ISMSIT 2017
1st International Symposium on Multidisciplinary Studies and Innovative Technologies Proceedings
(November 2-4, 2017, Tokat, Turkey)
Symposium homepage: www.ismsitconf.org

DEVELOPMENT OF THE EGYPTIAN VULTURE OPTIMIZATION ALGORITHM WITH THE NEAREST OR FAREST NEIGHBORHOOD METHODS

[1]Ay sh gol IHSAN,[2]Erkan ULKER

[1] Computer Engineering Dept.,Graduate School Of Natural Sciences, Selcuk University, Konya, Turkey
[2]Computer Engineering, Engineering Faculty, Selcuk University, Konya, Turkey

Abstract - Nowadays, there are old and new methods that involve the difficult problems and solution suggestions which many researchers research and try to solve. In the field of engineering, the success rates of the methods that produce solutions to these problems are often compared to the travelling salesman problems. Many studies have been done to solve the problems in the best way. In this paper, a new method, the Egyptian Vulture algorithm, was used. The purpose of the study is to solve the Travelling Salesman problem by adding the nearest and the farthest neighbour structures to the Egyptian Vulture algorithm in order to obtain better results in the problems that occur in the data sets of the paper. The nearest neighbors or the farthest neighbors is a scientific contribution to the algorithm. Performance and success of the proposed method are showed by tests.

Keywords - Nearest Neighbor, Farthest Neighbor, Travelling Salesman Problem, Shortest Path, Longest Path, Travelling Salesman Problem.

1. INTRODUCTION

Problems occurring in the field of engineering are presented different solutions. The Travelling Salesman Problem (TSP) is used to measure the success rates of the results of proposed methods[1]. As a result of researches made in the recent years, many problems have been solved. Comparisons are made with Travelling Salesman Problem to measure the success rates obtained[2].

There is a salesman in the problem of TSP and he intends to sell his goods around all the cities. It is very important to arrange randomly generated data sets in the most appropriate way to the solution of the problem. Properly arranged data sets enable us to solve problems in the shortest possible time and at the shortest distance. The use of the nearest neighbors and farthest neighbors algorithm can play an important role when doing these operations.

The Egyptian Vulture Optimization (EVO) algorithm is a new method proposed in 2013 [3]. Sur et al. [3] studied EVO's extraordinary habits and unique perceptions to find nutrients. In their work, they used the EVO algorithm to solve the problem of TSP and Knapsack. Sur et al. [4] in 2013 he continued to introduce and implement EVO, a meta-heuristic algorithm in another study.

Although the EVO algorithm is successful, its performance is limited. In this paper, the TSP is solved with the EVO algorithm. The nearest and farthest neighbors to the EVO algorithm are added. The processes performed with the nearest neighbor and the farthest neighbor were used in the arrangement of the cities in the data sets.

Although the nearest neighbors and the farthest neighbors were proposed in new, it was enabled to obtain very good results according to the pure EVO algorithm in tests. This proposed study has made a great contribution to the development of solutions by the TSP.

It is the smallest vulture in Turkey and Europe. The body of the Egyptian vulture that lives in Egypt is made up of generic

2. TRAVEL SALESMAN PROBLEM (TSP)

TSP was introduced in 1930. Today, it is also used in real life, such as finding a route, managing charts and road traffic. In addition to the proposed mathematical solutions, solutions to TSP have been sought with nature-inspired methods such as artificial neural networks and genetic algorithms[5].

TSP is used to compare the success of study done in the field of engineering. There is a dealer in the problem of the traveling salesman and he intends to sell his goods around all the cities. Sellers start to tour the cities they are targeting to sell their goods, and each city only stops once to go back to the starting city and make a full tour.

Length distances of cities are known in advance. The traveler's problem is to complete the tour by finding the shortest route as soon as possible. It is quite easy to understand this process, but the solution is so difficult. He travels to other cities in the shortest distance from the city he is traveling for and he returns to his starting point, thus completing a full tour. He pre-selects the cities he will visit in advance. It takes care to be as short as possible and at low cost during the selection. It calculates the results of the TSP and gives it as complete and intuitive.

In the TSP, n the number of city. TSP is divided into two as Symmetric and Asymmetric. The symmetric TSP is equal to the distance between the two cities for cities i and j ($d_{ij} = d_{ji}$), but the asymmetric TSP does not have equality of distances. The euclidean TSP is a special case of the Symmetric TSP where it is located in the R^m space of cities for some m values that provide the triangular inequality of the cost function: $m \geq 1$ $(x_1, \ldots, x_m),(y_1, \ldots, y_m)$, $(\sum_{m=1}^{d} |x_i - y_i|^m)^{1/m} \varepsilon \ R^m$ for ℓ_m (where m=2 is the Euclidean norm). Two dimensional Euclidean TSP is a very popular subject that has been studied in the literature[6].

white fur. After 1960, the Egyptian vulture began to extinction. For the reason, unconscious hunting and superstition in India is mercilessly murdered because it is believed that the

Egyptian vulture was brought to prosperity. The Egyptian vulture was protected by IUCN (World Conservation Union) and animal lovers.

Ostriches continue their lives by leaving their eggs in flat environments. The camel bird is about one kilogram, which means 20-30 chicken eggs in average. It's pretty big, so it's pretty hard to break[3].

In the wild, the vulture tries to break the egg. The shell of the ostrich's egg is very hard and big, so it is very difficult to break the skin and reach the food. In order to do this, he carries a stone he finds from nature with his mouth and brings it to the egg and throws the stone in from his throat to the egg to break the eggshell.

For this bird species, the ostrich eggs are very valuable because the protein in the bag keeps it free for hours. Ostriches continue their lives by leaving their eggs in flat environments. Only the Egyptian Vulture has the ability to use stone as an instrument. For this reason, the Egyptian Vulture is known as a skilled bird [4]. The flow chart of the EVO Algorithm is as follows;

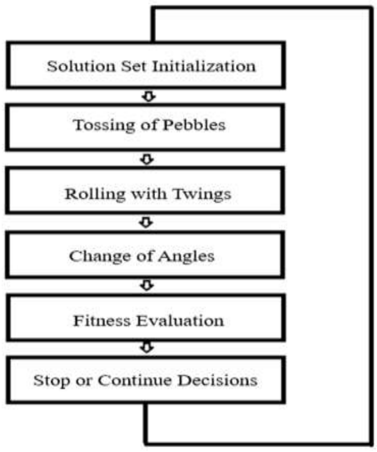

Figure 1. Flow Chart of Pure Egypt Vulture Optimization Algorithm (EVO)[3,4]

Egyptian vulture eggs find pebbles to break and randomly throw eggs. The Egyptian poultry breaks the eggshell a few times and successfully reaches the food. An Egyptian slaughtering stone finds the cracks in the egg. When he throws stones again, he allows the stone to coincide with the cracks. This is because the cracks in the egg are more easily broken with the stone crushed. The Egyptian gag that finds cracks on the egg changes the position of the egg and places the cracked part upwards.

To increase the likelihood of breakage of the egg, the cornflower takes a different angle to the egg which is not broken when the gravel is thrown. This increases the likelihood of fracture. The stones thrown from different angles provide breaking of the unbroken stone and reaching the bird food. The mutation makes egg breaking process easier, and this operation known as structural deterioration of eggs.

3. Addition of the Nearest Neighborhood Structure to EVO Algorithm for TSP Solving

The data sets used in almost every problem in engineering are first randomly generated. Every data in the data set has its own value. Sorting by randomly generated data sets, regardless of their values, will ensure that we get good results. During this process, two adjacent data are selected as random start and end points from the randomly generated data set. Then the nearest neighbor to the selected starting point is selected and written next to the starting point. Looking at the selected end point, the nearest neighbor to the end point is found and written next to the end point. Turning back to the starting point, the nearest neighbors of the selected neighbors are located near the starting point.

Thus, a solution is created in the dataset. If the dataset is larger, the same operations are repeated. From the starting point to the ending point, from the ending point to the starting point is decided by looking at the direction of the direction to be taken. Table 1 shows distances between sample cities. According to the selections made from these cities, the order in Table 2.a has been established. In the example, the decision from the starting point to the ending point is given by the closest distance (Table 2.e). The shortest distance was established because the cities in the nearest distances in the data set were taken into consideration.

In the example given in Table 2, *5* as the random start point and *3* as the end point were selected (Table 2.b). In the next process, the nearest city to the starting point, *5*, is written *4* (Table 2.c). Looking at the end point, the shortest city *1* is selected and added to the sequence (Table 2.d). The remaining number is added to the remaining city number *2* to complete the sequence. The first and last city is looked after for the completion of the turn, the shortest direction is preferred. For example, this direction is from *3* to *5*.

Table 1. Distance Values Between The Cities on the Sample Data Set which is Randomly Created.

Cities	1	2	3	4	5
1	0	9	17	3	5
2	3	0	5	16	9
3	2	7	0	9	3
4	10	1	8	0	7
5	8	3	15	2	0

Table 2. The Operation of the Nearest Neighbor Algorithm. a) First Data Set b) Selected Starting Points c) Neighbor Nearest to the Starting Point d) Neighbor Nearest to the Ending Point e) Newly Formed City Order

a)	2	1	5	3	4
b)	5				3
c)	5	4		1	3
d)	5	4		1	3
e)	5	4	2	1	3

4. Addition of the Farest Neighborhood Structure to EVO Algorithm for TSP Solving

Similar to the farthest neighbor algorithm, it works exactly the opposite way. A random data set is created as it is in the farthest neighbor. Two separate points are selected in the data set as random start and end points. In contrast to the nearest neighbor algorithm, the farthest point to the starting point is selected and written next to the starting point. The farthest neighbor to the end point is found and written next to the end point. By returning to the starting point again, the farthest neighbor of the selected neighbors away from the starting point is found and written. So the sequence is rebuilt. In case the number of cities is higher, the processes are repeated until the cities are finished.

From the starting point to the ending point, from the ending point to the starting point is decided by looking at the farthest distance to the direction information. Table 3 shows distances between cities. In the example given in Table 4, *5* as the random start point and *3* as the end point were selected (Table 4.b).

In the next process, the starting distance is *5*, which is the farthest city (Table 4.c). Looking at the end point, the two at the farthest distance are selected and added to the sequence (Table 4.d). The city is number *4*, the only remaining city, and the sequence is completed. The first and last city is looked for, and the longest direction is preferred. For example, this direction is from *3* to *5*.

Table 3. Distance Values Between the Cities on the Sample Data Set Which is Ramdomly Created

Cities	1	2	3	4	5
1	0	9	7	13	5
2	3	0	5	15	9
3	1	15	0	9	18
4	7	8	4	0	6
5	9	2	7	6	0

Table 4. The Operation Of The Farthest Neighbors Algorithm. a) The First Data Set b) The Selected Starting Points c) The Farthest Neighbors to the Starting Point d) The Farthest Neighbors to the Ending Point E) Newly Formed City Order

a)	2	1	5	3	4
b)	5				3
c)	5	1			3
d)	5	1		2	3
e)	5	1	4	2	3

The nearest neighbors or farthest neighbors are located after the mutation step in the pure EVO. The final flowchart shaped according to the developed neighbor approach is presented in Figure 2.

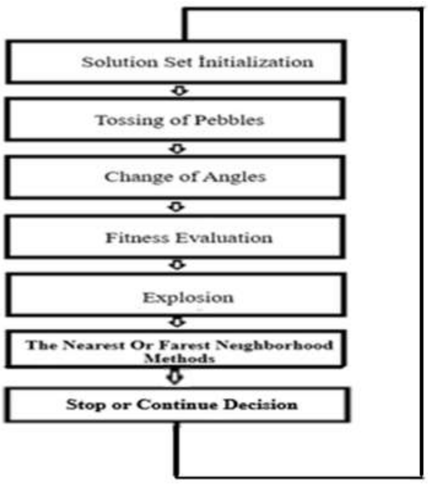

Figure 2. Usage of Developed EVO Algorithm in TSP

5. EXPERIMENTAL STUDIES

For the solution of the EVO algorithm and the TSP problem, the Berlin52, Eil51, Eil76, St70, Eil101 and Ch150 data sets selected from the TSPLIB library were used. Experiments were performed on an Intel (R) Core (TM) i5-5200U CPU @ 2,2 GHz 2,2 GHz processor with 8,00 GB RAM. The coding in the C # language has been implemented for the algorithm.

The Application of the Nearest Neighbor or The Farthest Neighbors Methods to the Egyptian Vulture Algorithm in the Pure EVO and the Travel Salesman Problem have been investigated with experimental studies. The experimental study was carried out with 20 iterations. In the proposed algorithm and pure EVO results, the best values for all iterations, the population average, and the average standard deviation values are stored.

Error values are calculated to measure the performance of the algorithm. The results for the library are given in Table 5 for Pure EVO and in Table 6 for improved EVO. In Figure 3, the best ways to obtain the results of the experimental study are shown visually.

Table 5. Success Rate of Pure EVO Algorithm

Iterasyon No	Berlin52	Eil51	Eil76	St70	Eil101	Kora100	ch150
1	27547	1490	2102	3262	3242	140887	50485
2	27308	1466	2272	3350	3100	144810	49987
3	27716	1486	2319	3375	3119	148448	50455
4	26931	1433	2330	3162	3127	148994	49819
5	25077	1381	2271	3327	3192	144484	50017
6	27431	1487	2243	3297	3035	145137	48134
7	27230	1468	2191	3400	3164	147323	49867
8	27739	1497	2268	3325	3083	151839	50387
9	27908	1481	2316	3350	3150	147719	49733
10	25748	1438	2268	3229	3091	140887	50556
11	27476	1466	2282	3367	3165	145860	50563
12	27148	1443	2289	3268	3117	140342	47531
13	25668	1434	2302	3300	3105	152216	48908
14	26252	1458	2257	3216	3168	146260	51308
15	27930	1410	2299	3162	3178	149159	50934
16	27133	1356	2292	3347	3147	150656	51338
17	27908	1469	2332	3381	3170	148924	49213
18	25496	1475	2295	3352	3173	146708	48006
19	28034	1443	2181	3326	3196	144810	49837
20	26563	1481	2295	3295	3158	150108	49956
best	25077	1356	2102	3162	3083	140342	47531
optimal:	7542	426	538	675	629	21282	6528
Avrage	27012,15	1453,1	2270,2	3304,55	3144	146778,6	49851,7
STDEV:	905,9632	37,04464	55,8952	69,16454	47,26966	3464,991	1042,977
Error(%):	905,9632	37,04464	55,8952	69,16454	47,26966	3464,991	1042,977
Max	28034	1497	2332	3400	3242	152216	51338

Table 6. Success Rate of Implementation of the Egyptian Vulture Algorithm with the Nearest Neighbor or the Farthest Neighbor Methods in the Travelling Salesman Problem

Iterasyon No	Berlin52	Eil51	Eil76	St70	Eil101	Kora100	ch150
1	9083	549	699	842	799	25713	7676
2	8843	505	681	846	793	27376	7664
3	9305	548	625	880	803	27231	7645
4	9504	529	691	881	848	26381	7529
5	8788	550	678	853	859	26254	7353
6	8583	561	625	894	869	26226	7547
7	8266	536	609	848	843	26404	7323
8	9083	575	696	846	826	26724	7329
9	8583	548	616	872	842	26122	7252
10	9004	546	678	859	817	26560	7559
11	9104	566	672	865	854	26314	7081
12	8266	499	613	879	843	26076	7323
13	9104	505	711	853	859	26719	7081
14	9305	548	683	840	793	26358	7645
15	9086	505	672	846	843	26703	7547
16	8583	536	693	863	859	25837	7664
17	9004	548	681	804	818	26533	7325
18	9104	529	616	842	858	26208	7518
19	8266	536	699	869	842	25713	7547
20	8788	566	681	834	793	25559	7353
best	8266	499	609	804	793	25559	7081
optimal:	7542	426	538	675	629	21282	6528
Avrage	8882,6	539,25	665,95	855,8	833,05	26350,55	7448,05
STDEV:	332,79	21,92	34,18	20,46	25,61	469,29	186,73
Error(%):	7,97	8,07	9,35	6,44	5,05	3,10	5,18
Max	9504	575	711	894	869	27376	7676

Using pure EVO, the error value for Berlin52 is 905.9632 while the error value obtained by the proposed method is ***7.97***. In the Eil51 problem, the error rate of pure EVO was 37,04464, while the error rate was found to be ***8.07*** with the proposed method. Error rates in other problems are found in Table 5 and Table 6. As you can see, the success rate of the proposed approach is very high according to pure EVO.

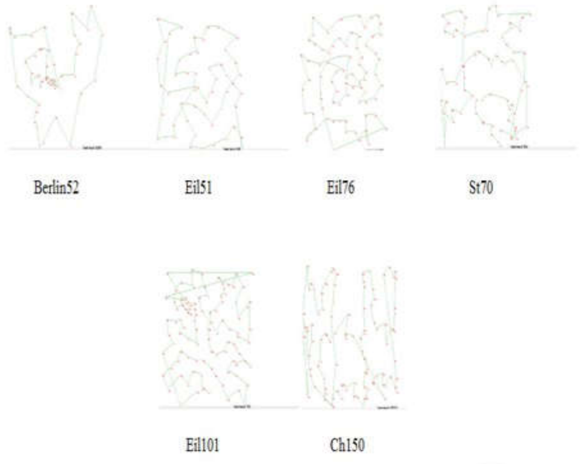

Figure 3. Visual presentation of best methods obtained with developed EVO

6. RESULT

The developing engineering field takes up a lot of space in our lives. The engineering field is looking for new algorithms to reach faster solutions in problems. The organization of data sets such as generating workflow scheduling table creation sequences is very important in the proposed studies.

The data sets used in almost every problem in engineering are first randomly generated. Each dataset in the dataset has its own distance value. Regardless of the values, it has been proved that the randomly generated data sets have good results in ordering by the studies done by their values.

In EVO algorithm, the cornflower that finds the ostrich egg tries to break the shell of the egg. The shell of ostrich egg is very hard and big, so it is very difficult to break the shell and reach the food. In order to do this, he again carries a stone he finds from nature, throwing it to the egg quickly and breaking the eggshell in a few moves. Using this feature of EVO, the pure EVO algorithm is proposed in the literature. The EVO has been developed with newly proposed structures because the performance of the Pure Egyptian vulture algorithm is poor. The algorithm developed is used in the solution of TSP. In the solutions, city lines are created and the nearest neighbors or farthest neighbors are looked at. Achieved solving with the obtained city orders is easier. In this paper, the nearest or farthest neighbors to the developed EVO algorithm have been added and shown with good results.

Acknowledgement: This paper is supported by Coordination of Selcuk University's Scientific Research Project

REFERENCES

1. F Glover, HJ Greenberg 1989 New approaches for heuristic search: A bilateral linkage with artificial intelligence, European Journal of Operational Research, pp 119 – 130, North-Holland.
2. DL Applegate, RE Bixby, V Chvatal, WJ Cook, 2007, the traveling salesman problem: a computational study, 59 pages, Princeton University Press (Date of access: 11.10.2017)
3. C. Sur, S. Sharma, A. Shukla, 2013, Egyptian Vulture Optimization Algorithm – A New Nature Inspired Meta-heuristics for Knapsack Problem, ABV Indıan Institute of in İnformation Technology & Managment Gwalior, pp 227-237, Madha Pradesh, India.
4. C. Sur, S. Sharma, A. Shukla, 2013, Solving Travelling Salesman Problem Using Egyptian Vulture Optimization Algorithm – A New Approach, ABV Indıan Institute of in İnformation Technology &Managment Gwalior, pp 254-267, Madha Pradesh, India.
5. M Dorigo, LM Gambardella, 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactıons On Evolutıonary Computation, pp 53 – 66.
6. H. Eldem, E. Ulker, 2014, Küre Üzerinde 3 Boyutlu Gezgin Satıcı Problemi Çözümünde Parçacık Sürü Optimizasyonu Uygulaması, XVI. Akademik Bilişim Konferansı Bildirileri Kitabı, pp 461-469, Mersin, Turkey (in Turkish language).