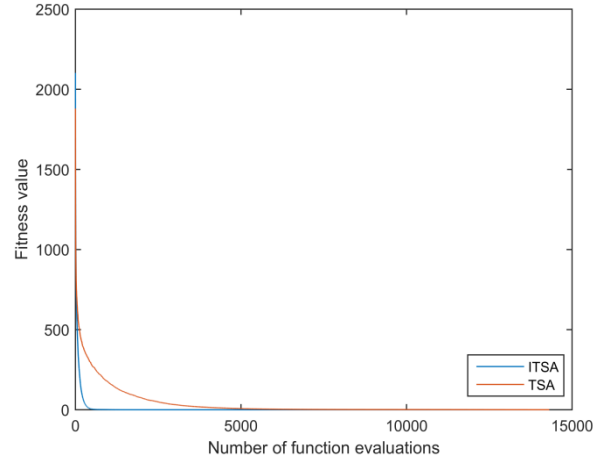


Fig. 3. Convergence of Griewank function.

TABLE II. EXPERIMENTAL RESULTS OBTAINED AFTER 30 INDEPENDENT RUNS FOR DIFFERENT POPULATION NUMBERS

Fn.	D	Value	pop =10		pop =20		pop =30		pop =40		pop =50	
			TSA	ITSA	TSA	ITSA	TSA	ITSA	TSA	ITSA	TSA	ITSA
F1	20	Best	5,15E-188	1,79E-194	6,30E-84	2,12E-87	3,92E-51	1,23E-52	9,95E-36	8,46E-37	2,12E-26	3,21E-27
		Average	2,38E-184	1,10E-189	6,84E-82	1,31E-84	1,03E-49	2,06E-51	7,09E-35	5,49E-36	1,22E-25	2,69E-26
		Std.	0,00E+00	0,00E+00	1,50E-81	5,35E-84	1,10E-49	2,83E-51	4,95E-35	4,33E-36	6,76E-26	2,41E-26
	50	Best	1,81E-74	7,85E-117	5,77E-26	3,34E-46	2,32E-14	2,69E-27	6,54E-09	1,09E-18	4,16E-06	1,72E-13
		Average	1,15E-71	2,38E-113	1,21E-24	4,33E-45	1,69E-13	1,65E-26	1,48E-08	5,82E-18	9,15E-06	4,12E-13
		Std.	2,66E-71	8,79E-113	1,13E-24	5,55E-45	9,48E-14	1,74E-26	6,99E-09	4,19E-18	3,33E-06	1,92E-13
100	Best	1,16E-21	8,99E-156	4,70E-02	5,88E-66	6,00E+01	1,86E-39	5,94E+02	3,59E-27	1,63E+03	5,16E-20	
	Average	1,95E-18	6,20E-154	3,36E-01	6,46E-65	1,14E+02	6,50E-39	9,50E+02	7,25E-27	2,34E+03	1,19E-19	
	Std.	6,61E-18	8,69E-154	3,72E-01	1,24E-64	4,52E+01	4,03E-39	2,00E+02	2,64E-27	3,58E+02	3,87E-20	
F2	20	Best	2,98E+00	9,95E-01	1,32E-02	7,21E+00	1,56E+01	3,68E+01	1,54E+01	3,78E+01	4,19E+01	4,04E+01
		Average	1,07E+01	1,42E+01	1,41E+01	4,24E+01	4,36E+01	5,45E+01	5,46E+01	5,93E+01	7,06E+01	6,10E+01
		Std.	5,00E+00	1,33E+01	1,37E+01	1,37E+01	1,77E+01	7,35E+00	1,63E+01	7,79E+00	1,39E+01	7,15E+00
	50	Best	2,59E+01	2,46E+01	2,59E+01	1,66E+02	3,38E+01	2,85E+02	1,09E+02	2,68E+02	1,85E+02	2,80E+02
		Average	5,65E+01	1,78E+02	5,20E+01	3,01E+02	1,44E+02	3,14E+02	2,46E+02	3,13E+02	2,97E+02	3,17E+02
		Std.	1,69E+01	8,00E+01	2,28E+01	2,98E+01	7,82E+01	1,18E+01	6,86E+01	1,51E+01	5,98E+01	1,26E+01
100	Best	1,33E+02	6,75E+02	9,78E+01	7,69E+02	1,25E+02	7,85E+02	2,17E+02	8,00E+02	3,58E+02	8,25E+02	
	Average	1,68E+02	8,02E+02	1,36E+02	8,25E+02	2,33E+02	8,39E+02	4,74E+02	8,45E+02	6,13E+02	8,54E+02	
	Std.	2,26E+01	3,74E+01	2,34E+01	2,41E+01	9,55E+01	2,28E+01	1,66E+02	1,55E+01	1,34E+02	1,81E+01	
F3	20	Best	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
		Average	2,38E-03	2,23E-03	9,25E-04	2,29E-06	5,01E-04	1,91E-04	2,47E-04	1,50E-04	1,86E-03	6,01E-04
		Std.	5,31E-03	3,98E-03	3,57E-03	1,25E-05	2,21E-03	9,48E-04	1,35E-03	5,70E-04	8,87E-03	1,86E-03
	50	Best	0,00E+00	0,00E+00	0,00E+00	0,00E+00	9,61E-14	0,00E+00	1,32E-08	0,00E+00	7,47E-06	4,57E-12
		Average	8,22E-04	9,82E-05	0,00E+00	1,85E-17	1,42E-12	2,59E-16	4,44E-08	2,16E-11	3,08E-05	2,98E-09
		Std.	2,53E-03	5,17E-04	0,00E+00	1,01E-16	3,14E-12	1,42E-15	2,15E-08	8,09E-11	2,09E-05	9,36E-09
100	Best	0,00E+00	0,00E+00	2,78E-02	0,00E+00	1,54E+00	0,00E+00	4,84E+00	0,00E+00	1,83E+01	0,00E+00	
	Average	9,86E-04	3,75E-04	1,78E-01	0,00E+00	2,03E+00	0,00E+00	8,83E+00	0,00E+00	2,29E+01	3,30E-13	
	Std.	2,56E-03	1,44E-03	1,10E-01	0,00E+00	4,07E-01	0,00E+00	1,97E+00	0,00E+00	3,13E+00	1,73E-12	
F4	20	Best	3,45E-03	6,30E-01	8,07E+00	8,48E+00	8,24E+00	1,21E+01	1,25E+01	1,32E+01	1,30E+01	1,40E+01
		Average	7,20E+00	7,04E+00	1,05E+01	1,02E+01	1,25E+01	1,26E+01	1,36E+01	1,36E+01	1,43E+01	1,43E+01
		Std.	1,17E+01	9,28E+00	1,06E+00	5,71E-01	9,47E-01	2,90E-01	2,83E-01	1,92E-01	3,54E-01	1,56E-01
	50	Best	1,38E+01	1,40E+01	3,96E+01	3,85E+01	4,44E+01	4,21E+01	4,57E+01	4,37E+01	5,29E+01	4,47E+01
		Average	5,00E+01	3,32E+01	4,43E+01	3,98E+01	4,52E+01	4,26E+01	5,26E+01	4,40E+01	1,16E+02	4,50E+01
		Std.	3,05E+01	9,85E+00	6,64E+00	5,94E-01	4,35E-01	3,95E-01	2,27E+01	1,78E-01	6,07E+01	1,66E-01
100	Best	1,34E+00	6,39E+01	1,80E+02	8,11E+01	2,01E+03	8,70E+01	3,56E+04	8,98E+01	4,97E+04	9,18E+01	
	Average	1,57E+02	8,23E+01	6,54E+02	8,24E+01	3,18E+04	8,75E+01	1,59E+05	9,03E+01	2,58E+05	9,21E+01	
	Std.	6,15E+01	2,03E+01	5,73E+02	2,13E+00	3,14E+04	1,05E+00	7,75E+04	1,75E-01	7,46E+04	1,53E-01	
F5	20	Best	2,66E-15	2,66E-15	2,66E-15	2,66E-15	2,66E-15	2,66E-15	2,66E-15	2,66E-15	4,17E-14	2,04E-14
		Average	3,49E-15	3,61E-15	2,78E-15	2,66E-15	2,66E-15	2,66E-15	2,66E-15	2,90E-15	1,07E-13	5,43E-14
		Std.	1,53E-15	1,60E-15	6,49E-16	0,00E+00	0,00E+00	0,00E+00	1,35E-15	9,01E-16	3,83E-14	2,20E-14
	50	Best	6,22E-15	6,22E-15	9,86E-14	6,22E-15	3,75E-08	2,40E-14	1,76E-05	3,45E-10	4,33E-04	1,15E-07
		Average	1,13E-14	1,08E-14	2,45E-13	1,00E-14	8,19E-08	4,88E-14	2,75E-05	5,87E-10	6,36E-04	1,81E-07
		Std.	2,75E-15	2,97E-15	1,35E-13	2,27E-15	2,71E-08	1,59E-14	7,56E-06	2,09E-10	1,36E-04	4,22E-08
100	Best	2,74E-12	2,75E-14	1,85E-02	2,04E-14	2,41E+00	2,04E-14	5,21E+00	4,88E-14	6,50E+00	1,14E-10	
	Average	3,12E-11	2,88E-14	5,37E-02	2,41E-14	3,51E+00	2,71E-14	5,99E+00	6,82E-14	7,71E+00	1,77E-10	
	Std.	2,88E-11	2,38E-15	3,09E-02	3,55E-15	4,86E-01	1,54E-15	4,44E-01	1,05E-14	4,24E-01	3,41E-11	

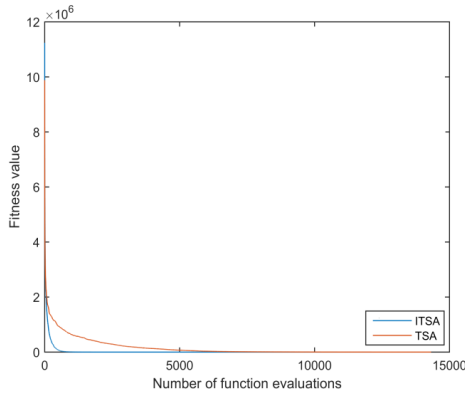


Fig. 4. Convergence of Rosenbrock function.

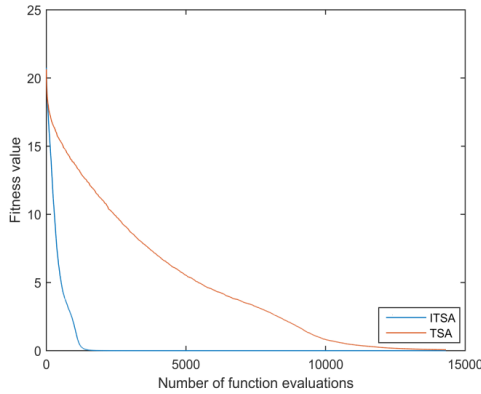


Fig. 5. Convergence of Ackley function.

V. CONCLUSION AND DISCUSSION

Tree-Seed Algorithm is a nature inspired algorithm that has recently been proposed by Kiran (2015) in the literature. The framework of the algorithm is quite simple. Therefore, TSA can be easily adapted many problems. In this study, it is aimed that progress the TSA's performance and solution quality by insertion new operators into TSA which is relatively a new algorithm. As a result, when the experimental results are taken into consideration, it is understood that ITSA is obtained more effective results for benchmark functions than TSA. ITSA uses the advantages of both local search and global search space. Thanks to this, it is prevented the solution gets trapped at a local optimum. According to experimental results, TSA finds better results just for F2 function. However, for the other benchmark functions ITSA finds quite better results. As the dimension of the problem increases, the solution quality of the ITSA increases. One of the important disadvantage of TSA is the number of seeds to be generated from a tree is selected randomly between 1 to numbers of population. Therefore, if the number of populations is increase, the number of seeds to be created from the tree will also increase. This situation will be reduced the efficiency of the algorithm. In future works, an operator may be defined that limits the number of seeds to be created from a tree depending on the size of the population. This modification may improve TSA's performance. In addition, new approaches can be offered to improve the parameter C in future studies.

REFERENCES

[1] K. G. Murty. (2003). *Optimization Models For Decision Making*. 1(1), pp. 1-18. [Online]. Available:

http://www-personal.umich.edu/~murty/books/opti_model/junior-1.pdf

[2] S. Jaballah, K. Rouis, F. B. Abdallah, and J. B. H. Tahar, "An improved shuffled frog leaping algorithm with a fast search strategy for optimization problems," presented at the 2014 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2014.

[3] J. Luo and M.-R. Chen, "Improved shuffled frog leaping algorithm and its multi-phase model for multi-depot vehicle routing problem," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2535-2545, 2014.

[4] D. Karaboğa, *Yapay Zekâ Optimizasyon Algoritmaları*, Nobel Yayın Dağıtım, p. 244, 2011.

[5] T. T. Nguyen, D. N. Vo, and A. V. Truong, "Cuckoo search algorithm for short-term hydrothermal scheduling," *Applied Energy*, vol. 132, pp. 276-287, 2014.

[6] M. S. Kiran, E. Özceylan, M. Gündüz, and T. Paksoy, "A novel hybrid approach based on particle swarm optimization and ant colony algorithm to forecast energy demand of Turkey," *Energy Conversion and Management*, vol. 53, no. 1, pp. 75-83, 2012.

[7] M. Gündüz, M. S. Kiran, and E. Özceylan, "A hierarchic approach based on swarm intelligence to solve the traveling salesman problem," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, no. 1, pp. 103-117, 2015.

[8] A. Ouaarab, B. Ahiod, and X.-S. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem," *Neural Computing and Applications*, vol. 24, no. 7-8, pp. 1659-1669, 2014.

[9] M. Aslan, 2017, "An approach based on shuffled frog leaping algorithm for vertex coloring problem in graphs," Selçuk University, The graduate school of natural and applied science, Konya.

[10] S. Mahmoudi and S. Lotfi, "Modified cuckoo optimization algorithm (MCOA) to solve graph coloring problem," *Applied Soft Computing*, vol. 33, pp. 48-64, 2015.

[11] N. Gould, "An introduction to algorithms for continuous optimization," *Oxford University Computing Laboratory Notes*, 2006.

[12] B. Alataş, 2007, "Development of chaotic maps embedded particle swarm optimization algorithms," PhD Thesis, Firat University, The graduate school of natural and applied science, Elazığ.

[13] M. Lovbjerg, "Improving particle swarm optimization by hybridization of stochastic search heuristics and self-organized criticality," PhD dissertation, Department of Computer Science, University of Aarhus, Datalogisk Institut, 2002.

[14] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," presented at the Sixth International Symposium on Micro Machine and Human Science, 1995.

[15] M. Dorigo and G. D. Caro, "Ant colony optimization: A new meta-heuristic," presented at Proceedings of the 1999 Congress on the Evolutionary Computation, 1999.

[16] D. Karaboga, *An idea Based on Honey Bee Swarm for Numerical Optimization*, 2005.

[17] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization*, pp. 65-74, 2010.

[18] S. A. Uymaz, G. Tezel, and E. Yel, "Artificial algae algorithm (AAA) for nonlinear global optimization," *Applied Soft Computing*, vol. 31, pp. 153-171, 2015.

[19] M. S. Kiran, "TSA: Tree-seed algorithm for continuous optimization," *Expert Systems with Applications*, vol. 42, no. 19, pp. 6686-6698, 2015.

[20] H. Hakli, *A Modified Cuckoo Search Using Different Search Strategies*, 2016.

[21] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," KanGAL Report, 2005.



Murat Aslan got his bachelor's degree in computer engineering from Selçuk University in 2011. Recently, he is a research assistant in the Department of the Computer Engineering at Selçuk University and working on his Ph.D. dissertation. His current interest include applications of graph coloring problem, optimization, artificial intelligent algorithms.



Mehmet Beskirlı got his bachelor degree of computer education in Selçuk University in 2009. Now he works as a research assistant and continues his doctoral degree in computer engineering at the same university. His research interests include artificial intelligence, algorithms and optimization techniques.



Halife Kodaz graduated from Computer Engineering Department of Selcuk University with B.S. degree in 1999. He received the Ph.D. in Electrical and Electronics Department from Selcuk University in 2008. He is an associate professor at the Computer Engineering Department at Selcuk University. His research interests are artificial intelligence and machine learning.



Mustafa Servet Kiran graduated from the Department of Computer Engineering at Selçuk University in 2007. His PhD degree was received from the Department of Computer Engineering at Selcuk University. He is an associate professor at the Computer Engineering Department at Selcuk university. His research interests are artificial intelligence, machine learning, optimization.